# Linux From Scratch

# Table of Contents

**Linux From Scratch....1**
Version 4.0....1
Gerard Beekmans....1

**Dédicace....2**

**Préface....7**

**Avant−propos....8**

**Qui voudrait lire ce livre....9**

**Qui ne voudrait pas lire ce livre....10**

**Organisation....11**
Partie I – Introduction....11
Partie II – Installation du système LFS....11
Partie III – Annexes....11

**I. Première Partie – Introduction....12**

**Chapitre 1. Introduction....13**

**Remerciements....14**

**Comment les choses se dérouleront−elles....16**

**Conventions utilisées dans ce livre....18**

**Version du livre....19**

**Sites miroirs....20**
Miroirs HTTP....20
Amérique du Nord....20
Europe....20
Australie....21
Miroirs FTP....21
Amérique du Nord....21
Europe....21
Australie....22

**Journal des modifications (Changelog)....23**

**Listes de diffusion et archives....36**
lfs−support....36
lfs−dev....36
lfs−announce....37
lfs−security....37

# Table of Contents

lfs−book....37
lfs−chat....37
alfs−discuss....37
blfs−dev....37
blfs−book....37
blfs−support....38
Archives des listes de diffusion....38
Comment envoyer un message sur une liste....38
Comment s'abonner ?....38
Comment se désabonner....38
Autres modes de listes....39
Résumé....39
En vacances....39

**Serveur News....40**

**FAQ....41**

**Contacts et informations....42**

**Chapitre 2. Informations importantes....43**

**A propos de $LFS....44**

**A propos des SBU....45**

**Où stocker les logiciels téléchargés....46**

**Comment installer les logiciels....47**

**Pour quelle plateforme?....49**

**Comment demander de l'aide....50**
Information de base....50
Problèmes de configuration....50
Problèmes de compilation....50

**II. Deuxième partie − Installation du système LFS....52**

**Chapitre 3. Packages qui doivent être téléchargés....53**

**Introduction....54**

**Packages nécessitant d'être téléchargés....55**

**Chapitre 4. Préparer une nouvelle partition....58**

# Table of Contents

**Introduction....59**

**Créer une nouvelle partition....60**

**Créer un système de fichiers sur la nouvelle partition....61**

**Monter la nouvelle partition....62**

**Chapitre 5. Préparer le système LFS....63**

**Introduction....64**

**Pourquoi utiliser une édition de lien statique ?....65**

**Création du répertoire $LFS/static....67**

**Installer tous les logiciels en tant qu'utilisateur non privilégié....68**

**Installer Bash−2.05a....69**
Installation de Bash....69
Détail de la commande....69
Contenu de Bash....70
Programmes....70
Descriptions....70
Dépendances d'installation de Bash....70

**Installer Binutils−2.13....71**
Installation de Binutils....71
Explication de la commande....71
Contenu de Binutils....71
Programmes....71
Descriptions....71
Bibliothèques....73
Descriptions....73
Dépendances d'installation de Binutils....73

**Installer Bzip2−1.0.2....74**
Installation de Bzip2....74
Explication des commandes....74
Contenu de Bzip2....74
Programmes....74
Descriptions....74
Bibliothèques....75
Dépendances d'installation de Bzip2....75

**Installer Diffutils−2.8.1....76**
Installation de Diffutils....76
Explication des commandes....76

# Table of Contents

Contenu de Diffutils....76
Programmes....76
Descriptions....76
Dépendances d'installation de Diffutils....76

**Installer Fileutils−4.1....78**
Installation de Fileutils....78
Explication de la commande....78
Contenu de Fileutils....78
Programmes....78
Descriptions....79
Dépendances d'installation de Fileutils....80

**Installer Findutils−4.1....82**
Installer Findutils....82
Explications des commandes....82
Contenu du package Findutils....82
Programmes....82
Descriptions....82
Dépendances d'installation de Findutils....83

**Installer Gawk−3.1.1....84**
Installation de Gawk....84
Contenu de Gawk....84
Programmes....84
Descriptions....84
Dépendances d'installation de Gawk....85

**Installer GCC−3.2....86**
Installation de GCC....86
Explication des commandes....86
Contenu de GCC....87
Programmes....87
Descriptions....87
Bibliothèques....88
Descriptions....88
Dépendances d'installation de GCC....88

**Installer Grep−2.5....89**
Installation de Grep....89
Explication des commandes....89
Contenu de Grep....89
Programmes....89
Descriptions....89
Dépendances d'installation de Grep....89

# Table of Contents

**Installer Gzip−1.2.4a....91**
Installation de Gzip....91
Contenu de Gzip....91
Programmes....91
Description....91
Dépendances d'installation de Gzip....92

**Installer Make−3.79.1....93**
Installation de Make....93
Contenu de Make....93
Programmes....93
Descriptions....93
Dépendances d'installation de Make....93

**Installer Patch−2.5.4....94**
Installation de Patch....94
Explication des commandes....94
Contenu de Patch....94
Programmes....94
Descriptions....94
Dépendances d'installation de Patch....94

**Installer Sed−3.02....95**
Installation de Sed....95
Contenu de Sed....95
Programmes....95
Descriptions....95
Dépendances d'installation de Sed....95

**Installer Sh−utils−2.0....96**
Installation de Sh−utils....96
Contenu de Sh−utils....96
Programmes....96
Descriptions....96
Dépendances d'installation de Sh−utils....99

**Installer Tar−1.13....100**
Installation de Tar....100
Contenu de Tar....100
Programmes....100
Descriptions....100
Dépendances d'installation de Tar....100

**Installer Texinfo−4.2....101**
Installation de Texinfo....101
Contenu de Texinfo....101
Programmes....101
Descriptions....101

# Table of Contents

Dépendances d'installation de Texinfo....102

**Installer Textutils−2.1....103**
Installation de Textutils....103
Contenu de Textutils....103
Programmes....103
Descriptions....103
Dépendances d'installation de Textutils....105

**Installer Util−linux−2.11u....106**
Installation de Util−linux....106
Dépendances d'installation de Util−linux....106

**Chapitre 6. Installation des logiciels de base du système....107**

**Introduction....108**

**A propos des symboles de débogage....109**

**Entrée dans l'environnement "chrooté"....110**

**Changer de propriétaire....111**

**Création des répertoires....112**
Notes de conformité FHS....112

**Monter le système de fichier proc....113**

**Créer le lien symbolique /etc/mtab....114**

**Création des liens symboliques bash et sh....115**

**Créer les fichiers passwd et group....116**

**Créer les périphériques (Makedev−1.7)....117**
Créer les périphériques....117
Explication des commandes....117
Contenu de MAKEDEV....117
Programmes....117
Descriptions....117
Dépendances d'installation de MAKEDEV....118

**Installer Linux−2.4.19....119**
Installation du noyau Linux....119
Explication des commandes....119
Pourquoi nous copions les entêtes du noyau et pourquoi nous ne créons pas de liens....119
Contenu de Kernel....120
Programmes....120

# Table of Contents

Descriptions....120
Dépendances d'installation de Linux....121

**Installer Man−pages−1.52....122**
Installation de Man−pages....122
Contents of Man−pages....122
Fichiers support....122
Descriptions....122
Dépendances d'installation de Man−pages....122

**Installer Glibc−2.2.5....123**
Installation de Glibc....123
Explication des commandes....124
Contenu de Glibc....125
Programmes....125
Descriptions....125
Bibliothèques....127
Descriptions....127
Dépendances d'installation de Glibc....130

**Installer GCC−3.2....131**
Installation de GCC....131
Explication des commandes....131
Contenu de GCC....131
Programmes....132
Descriptions....132
Bibliothèques....132
Descriptions....133
Dépendances d'installation de GCC....133

**Installer Zlib−1.1.4....134**
Installation de Zlib....134
Contenu de Zlib....134
Bibliothèques....134
Descriptions....134
Dépendances d'installation de Zlib....134

**Installer Findutils−4.1....135**
Installer Findutils....135
Note à propos de la conformité FHS....135
Explication des commandes....135
Contenu du package Findutils....135
Programmes....135
Descriptions....135
Dépendances d'installation de Findutils....136

# Table of Contents

**Installer Gawk−3.1.1....137**
Installation de Gawk....137
Explication de la commande....137
Contenu de Gawk....137
Programmes....137
Descriptions....137
Dépendances d'installation de Gawk....138

**Installer Ncurses−5.2....139**
Installation de Ncurses....139
Explication de la commande....139
Contenu de Ncurses....139
Programmes....139
Descriptions....139
Bibliothèques....141
Dépendances d'installation de Ncurses....141

**Installer Vim−6.1....142**
Installation de Vim....142
Explication des commandes....142
Contenu de Vim....142
Programmes....142
Descriptions....142
Dépendances d'installation de Vim....144

**Installation de Bison−1.35....145**
Installation de Bison....145
Contenu de bison....145
Programmes....145
Descriptions....145
Dépendances d'installation de Bison....146

**Installer Less−374....147**
Installation de Less....147
Contenu de Less....147
Programmes....147
Description....147
Dépendances d'installation de Less....147

**Installer Groff−1.18....148**
Installation de Groff....148
Explications des commandes....148
Contenu de Groff....148
Programmes....148
Descriptions....148
Dépendances d'installation de Groff....151

# Table of Contents

**Installer Textutils−2.1.....152**
Installation de Textutils.....152
Contenu de Textutils.....152
Programmes.....152
Descriptions.....152
Dépendances d'installation de Textutils.....154

**Installer Sed−3.02.....155**
Installation de Sed.....155
Contenu de Sed.....155
Programmes.....155
Descriptions.....155
Dépendances d'installation de Sed.....155

**Installer Flex−2.5.4a.....156**
Installation de Flex.....156
Contenu de Flex.....156
Programmes.....156
Descriptions.....156
Bibliothèques.....157
Descriptions.....157
Dépendances d'installation de Flex.....157

**Installation de Binutils−2.13.....158**
Installation de Binutils.....158
Explication des commandes.....158
Contenu de Binutils.....158
Programmes.....158
Descriptions.....158
Bibliothèques.....160
Descriptions.....160
Dépendances d'installation de Binutils.....160

**Installer Fileutils−4.1.....161**
Installation de Fileutils.....161
Contenu de Fileutils.....161
Programmes.....161
Descriptions.....161
Dépendances d'installation de Fileutils.....163

**Installer Sh−utils−2.0.....164**
Installation de Sh−utils.....164
Notes sur la conformité FHS.....164
Explications des commandes.....164
Contenu de Sh−utils.....164
Programmes.....164
Descriptions.....164
Dépendances d'installation de Sh−utils.....167

# Table of Contents

**Installer Gettext−0.11.5....168**
Installation de Gettext....168
Contenu de Gettext....168
Programmes....168
Descriptions....168
Bibliothèques....170
Descriptions....171
Dépendances d'installation de Gettext....171

**Installer Net−tools−1.60....172**
Installation de Net−tools....172
Explication des commandes....172
Contenu de Net−tools....172
Programmes....172
Descriptions....172
Dépendances d'installation de Net−tools....174

**Installer Perl−5.8.0....175**
Installation de Perl....175
Contenu de Perl....175
Programmes....175
Descriptions....175
Bibliothèques....177
Descriptions....177
Dépendances d'installation de Perl....179

**Installer Linux threads−2.2.5 man pages....180**
Installation des pages man Linux threads....180
Contenu des pages man Linux threads....180
Fichiers de support....180
Descriptions....180
Dépendances d'installation des pages man de Linux Threads....180

**Installer M4−1.4....181**
Installation de M4....181
Contenu de M4....181
Programmes....181
Descriptions....181
Dépendances d'installation de M4....181

**Installer Texinfo−4.2....182**
Installation de Texinfo....182
Explication des commandes....182
Contenu de Texinfo....182
Programmes....182
Descriptions....182
Dépendances d'installation de Texinfo....183

# Table of Contents

**Installation de Autoconf−2.53 ..... 184**
Installation de Autoconf ..... 184
Contenu de Autoconf ..... 184
Programmes ..... 184
Descriptions ..... 184
Dépendances d'installation d'Autoconf ..... 185

**Installation de Automake−1.6.3 ..... 186**
Installation de Automake ..... 186
Contenu de Automake ..... 186
Programmes ..... 186
Descriptions ..... 186
Dépendances d'installation d'Automake ..... 187

**Installation de Bash−2.05a ..... 188**
Installation de Bash ..... 188
Contenu de Bash ..... 188
Programmes ..... 188
Descriptions ..... 188
Dépendances d'installation de Bash ..... 188

**Installer File−3.39 ..... 189**
Installation de File ..... 189
Contenu de File ..... 189
Programmes ..... 189
Description ..... 189
Dépendances d'installation de File ..... 189

**Installer Libtool−1.4.2 ..... 190**
Installation de Libtool ..... 190
Contenu de libtool ..... 190
Programmes ..... 190
Descriptions ..... 190
Bibliothèques ..... 190
Descriptions ..... 190
Dépendances d'installation de Libtool ..... 190

**Installation de Bin86−0.16.3 ..... 191**
Installation de Bin86 ..... 191
Contenu de Bin86 ..... 191
Programmes ..... 191
Descriptions ..... 191
Dépendances d'installation de Bin86 ..... 192

**Installation de Bzip2−1.0.2 ..... 193**
Installation de Bzip2 ..... 193
Explication de la commande ..... 193
Contenu de Bzip2 ..... 193

# Table of Contents

Programmes....193
Descriptions....193
Bibliothèques....194
Dépendances d'installation de Bzip2....194

**Installer Ed−0.2....195**
Installation de Ed....195
Explication des commandes....195
Contenu de Ed....195
Programmes....195
Description....195
Dépendances d'installation de Ed....196

**Installer Kbd−1.06....197**
Installation de Kbd....197
Explications des commandes....197
Contenu de Kbd....197
Programmes....197
Descriptions....197
Dépendances d'installation de Kbd....199

**Installation de Diffutils−2.8.1....200**
Installation de Diffutils....200
Contenu de Diffutils....200
Programmes....200
Descriptions....200
Dépendances d'installation de Diffutils....200

**Installation de E2fsprogs−1.27....201**
Installation de E2fsprogs....201
Explication des commandes....201
Contenu du package E2fsprogs....201
Programmes....201
Descriptions....201
Bibliothèques....203
Descriptions....203
Dépendances d'installation d'E2fsprogs....204

**Installer Grep−2.5....205**
Installation de Grep....205
Contenu de Grep....205
Programmes....205
Descriptions....205
Dépendances d'installation de Grep....205

**Installer Gzip−1.2.4a....206**
Installation de Gzip....206
Explication de la commande....206

# Table of Contents

Contenu de Gzip....206
Programmes....206
Description....206
Dépendances d'installation de Gzip....207

**Installer Man−1.5k....208**
Installation de Man....208
Explication des commandes....208
Contenu de Man....208
Programmes....208
Descriptions....208
Dépendances d'installation de Man....209

**Installer Lilo−22.2....210**
Installation de Lilo....210
Contenu de Lilo....210
Programmes....210
Descriptions....210
Dépendances d'installation de Lilo....211

**Installer Make−3.79.1....212**
Installation de Make....212
Explication des commandes....212
Contenu de Make....212
Programmes....212
Descriptions....212
Dépendances d'installation de Make....212

**Installer Modutils−2.4.19....213**
Installation de Modutils....213
Contenu de Modutils....213
Programmes....213
Descriptions....213
Dépendances d'installation de Modutils....214

**Installer Netkit−base−0.17....215**
Installation de Netkit−base....215
Contenu de Netkit−base....215
Program Files....215
Descriptions....215
Dépendances d'installation de Netkit−base....215

**Installer Patch−2.5.4....216**
Installation de Patch....216
Contenu de Patch....216
Programmes....216
Descriptions....216
Dépendances d'installation de Patch....216

# Table of Contents

**Installer Procinfo−18 ..... 217**
Installation de Procinfo ..... 217
Explication des Commandes ..... 217
Contenu de Procinfo ..... 217
Programmes ..... 217
Descriptions ..... 217
Dépendances d'installation de Procinfo ..... 217

**Installer Procps−2.0.7 ..... 218**
Installation de Procps ..... 218
Explication des Commandes ..... 218
Contenu de Procps ..... 218
Programmes ..... 218
Descriptions ..... 218
Bibliothèques ..... 219
Descriptions ..... 219
Dépendances d'installation de Procps ..... 220

**Installer Psmisc−21 ..... 221**
Installation de Psmisc ..... 221
Explication des Commandes ..... 221
Contenu de Psmisc ..... 221
Programmes ..... 221
Descriptions ..... 221
Dépendances d'installation de Psmisc ..... 222

**Installer Shadow−4.0.3 ..... 223**
Installation de Shadow Password Suite ..... 223
Explication des commandes ..... 223
Contenu de Shadow ..... 223
Programmes ..... 223
Descriptions ..... 224
Bibliothèques ..... 226
Descriptions ..... 226
Dépendances d'installation de Shadow ..... 227

**Installer Sysklogd−1.4.1 ..... 228**
Installation de Sysklogd ..... 228
Contenu de Sysklogd ..... 228
Programmes ..... 228
Descriptions ..... 228
Dépendances d'installation de Sysklogd ..... 228

**Installer Sysvinit−2.84 ..... 229**
Installation de Sysvinit ..... 229
Contenu de Sysvinit ..... 229
Programmes ..... 229
Descriptions ..... 229

# Table of Contents

Dépendances d'installation de Sysvinit....231

**Installer Tar−1.13....232**
Installation de Tar....232
Contenu de Tar....232
Programmes....232
Descriptions....232
Dépendances d'installation de Tar....232

**Installer Util−linux−2.11u....233**
Notes sur la conformité FHS....233
Installation de Util−linux....233
Explication des commandes....233
Contenu de Util−linux....233
Programmes....233
Descriptions....233
Dépendances d'installation de Util−linux....239

**Installation de LFS−Bootscripts−1.10....240**
Installation de LFS−Bootscripts....240
Contenu de LFS−bootscripts....240
Scripts....240
Descriptions....240
Dépendances d'installation de Bootscripts....242

**Configurer les logiciels essentiels....243**
Configurer Vim....243
Configurer Glibc....243
Configurer le chargeur dynamique....244
Configurer Sysklogd....244
Configurer Shadow Password Suite....244
Configurer Sysvinit....245
Configuration du clavier....245
Créer les fichiers /var/run/utmp, /var/log/wtmp et /var/log/btmp....246
Créer un mot de passe pour root....246

**Chapitre 7. Mise en place des scripts de démarrage....247**

**Introduction....248**

**Comment fonctionne le processus de démarrage utilisant ces scripts?....249**

**Configuration du script setclock....251**

**Ai−je besoin du script loadkeys?....252**

# Table of Contents

**Configuration du script sysklogd....253**

**Configurer le script localnet....254**

**Création du fichier /etc/hosts....255**

**Configuration du script network....256**
Configuration de la passerelle par défaut....256
Création des fichiers de configuration d'interfaces....256

**Chapitre 8. Rendre le système LFS démarrable....257**

**Introduction....258**

**Créer le fichier /etc/fstab....259**

**Installer Linux−2.4.19....260**
Dépendances d'installation de Linux....260

**Rendre son système LFS démarrable....261**

**Chapitre 9. La fin....262**

**La fin....263**

**Enregistrez−vous....264**

**Redémarrer le système....265**

**III. Troisième Partie − Annexes....266**

**Annexe A. Descriptions des packages et dépendances....267**

**Introduction....268**

**Autoconf....269**
Site officiel de téléchargement....269
Contenu de Autoconf....269
Programmes....269
Descriptions....269
Dépendances d'installation d'Autoconf....270

**Automake....271**
Site Officiel de Téléchargement....271
Contenu de Automake....271
Programmes....271
Descriptions....271
Dépendances d'installation d'Automake....272

# Table of Contents

**Bash....273**
Site Officiel de Téléchargements....273
Contenu de Bash....273
Programmes....273
Descriptions....273
Dépendances d'installation de Bash....273

**Bin86....274**
Site Officiel de Téléchargement....274
Contenu de Bin86....274
Programmes....274
Descriptions....274
Dépendances d'installation de Bin86....274

**Binutils....275**
Site Officiel de Téléchargement....275
Contenu de Binutils....275
Programmes....275
Descriptions....275
Bibliothèques....276
Descriptions....277
Dépendances d'installation de Binutils....277

**Bison....278**
Site officiel de téléchargement....278
Contenu de bison....278
Programmes....278
Descriptions....278
Dépendances d'installation de Bison....279

**Bzip2....280**
Site officiel de....280
Contenu de Bzip2....280
Programmes....280
Descriptions....280
Bibliothèques....281
Dépendances d'installation de Bzip2....281

**Diffutils....282**
Site officiel de....282
Contenu de Diffutils....282
Programmes....282
Descriptions....282
Dépendances d'installation de Diffutils....282

**E2fsprogs....283**
Site officiel de téléchargement....283
Contenu du package E2fsprogs....283

# Table of Contents

Programmes....283
Descriptions....283
Bibliothèques....285
Descriptions....285
Dépendances d'installation d'E2fsprogs....285

**Ed....286**
Site officiel de téléchargement....286
Contenu de Ed....286
Programmes....286
Description....286
Dépendances d'installation de Ed....286

**File....287**
Site officiel de téléchargement....287
Contenu de File....287
Programmes....287
Description....287
Dépendances d'installation de File....287

**Fileutils....288**
Site officiel de téléchargement....288
Contenu de Fileutils....288
Programmes....288
Descriptions....288
Dépendances d'installation de Fileutils....290

**Findutils....291**
Site officiel de téléchargement....291
Contenu du package Findutils....291
Programmes....291
Descriptions....291
Dépendances d'installation de Findutils....292

**Flex....293**
Site officiel de téléchargement....293
Contenu de Flex....293
Programmes....293
Descriptions....293
Bibliothèques....293
Descriptions....293
Dépendances d'installation de Flex....293

**Gawk....295**
Site officiel de téléchargement....295
Contenu de Gawk....295
Programmes....295
Descriptions....295

# Table of Contents

Dépendances d'installation de Gawk....295

**GCC....296**
Site officiel de téléchargement....296
Contenu de GCC....296
Programmes....296
Descriptions....296
Bibliothèques....297
Descriptions....297
Dépendances d'installation de GCC....297

**Gettext....298**
Site officiel de téléchargement....298
Contenu de Gettext....298
Programmes....298
Descriptions....298
Bibliothèques....300
Descriptions....300
Dépendances d'installation de Gettext....301

**Glibc....302**
Site officiel de téléchargement....302
Contenu de Glibc....302
Programmes....302
Descriptions....302
Bibliothèques....304
Descriptions....304
Dépendances d'installation de Glibc....307

**Grep....308**
Site officiel de téléchargement....308
Contenu de Grep....308
Programmes....308
Descriptions....308
Dépendances d'installation de Grep....308

**Groff....309**
Site officiel de téléchargement....309
Contenu de Groff....309
Programmes....309
Descriptions....309
Dépendances d'installation de Groff....312

**Gzip....313**
Site officiel de téléchargement....313
Contenu de Gzip....313
Programmes....313
Description....313

# Table of Contents

Dépendances d'installation de Gzip....314

**Kbd....315**
Site officiel de téléchargement....315
Contenu de Kbd....315
Programmes....315
Descriptions....315
Dépendances d'installation de Kbd....317

**Less....318**
Site officiel de téléchargement....318
Contenu de Less....318
Programmes....318
Description....318
Dépendances d'installation de Less....318

**LFS−Bootscripts....319**
Site officiel de....319
Contenu de LFS−bootscripts....319
Scripts....319
Descriptions....319
Dépendances d'installation de Bootscripts....321

**Libtool....322**
Site officiel de téléchargement....322
Contenu de libtool....322
Programmes....322
Descriptions....322
Bibliothèques....322
Descriptions....322
Dépendances d'installation de Libtool....322

**Lilo....323**
Site officiel de téléchargement....323
Contenu de Lilo....323
Programmes....323
Descriptions....323
Dépendances d'installation de Lilo....323

**Linux (le noyau)....324**
Site officiel de téléchargement....324
Contenu de Kernel....324
Programmes....324
Descriptions....324
Dépendances d'installation de Linux....324

# Table of Contents

**M4** ..... **325**
Site officiel de téléchargement ..... 325
Contenu de M4 ..... 325
Programmes ..... 325
Descriptions ..... 325
Dépendances d'installation de M4 ..... 325

**Make** ..... **326**
Site officiel de téléchargement ..... 326
Contenu de Make ..... 326
Programmes ..... 326
Descriptions ..... 326
Dépendances d'installation de Make ..... 326

**MAKEDEV** ..... **327**
Site officiel de téléchargement ..... 327
Contenu de MAKEDEV ..... 327
Programmes ..... 327
Descriptions ..... 327
Dépendances d'installation de MAKEDEV ..... 327

**Man** ..... **328**
Site officiel de téléchargement ..... 328
Contenu de Man ..... 328
Programmes ..... 328
Descriptions ..... 328
Dépendances d'installation de Man ..... 329

**Man−pages** ..... **330**
Site officiel de téléchargement ..... 330
Contents of Man−pages ..... 330
Fichiers support ..... 330
Descriptions ..... 330
Dépendances d'installation de Man−pages ..... 330

**Modutils** ..... **331**
Site officiel de téléchargement ..... 331
Contenu de Modutils ..... 331
Programmes ..... 331
Descriptions ..... 331
Dépendances d'installation de Modutils ..... 332

**Ncurses** ..... **333**
Site officiel de téléchargement ..... 333
Contenu de Ncurses ..... 333
Programmes ..... 333
Descriptions ..... 333
Bibliothèques ..... 334

# Table of Contents

Dépendances d'installation de Ncurses....334

**Netkit−base....336**
Site officiel de téléchargement....336
Contenu de Netkit−base....336
Program Files....336
Descriptions....336
Dépendances d'installation de Netkit−base....336

**Net−tools....337**
Site officiel de téléchargement....337
Contenu de Net−tools....337
Programmes....337
Descriptions....337
Dépendances d'installation de Net−tools....338

**Patch....339**
Site officiel de téléchargement....339
Contenu de Patch....339
Programmes....339
Descriptions....339
Dépendances d'installation de Patch....339

**Perl....340**
Site officiel de téléchargement....340
Contenu de Perl....340
Programmes....340
Descriptions....340
Bibliothèques....342
Descriptions....342
Dépendances d'installation de Perl....343

**Procinfo....345**
Site officiel de téléchargement....345
Contenu de Procinfo....345
Programmes....345
Descriptions....345
Dépendances d'installation de Procinfo....345

**Procps....346**
Site officiel de téléchargement....346
Contenu de Procps....346
Programmes....346
Descriptions....346
Bibliothèques....347
Descriptions....347
Dépendances d'installation de Procps....347

# Table of Contents

**Psmisc....348**
Site officiel de téléchargement....348
Contenu de Psmisc....348
Programmes....348
Descriptions....348
Dépendances d'installation de Psmisc....348

**Sed....349**
Site officiel de téléchargement....349
Contenu de Sed....349
Programmes....349
Descriptions....349
Dépendances d'installation de Sed....349

**Shadow....350**
Site officiel de téléchargement....350
Contenu de Shadow....350
Programmes....350
Descriptions....350
Bibliothèques....353
Descriptions....353
Dépendances d'installation de Shadow....353

**Sh−utils....354**
Site officiel de téléchargement....354
Contenu de Sh−utils....354
Programmes....354
Descriptions....354
Dépendances d'installation de Sh−utils....357

**Sysklogd....358**
Site officiel de téléchargement....358
Contenu de Sysklogd....358
Programmes....358
Descriptions....358
Dépendances d'installation de Sysklogd....358

**Sysvinit....359**
Site officiel de téléchargement....359
Contenu de Sysvinit....359
Programmes....359
Descriptions....359
Dépendances d'installation de Sysvinit....361

**Tar....362**
Site officiel de téléchargement....362
Contenu de Tar....362
Programmes....362

# Table of Contents

Descriptions....362
Dépendances d'installation de Tar....362

**Texinfo....363**
Site officiel de téléchargement....363
Contenu de Texinfo....363
Programmes....363
Descriptions....363
Dépendances d'installation de Texinfo....364

**Textutils....365**
Site officiel de téléchargement....365
Contenu de Textutils....365
Programmes....365
Descriptions....365
Dépendances d'installation de Textutils....367

**Util−linux....368**
Site officiel de téléchargement....368
Contenu de Util−linux....368
Programmes....368
Descriptions....368
Dépendances d'installation de Util−linux....373

**Vim....374**
Site officiel de téléchargement....374
Contenu de Vim....374
Programmes....374
Descriptions....374
Dépendances d'installation de Vim....376

**Zlib....377**
Site officiel de téléchargement....377
Contenu de Zlib....377
Bibliothèques....377
Descriptions....377
Dépendances d'installation de Zlib....377

**Annexe B. Ressources....378**

**Introduction....379**

**Livres....380**

**HOWTOs Et Guides....381**

# Table of Contents

**Autres.....382**

# Linux From Scratch

## Version 4.0


### Gerard Beekmans



Copyright © 1999−2002 par Gerard Beekmans


Ce livre décrit le processus de création d'un système Linux depuis rien, en utilisant uniquement les sources des logiciels utilisés.


Copyright (c) 1999−2002, Gerard Beekmans

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions in any form must retain the above copyright notice, this list of conditions and the following disclaimer.
- Neither the name of "Linux From Scratch" nor the names of its contributors may be used to endorse or promote products derived from this material without specific prior written permission.
- Any material derived from Linux From Scratch must contain a reference to the "Linux From Scratch" project.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# Dédicace

Ce livre est dédié à ma femme aimante, *Beverly Beekmans*, au soutien indéfectible.

***Table des matières***

*Préface*
- *Avant−propos*
- *Qui voudrait lire ce livre*
- *Qui ne voudrait pas lire ce livre*
- *Organisation*
  - *Partie I − Introduction*
  - *Partie II − Installation du système LFS*
  - *Partie III − Annexes*

*I. Première Partie − Introduction*
- *1. Introduction*
  - *Remerciements*
  - *Comment les choses se dérouleront−elles*
  - *Conventions utilisées dans ce livre*
  - *Version du livre*
  - *Sites miroirs*
  - *Journal des modifications (Changelog)*
  - *Listes de diffusion et archives*
  - *Serveur News*
  - *FAQ*
  - *Contacts et informations*
- *2. Informations importantes*
  - *A propos de $LFS*
  - *A propos des SBU*
  - *Où stocker les logiciels téléchargés*
  - *Comment installer les logiciels*
  - *Pour quelle plateforme?*
  - *Comment demander de l'aide*

*II. Deuxième partie − Installation du système LFS*
- *3. Packages qui doivent être téléchargés*
  - *Introduction*
  - *Packages nécessitant d'être téléchargés*
- *4. Préparer une nouvelle partition*
  - *Introduction*
  - *Créer une nouvelle partition*
  - *Créer un système de fichiers sur la nouvelle partition*
  - *Monter la nouvelle partition*
- *5. Préparer le système LFS*
  - *Introduction*
  - *Pourquoi utiliser une édition de lien statique ?*
  - *Création du répertoire $LFS/static*
  - *Installer tous les logiciels en tant qu'utilisateur non privilégié*
  - *Installer Bash−2.05a*
  - *Installer Binutils−2.13*
  - *Installer Bzip2−1.0.2*
  - *Installer Diffutils−2.8.1*

*Installer Fileutils−4.1*
*Installer Findutils−4.1*
*Installer Gawk−3.1.1*
*Installer GCC−3.2*
*Installer Grep−2.5*
*Installer Gzip−1.2.4a*
*Installer Make−3.79.1*
*Installer Patch−2.5.4*
*Installer Sed−3.02*
*Installer Sh−utils−2.0*
*Installer Tar−1.13*
*Installer Texinfo−4.2*
*Installer Textutils−2.1*
*Installer Util−linux−2.11u*
*6. Installation des logiciels de base du système*
*Introduction*
*A propos des symboles de débogage*
*Entrée dans l'environnement "chrooté"*
*Changer de propriétaire*
*Création des répertoires*
*Monter le système de fichier proc*
*Créer le lien symbolique /etc/mtab*
*Création des liens symboliques bash et sh*
*Créer les fichiers passwd et group*
*Créer les périphériques (Makedev−1.7)*
*Installer Linux−2.4.19*
*Installer Man−pages−1.52*
*Installer Glibc−2.2.5*
*Installer GCC−3.2*
*Installer Zlib−1.1.4*
*Installer Findutils−4.1*
*Installer Gawk−3.1.1*
*Installer Ncurses−5.2*
*Installer Vim−6.1*
*Installation de Bison−1.35*
*Installer Less−374*
*Installer Groff−1.18*
*Installer Textutils−2.1*
*Installer Sed−3.02*
*Installer Flex−2.5.4a*
*Installation de Binutils−2.13*
*Installer Fileutils−4.1*
*Installer Sh−utils−2.0*
*Installer Gettext−0.11.5*
*Installer Net−tools−1.60*
*Installer Perl−5.8.0*
*Installer Linux threads−2.2.5 man pages*
*Installer M4−1.4*
*Installer Texinfo−4.2*
*Installation de Autoconf−2.53*
*Installation de Automake−1.6.3*

*Installation de Bash−2.05a*
*Installer File−3.39*
*Installer Libtool−1.4.2*
*Installation de Bin86−0.16.3*
*Installation de Bzip2−1.0.2*
*Installer Ed−0.2*
*Installer Kbd−1.06*
*Installation de Diffutils−2.8.1*
*Installation de E2fsprogs−1.27*
*Installer Grep−2.5*
*Installer Gzip−1.2.4a*
*Installer Man−1.5k*
*Installer Lilo−22.2*
*Installer Make−3.79.1*
*Installer Modutils−2.4.19*
*Installer Netkit−base−0.17*
*Installer Patch−2.5.4*
*Installer Procinfo−18*
*Installer Procps−2.0.7*
*Installer Psmisc−21*
*Installer Shadow−4.0.3*
*Installer Sysklogd−1.4.1*
*Installer Sysvinit−2.84*
*Installer Tar−1.13*
*Installer Util−linux−2.11u*
*Installation de LFS−Bootscripts−1.10*
*Configurer les logiciels essentiels*
*7. Mise en place des scripts de démarrage*
*Introduction*
*Comment fonctionne le processus de démarrage utilisant ces scripts?*
*Configuration du script setclock*
*Ai−je besoin du script loadkeys?*
*Configuration du script sysklogd*
*Configurer le script localnet*
*Création du fichier /etc/hosts*
*Configuration du script network*
*8. Rendre le système LFS démarrable*
*Introduction*
*Créer le fichier /etc/fstab*
*Installer Linux−2.4.19*
*Rendre son système LFS démarrable*
*9. La fin*
*La fin*
*Enregistrez−vous*
*Redémarrer le système*
*III. Troisième Partie − Annexes*
*A. Descriptions des packages et dépendances*
*Introduction*
*Autoconf*
*Automake*
*Bash*

*Bin86*
*Binutils*
*Bison*
*Bzip2*
*Diffutils*
*E2fsprogs*
*Ed*
*File*
*Fileutils*
*Findutils*
*Flex*
*Gawk*
*GCC*
*Gettext*
*Glibc*
*Grep*
*Groff*
*Gzip*
*Kbd*
*Less*
*LFS−Bootscripts*
*Libtool*
*Lilo*
*Linux (le noyau)*
*M4*
*Make*
*MAKEDEV*
*Man*
*Man−pages*
*Modutils*
*Ncurses*
*Netkit−base*
*Net−tools*
*Patch*
*Perl*
*Procinfo*
*Procps*
*Psmisc*
*Sed*
*Shadow*
*Sh−utils*
*Sysklogd*
*Sysvinit*
*Tar*
*Texinfo*
*Textutils*
*Util−linux*
*Vim*
*Zlib*

*B. Ressources*

*Introduction*

*Livres*
*HOWTOs Et Guides*
*Autres*

# Préface

# Avant–propos

Ayant utilisé certaines distributions Linux, je n'ai jamais été satisfait par aucune d'entre elles. Je n'aimais pas la façon dont les scripts de démarrage étaient arrangés. Je n'aimais pas la façon dont certains programmes étaient configurés par défaut. Beaucoup de choses comme celles–ci m'ennuyaient. Finalement, j'ai réalisé que si je souhaitais avoir une complète satisfaction de mon système Linux, je devrais construire mon propre système à partir de rien, en utilisant uniquement les codes source. Je me suis résolu à n'utiliser aucun package pré–compilé, aucun CDRom ou disque de démarrage qui auraient installé quelques utilitaires simples. J'utiliserais mon système Linux courant pour développer le mien.

Cette idée étrange semblait très difficile à ce moment et m'a souvent semblé une tâche impossible. Après avoir traité toutes sortes de problèmes, comme les erreurs de dépendances et de compilation, un système Linux personnalisé et complètement opérationnel était créé. Je l'ai appellé Linux From Scratch (Linux Par Le Début, ou Linux à Partir de Rien), ou, plus simplement, LFS.

J'espère que vous prendrez plaisir à travailler sur votre LFS!

–– Gerard Beekmans gerard@linuxfromscratch.org

# Qui voudrait lire ce livre

Il y a beaucoup de raisons qui pousseraient quelqu'un à vouloir lire ce livre. La raison principale est d'installer un système LFS. La question que beaucoup de personnes se posent est "pourquoi se fatiguer à installer manuellement un système Linux depuis le début alors qu'il suffit de télécharger une distribution existante?". C'est une bonne question.

Une raison importante de l'existence de LFS est d'apprendre comment fonctionne un système Linux de l'intérieur. Construire un système LFS vous apprend tout ce qui fait que Linux fonctionne, et comment les choses interagissent et dépendent les unes des autres, et le plus important, vous apprend à le personnaliser afin qu'il soit à votre goût et réponde à vos besoins.

Un avantage clé de LFS est que vous avez plus de contrôle sur votre système sans avoir à dépendre d'une implémentation créée par quelqu'un d'autre. Avec LFS, vous êtes maintenant sur le siège conducteur et êtes capable de décider chaque aspect de votre système, comme la disposition des répertoires ainsi que la configuration des scripts de démarrage. Vous saurez également exactement où, pourquoi et comment les programmes sont installés.

Un autre avantage de LFS est la possibilité de créer un système Linux très compact. Quand vous installez une distribution courante, vous finissez par installer beaucoup de programmes que vous n'utiliserez jamais de votre vie. Ils sont juste là et occupent un espace disque précieux. Il n'est pas difficile de construire un système LFS de moins de 100 Mo. Cela vous semble−t−il toujours beaucoup ? Certains d'entre nous ont travaillé afin de créer un système LFS minuscule. Nous avons installé un système juste suffisant pour faire fonctionner le serveur web Apache; l'espace disque total occupé était approximativement de 8 Mo. Avec plus de dépouillement encore, cela peut être ramené à 5 Mo ou moins. Essayez donc d'en faire autant avec une distribution courante!

Si nous devions comparer une distribution Linux à un hamburger que vous achetez au restaurant fast−food, vous n'avez aucune idée de ce que vous mangez. LFS ne vous donne pas un hamburger, mais la recette pour faire un hamburger. Cela vous permet de prudemment l'inspecter, d'enlever les ingrédients non désirés, et par la même occasion vous permet de rajouter des ingrédients qui correspondent mieux à la saveur que vous attendez de ce hamburger. Quand vous êtes satisfait des ingrédients, vous passez à l'étape suivante en les combinant ensemble. Vous avez désormais la chance de pouvoir le faire de la façon dont vous le souhaitez: grillez−le, faites−le cuire au four, faites−le frire, faites−le au barbecue, ou mangez−le cru.

Une autre analogie que nous pouvons utiliser est de comparer LFS à une maison construite. LFS vous donnera les murs de la maison, mais c'est à vous de la construire, en vous donnant la liberté d'ajuster vos plans comme vous le souhaitez.

Un autre avantage d'un système Linux personnalisé est un surcroît de sécurité. Vous compilerez le système complet à partir de la base, ce qui vous permet de tout vérifier, si vous le voulez, et d'appliquer tous les correctifs de sécurité que vous voulez ou devez appliquer. Vous n'avez pas à attendre que quelqu'un d'autre vous fournisse un package réparant une faille de sécurité. Cependant, vous n'avez aucune garantie que le nouveau package résolve le problème (adéquatement). Vous ne pourrez jamais savoir si une faille de sécurité est réparée si vous ne le faites pas vous−même.

# Qui ne voudrait pas lire ce livre

Si vous ne souhaitez pas contruire votre propre système à partir des sources, alors vous ne voudrez probablement pas lire ce livre. Notre but est de construire les fondations d'un système complet et utilisable. Si vous souhaitez seulement connaître ce qui se passe lorsque votre ordinateur démarre, alors nous vous recommendons le document HOWTO << From−PowerUp−To−Bash−Prompt>> . Ce HOWTO construit un système basique similaire à celui de ce livre, mais il s'occupe de la création d'un système capable de démarrer jusqu'au prompt BASH.

Pendant que vous réfléchissez à celui que vous allez lire, cherchez votre objectif. Si vous souhaitez construire un système Linux tout en apprenant, alors ce livre est certainement le meilleur choix. Si votre objectif est strictement éducatif, et que vous n'avez aucune envie d'utiliser le système en question, alors le HOWTO << From−PowerUp−To−Bash−Prompt>> est probablement le meilleur choix.

Le HOWTO << From−PowerUp−To−Bash−Prompt>> est disponible sur http://www.netspace.net.au/~gok/power2bash/

# Organisation

La plupart des annexes est intégré dans la deuxième partie (ce qui d'une certaine façon agrandit le livre). Nous pensons que ceci rend la lecture plus facile. De cette façon, vous n'avez pas à vous référer en permanence à une annexe lors de la lecture de la deuxième partie. Ceci est une vrai corvée, particulièrement si vous lisez la version texte de ce livre. Ce livre est partagé en plusieurs parties:

## Partie I – Introduction

La première partie donne les informations générales à propos du contenu de ce livre (les versions, où se le procurer, le journal des modifications, les listes de diffusions, et comment nous contacter). Elle contient des lectures suggérées, comportant quelques considérations importantes qu'il est intéressant de connaître avant de commencer son système LFS.

## Partie II – Installation du système LFS

La seconde partie va vous guider à travers la construction et l'installation d'un système LFS. Ce système sera la fondation du reste de votre système Linux. Quel que soit l'usage que vous ferez de votre nouveau système LFS, il devra être construit sur les fondations installées dans cette partie.

## Partie III – Annexes

La troisième partie contient différentes annexes.

# I. Première Partie – Introduction

***Table des matières***

*1. Introduction*
*2. Informations importantes*

# Chapitre 1. Introduction

# Remerciements

Nous remercions les personnes et les organisations suivantes pour leurs contributions envers le projet Linux From Scratch:

- Mark Stone <mstone@linux.com> pour le don du serveur linuxfromscratch.org.
- VA Linux Systems pour assurer la maintenance de la machine et la bande passante du serveur linuxfromscratch.org.
- Fredrik Danerklint pour la maintenance du miroir se.linuxfromscratch.org.
- Tim Jackson <tim@idge.net> pour la maintenance du miroir linuxfromscratch.idge.net.
- Hagen Herrschaft <hrx@hrxnet.de> pour la maintenance du miroir de.linuxfromscratch.org, et pour sa donation d'un système à base de P4−2.2GHz au projet LFS.
- UK Mirror Service pour la maintenance du site miroir linuxfromscratch.mirror.ac.uk.
- Guido Passet <guido@primerelay.net> pour la maintenance du miroir www.nl.linuxfromscratch.org et ftp.snt.utwente.nl.
- Timothy Bauscher <timothy@linuxfromscratch.org> pour avoir été plus que d'une grande aide dans l'édition de ce livre.
- Mark Hymers <markh@linuxfromscratch.org> pour avoir été plus que d'une grande aide dans l'édition de ce livre.
- Marc Heerdink <marc_heerdink@softhome.net> pour avoir aussi été d'une grande aide dans l'édition de ce livre.
- DREAMWVR.COM pour leur parrainage et l'apport de diverses ressources au projet LFS et ses projets affiliés.
- Jan Niemann <jan.niemann@tu.bs.de> pour la maintenance du miroir www.de.linuxfromscratch.org.
-

Torsten Westermann <westermann@linux−provider.net> pour la maintenance du miroir lfs.linux−provider.net.

- Ian Chilton <ian@ichilton.co.uk> pour la maintenance des miroirs www.us.linuxfromscratch.org et www.linuxfromscratch.co.uk.
- Dag Stenstad <dag@stenstad.net> pour avoir fourni le miroir www.no.linuxfromscratch.org, et Ian Chilton <ian@ichilton.co.uk> pour assurer sa maintenance.
- Antonin Sprinzl <Antonin.Sprinzl@tuwien.ac.at> pour la maintenance du miroir www.at.linuxfromscratch.org.
- Jason Andrade <jason@dstc.edu.au> pour la maintenance du miroir www.au.linuxfromscratch.org.
- Ian Cooper <ian@wpi.edu> pour la maintenance du miroir www.us2.linuxfromscratch.org mirror.
- VA Linux Systems qui, au nom de Linux.com, a fait don d'une station de travail VA Linux 420 (anciennement StartX SP2) pour ce projet.
- Johan Lenglet <johan@linuxfromscratch.org> pour la direction du projet LFS de traduction française.
- Jesse Tie Ten Quee <highos@linuxfromscratch.org> pour le don d'un graveur Yamaha CDRW 8824E.
- O'Reilly pour le don de livres sur SQL et PHP.
- Robert Briggs pour le don des noms de domaines linuxfromscratch.org et linuxfromscratch.com.
- Frank Skettino <bkenoah@oswd.org> à OSWD pour la conception initiale du site web LFS.
- Garrett LeSage <garrett@linux.com> pour la création du logo LFS.
- Dean Benson <dean@vipersoft.co.uk> pour l'aide financière apportée lors de la création de l'organisation sans but lucratif LFS.
- Sans compter les autres personnes membres des diverses listes de diffusions LFS qui, par leurs suggestions, leurs tests et leurs rapports de bogues, permettent la réalisation de ce livre.

# Comment les choses se dérouleront–elles

Nous allons construire le système LFS en utilisant une distribution Linux déjà installée, telle que Debian, SuSE, Slackware, Mandrake, RedHat, etc. Nous allons utiliser le système Linux existant en tant que plateforme de développement, parce que nous avons besoin d'outils comme un compilateur, un éditeur de liens, un éditeur de texte, et d'autres outils de développement pour créer notre système. D'ordinaire, les outils requis sont disponibles par défaut si nous avons sélectionné "développement" comme option d'installation lorsque nous avons installé le système actuel.

Après avoir téléchargé les packages nécessaires à la constitution d'un système LFS, vous devrez créer une nouvelle partition native Linux et son système de fichiers. C'est là que le système LFS sera compilé et installé.

L'étape suivante, le chapitre 5, discutera de l'installation d'un certain nombre de packages formant la suite basique de développement qui a été utilisé pour construire le système actuel, ou nécessaire pour résoudre certaines dépendances circulaires. Par exemple, vous avez besoin d'un compilateur pour construire un compilateur, et vous avez besoin d'un shell pour installer un nouveau shell. Les packages de ce chapitre vont être liés statiquement.

Les liens statiques décrivent une méthode de compilation de logiciels, de façon à ce que la présence des bibliothèques ne soit pas requise. Le programme résultant est capable de fonctionner tout seul. Ce programmme est capable de le faire, parce que les pièces de ce programme, qui seraient normalement comprises dans des bibliothèques, sont copiées depuis les bibliothèques et installées dans ce programme. Habituellement, les programmes sont construits en utilisant les bibliothèques dynamiques. Ceci sauve de l'espace disque et accroît l'efficacité de beaucoup de programmes. Nous établissons des liens statiques de nos logiciels dans le chapitre 5 parce que nous allons déplacer notre système de développement dans un environnement virtuel où les bibliothèques déjà mentionnées sont absentes. Si ces logiciels étaient construits dynamiquement, notre suite de développement ne fonctionnera pas. Comme les bibliothèques ont été fournies par notre distribution Linux, le but du chapitre 5 est de construire un environnement de développement où les bibliothèques ne sont pas requises et qui donc est indépendant de la distribution.

Dans le chapitre 6, nous allons construire et installer le système final. Nous allons utiliser le programme chroot pour entrer dans un environnement virtuel et démarrer un nouveau shell dont le répertoire racine sera la partition où nous avons construit tous les logiciels du chapitre 5. Ceci est très similaire à redémarrer en indiquant au kernel de monter notre partition LFS en tant que partition root. La raison pour laquelle nous ne redémarrons pas, mais entrons dans chroot, est que créer un système statique dans lequel nous pouvons démarrer demande plus de travail, qui n'est pas nécessaire. Nous allons donc continuer à utiliser notre système où nous construisons LFS. Pendant la compilation et l'installation, vous pouvez tout simplement passer à une autre console virtuelle et continuer à utiliser votre ordinateur normalement.

Lorsque tous les logiciels du chapitre 6 seront entièrement installés, les chapitres 7, 8 et 9 nous aideront à finaliser notre installation. Nous passerons à la configuration des scripts de démarrage dans le chapitre 7. Dans le chapitre 8, nous installerons le noyau Linux et configurerons le chargeur de démarrage. Le chapitre 9 dispose d'astuces que vous pourrez mettre en oeuvre après avoir fini le livre. Enfin vous pourrez redémarrer la machine sur votre nouveau système LFS, et commencer à vraiment l'utiliser.

En résumé, telle est la démarche à suivre. Des informations détaillées sur les différentes étapes sont fournies tout au long des chapitres au fur et à mesure de votre progression. Si tout n'est pas encore clair, ne vous inquiétez pas, cela ne saurait tarder.

Veuillez lire le Chapitre 2 avec attention car il explique un certain nombre de points importants qu'il vous faut connaître avant d'en arriver au chapitre 5 et au−delà.

# Conventions utilisées dans ce livre

Pour rendre les choses faciles à comprendre, il y a un certain nombre de conventions qui sont utilisées tout au long du livre. Voici quelques exemples:

**`./configure --prefix=/usr`**

Cette façon de présenter montre les textes qui doivent être tapés exactement comme ils sont écrits, sauf si le texte autour dit le contraire. Cela est aussi utilisé dans les explications pour mettre en évidence les commandes auxquelles on fait référence.

`install-info: unknown option `--dir-file=/mnt/lfs/usr/info/dir'`

Cette façon de présenter (texte de largeur fixe) montre les textes qui apparaissent à l'écran, très souvent comme résultat à la fin d'une commande. C'est aussi utilisé pour faire ressortir les noms de fichiers comme `/etc/lilo.conf`.

*Emphasis*

Cette façon de présenter est utilisée dans ce livre pour différentes choses, principalement pour attirer l'attention sur des points importants mais aussi pour donner des exemples de ce qu'il faut taper.

http://www.linuxfromscratch.org/

Cette façon de présenter est utilisée pour les liens hypertextes internes à ce livre mais aussi externes comme les HOWTOs, les sites de téléchargement, les sites web, etc.

```
cat > $LFS/etc/group << "EOF"
root:x:0:
bin:x:1:
......
EOF
```

Cette façon de présenter est utilisée principalement lorsqu'il y a création de fichiers de configuration. La première commande (en gras) demande au système de créer le fichier $LFS/etc/group à partir des lignes qui suivent jusqu'à que la séquence EOF soit rencontrée. Par conséquent, l'ensemble de cette partie est généralement tapé comme elle est présentée.

# Version du livre

Ceci est le livre LFS version 4.0 datée du 5 octobre 2002. Si cette version a plus d'un mois, une version plus récente est probablement disponible en téléchargement. Vérifiez sur l'un de nos sites miroirs ci−dessous l'existence de versions remises à jour.

# Sites miroirs

Ci−dessous se trouve une liste de sites miroirs HTTP et FTP en date du 10 avril 2002. Cette liste peut ne plus être d'actualité. Les informations les plus récentes peuvent être trouvées sur notre site web http://www.linuxfromscratch.org.

## Miroirs HTTP

### Amérique du Nord

- Fremont, Californie, USA [8 Mbit] − http://www.linuxfromscratch.org/lfs/intro.shtml
- Lufkin, Texas, USA [6 Mbit] − http://linuxfromscratch.idge.net/lfs/intro.shtml
- Columbus, Ohio, USA [1 Mbit] − http://www.us.linuxfromscratch.org/lfs/intro.shtml

### Europe

- Mainz, Allemagne [100 Mbit] − http://lfs.linux−provider.net/lfs/intro.shtml
- Amsterdam, Hollande [100 Mbit] − http://www.nl.linuxfromscratch.org/lfs/intro.shtml
- Oslo, Norvège [100 Mbit] − http://www.no.linuxfromscratch.org/lfs/intro.shtml
- Lancaster, Royaume−Uni [100 Mbit] − http://linuxfromscratch.mirror.ac.uk/lfs/intro.shtml
- Université Technique de Vienne, Autriche [64 Mbit] − http://www.at.linuxfromscratch.org/lfs/intro.shtml
- Karlskrona, Suède [10 Mbit] − http://www.se.linuxfromscratch.org/lfs/intro.shtml
- Freising, Allemagne [4 Mbit] − http://www.de.linuxfromscratch.org/lfs/intro.shtml
- Teeside, Royaume−Uni [256 Kbit] − http://www.linuxfromscratch.co.uk/lfs/intro.shtml

### Australie

- Brisbane, Australie [155 Mbit] – http://www.au.linuxfromscratch.org/lfs/intro.shtml

## Miroirs FTP

### Amérique du Nord

- Fremont, Californie, USA [FTP] [8 Mbit] – ftp://ftp.linuxfromscratch.org
- Fremont, Californie, USA [HTTP] [8 Mbit] – http://ftp.linuxfromscratch.org
- Lufkin, Texas, USA [FTP] [6 Mbit] – ftp://linuxfromscratch.idge.net/linuxfromscratch
- Lufkin, Texas, USA [HTTP] [6 Mbit] – http://ftp.idge.net/linuxfromscratch

### Europe

- Université de Twente, Hollande [HTTP] [100 Mbit] – http://ftp.snt.utwente.nl/linux/lfs
- Université de Twente, Hollande [FTP] [100 Mbit] – ftp://ftp.snt.utwente.nl/pub/linux/lfs
- Oslo, Norvège [FTP] [100 Mbit] – ftp://ftp.no.linuxfromscratch.org/mirrors/lfs/
- Lancaster, Royaume–Uni [HTTP] [100 Mbit] – http://www.mirror.ac.uk/sites/ftp.linuxfromscratch.org
- Université Technique de Vienne, Autriche [FTP] [64 Mbit] – ftp://ftp.at.linuxfromscratch.org/pub/lfs
- Université Technique de Vienne, Autriche [HTTP] [64 Mbit] – http://ftp.at.linuxfromscratch.org/pub/lfs
- Karlskrona, Suède [FTP] [10 Mbit] – ftp://ftp.se.linuxfromscratch.org/lfs
- Freising, Allemagne [HTTP] [4 Mbit] – http://ftp.de.linuxfromscratch.org
-

Freising, Allemagne [FTP] [4 Mbit] – ftp://ftp.de.linuxfromscratch.org/mirrors/ftp.linuxfromscratch.org/

## Australie

- Brisbane, Australie [FTP] [155 Mbit] – ftp://ftp.planetmirror.com/pub/lfs/

# Journal des modifications (Changelog)

4.0 – 5 octobre 2002

- 3 octobre2002 [gerard]: Chapitre 06 – Linuxthreads: Au lieu de **`cd man`**, on utilise l'option *–C* de **`make`** qui a plus de sens (et est plus court aussi).
- 29 septembre 2002 [gerard]: Chapitre 05 – GCC: Correction du vieux correctif nofixincludes et réinstallation.
- 29 septembre 2002 [markh]: Chapitre 05 – Bash: Référence correcte à Debian pour ce qui est de l'ancienne version stable (potato) car cela ne s'applique pas à la version actuelle (woody). Rapporté par h2k1 sur #lfs.

4.0–RC1 – 28 septembre 2002

- Mise à jour vers:
  - automake–1.6.3
  - bin86–0.16.3
  - binutils–2.13
  - bison–1.35
  - diffutils–2.8.1
  - file–3.39
  - gawk–3.1.1
  - gcc–3.2
  - gettext–0.11.5
  -

groff−1.18

- gzip−1.2.4b.patch
- lfs−bootscripts−1.10
- linux−2.4.19
- MAKEDEV−1.7
- man−1.5k
- man−pages−1.52
- modutils−2.4.19
- ncurses−5.2−2.patch
- perl−5.8.0
- psmisc−21
- texinfo−4.2
- textutils−2.1
- util−linux−2.11u

- Ajout de:
  - ed−0.2.patch
  - fileutils−4.1.patch
  -

gawk−3.1.1−2.patch

- gcc−3.2.patch
- glibc−2.2.5−2.patch
- gzip−1.2.4b.patch
- kbd−1.06−3.patch
- ncurses−5.2.patch
- procps−2.0.7.patch
- sh−utils−2.0−hostname.patch
- vim−6.1.patch
- zlib−1.1.4

- Suppression de:
  - gzip−1.2.4a.patch
  - kbd−1.06−2.patch
  - reiserfsprogs−3.x.1b

- 28 septembre 2002 [gerard]: Chapitre 05 − GCC: Ajout du correctif nofixincludes pour empêcher ce script de s'exécuter dans le chapitre 5. Il doit être lancé au chapitre 6, donc nous avons besoin que ce soit un patch séparé.

- 28 septembre 2002 [gerard]: Chapitre 06 − Man: Remplacement de l'instruction sed avec un correctif régulier.

-

28 septembre 2002 [gerard]: Chapitre 06 – Bzip2: Suppression de *PREFIX=/usr* de la commande **make install** car *PREFIX* vaut déjà */usr* par défaut.

- 28 septembre 2002 [gerard]: Chapitre 06 – Vim: Suppression de la note pour la conformité avec FHS. Elle est erronée car Vim n'utilise pas du tout localstatedir.

- 28 septembre 2002 [timothy]: Application du correctif, pour les erreurs grammaticaux, de Bill Maltby. Changement de "$LFS" en "LFS" lorsque nous parlons de la variable d'environnement LFS.

- 23 septembre 2002 [timothy]: Application des correctifs de corrections grammaticaux de Bill Maltby.

- 23 septembre 2002 [timothy]: Ajout de **–** avant les options **tar** (pour clarification).

- 22 septembre 2002 [timothy]: Chapitre 06: Application du correctif de corrections grammaticaux d'Alex.

- 21 septembre 2002 [timothy]: Chapitre 02: Application du correctif de corrections grammaticaux de Bill Maltby.

- 21 septembre 2002 [timothy]: Chapitre 06 – Zlib: **mv** des bibliothèques partagées dans /lib.

- 20 septembre 2002 [timothy]: Chapitre 05 – GCC: Suppression de l'option **--enable-threads=posix** comme nous ne construisons pas de compilateur C++ dans ce chapitre.

- 18 septembre 2002 [timothy]: Chapitre 05 – Introduction: Suppression du paragraphe sur l'édition de liens statiques car il semble trompeur et est décrit avec plus détails dans *Pourquoi utiliser une édition de lien statique ?*.

- 18 septembre 2002 [timothy]: Chapitre 08 – Linux: Suppression de la commande **cd**.

- 18 septembre 2002 [timothy]: Chapitre 06 – Ncurses: Suppression de l'explication sur l'ancienne commande **mv /lib/*.a /usr/lib**.

- 13 septembre 2002 [gerard]: Chapitre 06 – Shadow: Ajout de –-libdir=/usr/lib aux options du script configure. De cette façon, un `libshadow.la` est généré correctement. Modification de la commande **mv** pour déplacer tous les fichiers `libshadow.so*` dans le répertoire `/lib`. Les fichiers `lib*a` sont déjà dans le répertoire `/usr/lib`.

-

13 septembre 2002 [gerard]: Chapitre 06 – Man: Ajout d'une autre expression régulière à la commande **sed** pour modifier le fichier `man.conf`. Cette expression régulière mets en commentaire la ligne *MANPATH /usr/man*, car elle cause des résultats dupliqués pour la commande **whatis**.

- 13 septembre 2002 [gerard]: Chapitre 06: Ajout de l'installation des *Linux Threads Man Pages* après l'installation de Perl.

- 12 septembre 2002 [gerard]: Chapitre 06 – Création du lien symbolique mtab: Remplace la commande **ln** par **ln -sf** pour que le fichier `/etc/mtab` existant, créé par la commande **mount** soit supprimé avant d'être recréé en tant que lien symbolique.

- 12 septembre 2002 [gerard]: Chapitre 06 – Sh–utils: Ajout d'un correctif sh–utils–hostname empêchant la construction du programme hostname. Ceci est fait car le programme hostname du package net–tools est supérieur à cette version.

- 12 septembre 2002 [gerard]: Chapitre 06 – Gawk: Mise à jour du correctif Gawk. Il change aussi l'emplacement du répertoire *DDEFPATH*.

- 12 septembre 2002 [gerard]: Chapitre 06 – Procps: Ajout d'un correctif pour corriger le problème de locale qui fait s'arrêter brutalement **top** avec un certain paramétrage de locale.

- 12 septembre 2002 [timothy]: Chapitre 04 – Création d'un système de fichiers: Quelques références aux alternatives dans BLFS.

- 12 septembre 2002 [gerard]: Suppression de tous les liens symboliques `/usr/lib/*.so` superflus lors des installations de bibliothèques.

- 12 septembre 2002 [gerard]: Mise à jour vers lfs–bootscripts–1.10

- 12 septembre 2002 [gerard]: Chapitre 06 – Configurer Sysvinit: Changement de la ligne sulogin en *once* à la place de respawn. De cette façon, il va se comporter comme souhaité (c'est–à–dire qu'un CTRL+D continuera au lieu de relancer sulogin).

- 12 septembre 2002 [gerard]: Chapitre 06 – GCC: Ajout de l'option *––enable–clocale=gnu* pour s'assurer que le mode locale est utilisé par les bibliothèques C++.

- 11 septembre 2002 [timothy]: Préface: Quelques corrections grammaticaux.

- 8 septembre 2002 [timothy]: Chapitre 06: Application du correctif de grammaire d'Alex.

7 septembre 2002 [timothy]: Chapitre 06 – Gzip: Ajout de gzip–1.2.4b.patch.

- 6 septembre 2002 [timothy]: Chapitre 05 – Textutils: Ajout de re_max_failures2 pour les anciens systèmes hôtes.

- 2 septembre 2002 [timothy]: Chapitre 06 – Bash: Suppression de la création du lien symbolique sh. Création des liens symboliques bash et sh: Ajout du lien symbolique /bin/bash, par lien entre sh et bash. Gzip, Sysvinit, Util–Linux: Diminution de la commande cp. Makedev: Suppression de la création et de la suppression de /bin/bash. Man: Modification du sed en éditant l'appel à less, de façon à ce que SGR fonctionne.

- 1er septembre 2002 [timothy]: Chapitre 06 – A propos des symbôles de débuggage: Suppression de l'information sur le 'strip' de /static. Man: Ajout du sed pour empêcher groff d'utiliser les séquences d'échappement SGR.

- 1er septembre 2002 [timothy]: Chapitre 05 – Installer tous les logiciels en tant qu'utilisateur non privilégié. Ajout de $CC='gcc –s' pour ne pas prendre en compte la compilation des symbôles dans les packages statiques.

- 30 août 2002 [timothy]: Chapitre 06 – Makedev: Ajout de rm /bin/bash après la création du périphérique. Perl: Suppression de l'information sur l'ancien correctif.

- 30 août 2002 [timothy]: Chapitre 05 – GCC: Rajout de HAVE_GAS_HIDDEN; suppression de ––enable–__cxa–atexit qui était incorrect et non requis dans ce chapitre; ajout de l'information sur le correctif.

- 26 août 2002 [gerard]: Ajout du nouveau correctif de Glibc et introduction d'un correctif pour GCC.

- 26 août 2002 [gerard]: Mise à jour vers automake–1.6.3, gcc–3.2, groff–1.18, makedev–1.7, perl–5.8.0, util–linux–2.11u

- 22 août 2002 [timothy]: Annexe: Ajout des URLs manquantes pour les correctifs.

- 18 août 2002 [timothy]: Chapitres 05 & 06: Modification des commandes "ln –sf" en "ln –s" à chaque fois que cela est possible.

- 18 août 2002 [timothy]: Chapitre 06 – Binutils: copie de libiberty.h après l'installation, comme ce fichier est requis par certains logiciels. Shadow: Ajout d'une commande pour supprimer le programme groups installé par Shadow, car Sh–utils en installe un (meilleur).

18 août 2002 [timothy]: Chapitre 05 – Sh–utils: Rajout du correctif sh–utils–2.0.patch.

- 16 août 2002 [markh]: Chapitre 06 – Déplacement de man–pages juste après l'installation des entêtes du noyau.

- 15 août 2002 [markh]: Chapitre 06 – Déplacement de l'installation de MAKEDEV, juste avant glibc et suppression de la création temporaire de `/dev/null` car nous n'en avons plus besoin.

- 15 août 2002 [timothy]: Chapitre 04 – Créer une nouvelle partition: Ajout d'une mention stipulant que la partition swap peut être partagée entre le système LFS et le système hôte, changement grammaticaux.

- 13 août 2002 [gerard]: Chapitre 06: Suppression de l'option *––with–curses* de l'installation de Bash car c'est inutile ici.

- 9 août 2002 [timothy]: Mise à jour vers modutils–2.4.19, linux–2.4.19, gettext–0.11.5, binutils–2.13, textutils–2.1.

- 9 août 2002 [timothy]: Chapitre 06 – Vim: Changement des liens vers des éditeurs alternatifs, des hints vers BLFS.

- 8 août 2002 [gerard]: Chapitre 06 – Ncurses: Suppression de l'option de configure *––disable–termcap*. Termcap est désactivé par défaut maintenant, donc pas besoin de cette option (laissée depuis un long moment quand elle était nécessaire).

- 8 août 2002 [gerard]: Chapitre 06 – Linux: Ajout de la commande **`cp include/asm–generic /usr/include`**. Il existe des programmes qui utilisent les fichiers qui s'y trouvent, de même que les entêtes dans le répertoire `asm` pourront être séparés dans le futur, et être mis dans `asm-generic`.

- 8 août 2002 [gerard]: Annexe A – Gettext: Ajout de la description d'un programme manquant, msgcat.

- 4 août 2002 [timothy]: Ajout de zlib–1.1.4.

- 3 août 2002 [timothy]: Mise à jour vers man–pages–1.52, man–1.5k, gettext–0.11.4, modutils–2.4.18.

- 29 juillet 2002 [timothy]: Suppression de Reiserfsprogs. Mise à jour vers util–linux–2.11t et file–3.39.

29 juillet 2002 [timothy]: Chapitres 04 & 05 – Créer une nouvelle partition, Introduction, Pourquoi utiliser une édition de lien statique ?: corrections grammaticaux. Diffutils, Fileutils, Grep, Texinfo: mise en place de LDFLAGS=−static avant configure au lieu de l'indiquer en tant qu'argument de make. GCC: Ajout de HAVE_GAS_HIDDEN pour auto−host.h.

- 29 juillet [timothy]: Chapitre 06 – Glibc: Ajout de l'option −−disable−profile.
- 29 juillet 2002 [timothy]: Chapitre 08 – Linux: Ajout d'information sur les modules et la documentation du noyau.
- 29 juillet 2002 [timothy]: Chapitre 09 – Redémarrer le système: Ajout d'une commande pour supprimer le répertoire static.
- 8 juillet 2002 [timothy]: Chapitre 09 – Redémarrer le système: Indication de BLFS comme prochaine étape.
- 3 juillet 2002 [timothy]: Chapitre 06 – Sysvinit: Simplification de la commande sed et mise à jour de la description d'installation car init affiche maintenant "Sending processes" au lieu de "Sending all processes".
- 2 juillet 2002 [markh]: Changement interne – Modification de tous les correctifs pour ajouter l'entité &package−patch−version; et suppression de toutes les versions de correctifs indiquées en dur.
- 30 juin 2002 [timothy]: Mise à jour vers man−pages−1.51 et automake−1.6.2
- 24 juin 2002 [timothy]: Chapitre 06 – Shadow, Util−linux, LFS−Bootscripts: Mise à jour du contenu des packages.
- 23 juin 2002 [timothy]: Chapitres 05 & 06 – Net−tools, Perl, Texinfo, Autoconf, Automake, File, Libtool, Bin86, Vim, Linux, Bison, Less, Man−pages, Groff, Bzip2, E2fsprogs, Grep, Lilo, Modutils, Procps, Psmisc, Reiserfsprogs: Mise à jour du contenu des packages.
- 23 juin 2002 [timothy]: Chapitres 05 & 06 – M4, Bzip2, File, E2fsprogs: Ajout de la mention de dernière vérification pour uniformiser.
- 23 juin 2002 [timothy]: Chapitres 05 & 06 – GCC: Suppression des programmes spécifiques aux 686.
- 16 juin 2002 [timothy]: Chapitre 05 & 06 – Gettext: Mise à jour du contenu du package.

14 juin 2002 [timothy]: Chapitres 05 & 06 – Binutils, Bzip2, Diffutils, Grep: Mise à jour du contenu des packages.

- 14 juin 2002 [timothy]: Chapitres 05 & 06 – GCC: Mise à jour de la description de c++filt.
- 13 juin 2002 [timothy]: Chapitre 09 – La fin: Modification de $LFS/etc/lfs–4.0 en $LFS/etc/lfs et insertion du numéro de version dans ce fichier.
- 12 juin 2002 [timothy]: Chapitre 05 – GCC: Modification des instructions de construction et des explications sur les commandes pour construire uniquement le compilateur C. Le compilateur C++ n'est pas nécessaire avant la deuxième construction de GCC.
- 12 juin 2002 [timothy]: Chapitre 06 – Shadow: Quelques corrections de grammaire.
- 11 juin 2002 [timothy]: Chapitres 05 & 06 – Gawk: Création d'une liste du contenu du package avec les descriptions.
- 11 juin 2002 [timothy]: Chapitre 05 – Fileutils: Suppression d'un paragraphe pouvant apporter de la confusion à propos du correctif de fileutils, suite à la demande de Christophe Devine.
- 11 juin 2002 [timothy]: Chapitres 05 & 06 – Tous les logiciels: Mise à jour des espaces disque requis estimés.
- 11 juin 2002 [timothy]: Chapitres 05 & 06 – GCC: Mise à jour du contenu du package et de l'espace disque requis estimé.
- 9 juin 2002 [markh]: Chapitre 06 – Création des répertoires: Modification de usr,usr/local en simplement usr/local comme nous utilisons l'option –p avec mkdir, qui créera le répertoire usr de toute façon.
- 7 juin 2002 [timothy] – Chapitre 06 – Reiserfsprogs: Ajout d'une description pour unpack.
- 7 juin 2002 [timothy] – Chapitre 02 – Comment demander de l'aide: Ajout de la mention de la FAQ.
- 6 juin 2002 [markh] – Chapitre 05 – Amélioration des explications sur le changement vers /static.
- 5 juin 2002 [timothy] Préface – Qui ne voudra pas lire ce livre: Application d'une version revu du correctif grammaticale de Scot.

5 juin 2002 [timothy] Chapitre 09 – Redémarrer le système: L'auteur du hint est nommé.

- 5 juin 2002 [timothy] Chapitre 06 – Vim: Mise à jour de l'URL de l'astuce.
- 5 juin 2002 [timothy] Chapitre 06 – LILO: L'auteur du hint est nommé.
- 5 juin 2002 [timothy] Chapitre 06 – Scripts de démarrage: L'auteur du hint est nommé.
- 5 juin 2002 [timothy] Chapitre 05 – Gawk: Ajout de la mention que le correctif sera appliqué dans le chapitre 6. Ce correctif a occasionné quelques confusions.
- 3 juin 2002 [timothy] Chapitre 01 – FAQ: Corrections de quelques erreurs.
- 31 mai 2002 [gerard] Chapitre 05 – Findutils: Ajout de la correction CPPFLAGS...re_max_failures qui est nécessaire avec les systèmes à base de Glibc–2.1.
- 30 mai 2002 [markh]: Chapitres 05 & 06 – Mise à jour vers binutils–2.12.1.
- 30 mai 2002 [markh]: Chapitre 05 – Bash: Suppression de la section sur les deux dernières commandes car nous n'avons plus besoin des commandes en question.
- 30 mai 2002 [gerard]: Chapitre 06 – Glibc: Remplacement de diférentes corrections sed avec un correctif.
- 30 mai 2002 [gerard]: Chapitre 06 – Gawk: Remplacement de la correction sed avec un correctif.
- 30 mai 2002 [gerard]: Chapitre 05 – Fileutils: Remplacement de la correction sed avec un correctif.
- 30 mai, 2002 [gerard]: Chapitre 06 – Ed: Remplacement de la correction sed avec un correctif.
- 28 mai 2002 [gerard]: Chapitre 06 – Changer le propriétaire de la partition LFS: Suppression de la commande explicite pour modifier l'utilisateur de `/lost+found`. Ceci est fait par la première commande maintenant que proc n'est plus monté dans le chapitre 5.
- 27 mai 2002 [gerard]: Mise à jour vers ncurses–5.2–2.patch (ce correctif est plus petit que celui précédemment utilisé).
-

26 mai 2002 [gerard]: Mise à jour vers: Automake−1.6.1, bin86−0.16.3, file−3.38, gawk−3.1.1, gcc−3.1, gettext−0.11.2, modutils−2.4.16, psmisc−21 et util−linux−2.11r. Ajout des correctifs de compilation gcc−3.1 pour ncurses, perl et vim.

- 26 mai 2002 [gerard]: Chapitre 05+06 − Binutils: Suppression de la configuration de la variable tooldir pour le chapitre 05−binutils, et déplacement de sa description dans le chapitre 06−binutils.

- 26 mai 2002 [gerard]: Chapitre 05 − Gawk & Findutils: simplification de l'installation en supprimant les modifications de libexecdir. Nous pouvons continuer avec la création de `$LFS/static/libexecdir`. De toute façon, tout le répertoire `$LFS/static` est temporaire, donc nous ne sommes pas autant concernés que cela par à quoi il ressemble.

- 26 mai 2002 [gerard]: Chapitre 06 − Créer les répertoires: Suppression de la commande **`cd /`** et modification des deux commandes **`chmod`** pour utiliser à la place des chemins absolus.

- 25 mai 2002 [markh]: Chapitre 06 − Quelques corrections mineures en corrélation avec la suppression de la variable $LFS où elle n'est pas voulue.

- 23 mai 2002 [gerard]: Implémentation de l'astuce LFS keep_chap5_and_chap6_sep. Points importants des changements: Ajout de findutils et util−linux dans le chapitre 5, installation de tout ce qui se trouve dans le chapitre 5 dans `$LFS/static` et nouvel ordonnancement des packages du chapitre 6 pour prévenir la sauvegarde en dur du mauvais chemin (fichiers du répertoire `$LFS/static`).

- 23 mai 2002 [gerard]: Annexe A − E2fsprogs: Ajout de quelques descriptions supplémentaires.

- 23 mai 2002 [gerard]: Annexe A − Bin86: Ajout de quelques descriptions.

- 23 mai 2002 [gerard]: Annexe A − Flex: Ajout de quelques descriptions.

- 23 mai 2002 [gerard]: Annexe A − Glibc: Ajout de quelques descriptions supplémentaires.

- 18 mai 2002 [gerard]: Annexe A − E2fsprogs: Ajout de quelques descriptions supplémentaires.

- 18 mai 2002 [gerard]: Annexe A − Glibc: Ajout de quelques descriptions supplémentaires.

- 17 mai 2002 [markh]: Changement de tous les chown X.X en chown X:X, qui risquent moins de poser problème (d'après les informations contenues dans la page info de chown).

-

16 mai 2002 [gerard]: Chapitre 01 – Sites miroirs: Ajout d'une interface http au miroir FTP à idge.net

- 16 mai 2002 [gerard]: Annexe A – Glibc: Ajout de quelques descriptions supplémentaires.

- 15 mai 2002 [markh]: Chapitre 05 – Bzip2. Modification des instructions pour corriger le problème des liens avec les anciennes distributions, de manière identique aux instructions pour gzip au chapitre 5.

- 11 mai 2002 [markh]: De nombreuses corrections XML; principalement des modifications des tags <ulink> pour supprimer les  erronés dans la sortie HTML.

- 9 mai 2002 [gerard]: Annexe A – Glibc: Ajout des descriptions manquantes.

- 6 mai 2002 [gerard]: Chapitre 06 – Shadow: Correction du lien symbolique de `vigr` vers `/usr/sbin`

- 2 mai 2002 [gerard]: Chapitre 06 – Procps: Remplacement des deux simple cotes par deux doubles cotes (les simples cotes peuvent être confondus pour une double cote, ce qui causerait une erreur).

- 2 mai 2002 [gerard]: Remplacement des commandes **`cd dir && ln -sf`** par une simple commande (comme **`ln -sf bash $LFS/bin/sh`**). De la même façon, les constructions du type **`cd dir && mv/cp`** sont remplacées par des constructions du type **`mv $LFS/usr/bin/{bzcat,bzip2} $LFS/bin`**

- 2 mai 2002 [markh]: Suppression de la section "Suppression des anciennes bibliothèques NSS".

- 1er mai 2002 [gerard]: Suppression de toutes les manipulations pour Glibc–2.0 – correctif gzip, correctif sh–utils, copie des fichiers libnss. Suppression des constructions du type **`export VAR=VALUE...unset VAR`** et modification en **`VAR=VALUE ./configure`**.

- 26 avril 2002 [marcheerdink]: Chapitre 06 Findutils: Ajout de libexecdir=/usr/bin à la commande make pour corriger un mauvais chemin pour libexecdir dans updatedb.

- 25 avril 2002 [gerard]: Chapitre 06 Glibc: Ajout d'une note qui indique que si vous voulez installer manuellement certains locales, au lieu de tous, alors vous avez d'abord besoin de créer le répertoire `/usr/lib/locale`.

- 21 avril 2002 [gerard & markh]: Mise à jour vers MAKEDEV–1.5

-

12 avril 2002 [markh]: Ajout du répertoire entities/ au cvs et éclatement du fichier index.xml.

- 10 avril 2002 [marcheerdink]: Mise à jour vers les packages suivants: bison−1.35, diffutils−2.8.1, texinfo−4.2, util−linux−2.11q.

- 9 avril 2002 [marcheerdink]: Ajout de −−disable−perl−regexp aux options du script configure de grep pour éviter qu'il soit lié avec une bibliothèque statique pcre inexistante.

- 8 avril 2002 [gerard]: Ajout du miroir http://ftp.de.linuxfromscratch.org (en complément de ftp://ftp.de).

# Listes de diffusion et archives

Le serveur linuxfromscratch.org accueille les listes de diffusion publiques suivantes:

- lfs−support
- lfs−dev
- lfs−announce
- lfs−security
- lfs−book
- lfs−chat
- alfs−discuss
- blfs−dev
- blfs−book
- blfs−support

## lfs−support

La liste de diffusion "lfs−support" apporte une aide aux utilisateurs construisant un système LFS, aide s'arrêtant à la fin du livre principal. Les demandes d'aide pour installer un programme au−delà de ce point doivent être adressées à la liste "blfs−support".

## lfs−dev

La liste de diffusion "lfs−dev" traite uniquement de sujets en rapport avec le livre LFS. Si le livre laisse apparaître des problèmes, qu'un bogue ou deux doivent être rapportés, ou que des suggestions pour l'amélioration du livre doivent être faites, cette liste de diffusion est la bonne.

Les demandes d'aide doivent être adressées à "lfs−support" ou à "blfs−support".

## lfs−announce

La liste lfs−announce est une liste restreinte. Vous pouvez y souscrire, mais vous ne pouvez pas y poster de messages. Cette liste sert à annoncer les nouvelles versions stables. La liste "lfs−dev", quant à elle, annonce aussi les nouvelles versions de développement. Il n'est pas utile pour un utilisateur de s'inscrire à la liste "lfs−announce" si il est déjà inscrit à "lfs−dev", car tout ce qui est posté sur la première l'est aussi sur la seconde.

## lfs−security

La liste de diffusion "lfs−security" traite des sujets touchant à la sécurité. Les préoccupations concernant la sécurité ou les problèmes de sécurité avec un package utilisé par LFS doivent être adressés à cette liste.

## lfs−book

La liste "lfs−book" est utilisée par les éditeurs du livre pour coordonner la maintenance du livre LFS, comme des problèmes avec XML et d'autres. Les discussions du moment à propos de ce qui doivent être ajoutées ou supprimées se trouvent dans la liste "lfs−dev".

## lfs−chat

La liste "lfs−chat" est un endroit destiné aux membres de la communauté LFS (ce qui vous inclut) pour parler de tout. Cela n'a même pas besoin de concerner l'informatique. Tout peut y être, rien n'est hors de propos.

## alfs−discuss

La liste de diffusion "alfs−discuss" traite du développement de ALFS, qui est l'acronyme de "Automated Linux From Scratch". Le but de ce projet est de développer un outil d'installation qui permettrait d'installer un système LFS de façon automatique et rapide, l'accelération de la compilation venant par la suppression de la nécessité d'entrer toutes les commandes manuellement.

## blfs−dev

La liste de diffusion "blfs−dev" s'occupe du développement du livre BLFS (Beyond LFS, c'est−à−dire après LFS). Cette liste de diffusion permet l'envoi de rapports de bugs ou de suggestions visant à améliorer le livre BLFS.

Toutes demandes d'aide concernant l'installation ou la configuration de programmes non inclus dans LFS doivent être postées sur la liste de diffusion blfs−support.

## blfs−book

La liste de diffusion "blfs−book" est utilisée par les éditeurs du livre BLFS pour coordonner la maintenance du livre BLFS, par exemple pour des problèmes en relation avec XML. Les discussions sur ce qui doit être ajouté ou supprimé doivent avoir lieu sur "blfs−dev".

## blfs−support

La liste de diffusion "blfs−support" gère les demandes d'assistance pour tout logiciel installé mais non inclus dans LFS. Tout logiciel installé après LFS y a sa place.

## Archives des listes de diffusion

Toutes ces listes de diffusion sont archivées et peuvent être consultées en ligne à l'adresse http://archive.linuxfromscratch.org/mail−archives ou téléchargées depuis http://ftp.linuxfromscratch.org/mail−archives ou ftp://ftp.linuxfromscratch.org/mail−archives.

## Comment envoyer un message sur une liste

Vous n'avez pas besoin de souscrire à une liste de diffusion pour pouvoir envoyer un message sur celle−ci. Néanmoins, si vous envoyez un message à une liste où vous n'êtes pas inscrit, indiquez−le dans votre message, de manière à ce que les membres de la liste puissent vous inclure dans l'en−tête en tant que Copie Conforme, pour que vous receviez votre réponse.

L'adresse d'envoi pour une liste est de la forme *nom−de−la−liste@linuxfromscratch.org*, où *nom−de−la−liste* peut être une des listes indiquées dans la section Listes disponibles ci−dessus. Voici quelques exemples d'adresses d'envoi: *lfs−dev@linuxfromscratch.org, lfs−support@linuxfromscratch.org et blfs−support@linuxfromscratch.org*.

## Comment s'abonner ?

Il est possible de s'abonner à chacune des listes de diffusion indiquées ci−dessus en envoyant un email à listar@linuxfromscratch.org et en tapant *subscribe nom−de−la−liste* dans le champ Objet de l'en−tête du message.

Il est possible de s'abonner à plusieurs listes à la fois en un seul email. Ceci peut être fait en laissant le champ Objet vide et en mettant toutes les commandes dans le corps de l'email. Cet email peut ressembler à :

> To: listar@linuxfromscratch.org Subject: subscribe lfs−dev subscribe blfs−support subscribe alfs−discuss

Après que l'email ait été envoyé, le programme Listar répondra avec un email de demande de confirmation d'abonnement. Après que cette confirmation ait été renvoyée, Listar retournera un autre email avec un message indiquant l'abonnement de l'utilisateur à la/les liste(s) ainsi qu'un message d'introduction pour chacune d'entre elles.

## Comment se désabonner

Pour se désabonner d'une liste, envoyer un email à listar@linuxfromscratch.org et taper *unsubscribe nom−de−la−liste* dans le champ Objet de l'en−tête du message.

Il est possible de se désabonner de plusieurs listes à la fois en un seul email. Ceci peut être fait en laissant le champ Objet vide et en mettant toutes les commandes dans le corps de l'email. Cet email peut ressembler à:

> To: listar@linuxfromscratch.org Subject: unsubscribe lfs−dev unsubscribe blfs−support unsubscribe alfs−discuss

Après que l'email ait été envoyé, le programme Listar répondra avec un email de demande de confirmation de désabonnement. Après que cette confirmation ait été renvoyée, Listar retournera un autre email avec un message confirmant le désabonnement de l'utilisateur de la ou des listes.

## Autres modes de listes

Les modes pouvant être activés par l'utilisateur requiert l'envoi d'un message à listar@linuxfromscratch.org. Les modes en eux−même sont activés en tapant les commandes appropriées dans le champ Sujet du message.

Comme les noms l'indiquent, le *Set command* indique quoi écrire pour activer un mode. La commande *Unset command* indique quoi écrire pour désactiver un mode.

Le mot "nom−de−la−liste" dans l'exemple de champ Objet doit être remplacé avec le nom de la liste pour laquelle le mode doit être activé. Si plus d'un mode est à activer (pour une unique liste ou un ensemble d'entre elles) en un message, ceci peut être réalisé en laissant le champ Sujet vide et en tapant les commandes dans le corps du message.

## Résumé

Commande d'activation: *set nom−de−la−liste digest* Commande de désactivation: *unset nom−de−la−liste digest*

Le mode "Digest" peut être appliqué à toutes les listes pour lesquelles l'utilisateur s'est abonné. Le mode Digest vous empêche de recevoir les messages à chaque fois qu'ils sont postés sur la liste et vous fait envoyer un unique message par jour contenant tous les messages postés sur la liste durant cette journée.

Le second mode Digest est appellé "Digest2". Si l'utilisateur active ce mode, il recevra le résumé quotidien ainsi que chacun des messages dès qu'ils sont postés. Pour activer ce mode, remplacez *digest* par *digest2* dans la commande.

## En vacances

Commande d'activation: *set nom−de−la−liste vacation* Commande de désactivation: *unset nom−de−la−liste vacation*

Lorsqu'un utilisateur est en déplacement pour un certain temps et qu'il ne souhaite plus recevoir de message sans toutefois être désabonné de la liste, il peut basculer vers le mode "vacation". Cela revient à se désabonner, sans avoir à le faire et sans avoir à se réabonner de nouveau par la suite.

# Serveur News

Toutes les listes de diffusion hébergées par linuxfromscratch.org sont aussi accessibles via le serveur NNTP. Tous les messages postés sur la liste de diffusion sont copiés vers le newsgroup correspondant, et vice versa.

Le serveur news peut être accédé par *news.linuxfromscratch.org*.

# FAQ

Si vous rencontrez des erreurs, si vous vous posez des questions, ou si vous avez trouvé une erreur typographique dans ce livre, alors, s'il vous plait, consultez la page de la FAQ (Frequently Asked Questions ou Questions Posées Fréquemment).

http://www.linuxfromscratch.org/faq/

# Contacts et informations

Envoyez de préférence tous vos emails vers l'une des listes de diffusion. Voir Chapter 1 – Listes de diffusion et archives pour plus d'informations sur les listes de diffusion disponibles.

Si vous avez besoin de joindre Gerard Beekmans personnellement, envoyez un email à gerard@linuxfromscratch.org.

# Chapitre 2. Informations importantes

# A propos de $LFS

S'il vous plaît, lisez ce paragraphe attentivement : tout au long de ce livre, la variable LFS sera utilisée fréquemment. Vous devrez la remplacer partout où vous la trouverez par le répertoire dans lequel vous avez monté la partition contenant votre système LFS. La méthode pour créer et l'endroit où monter cette partition seront expliqués en détail au chapitre 4. A titre d'exemple, supposons que la partition LFS soit montée dans le répertoire /mnt/lfs.

Quand il vous est demandé de lancer une commande telle que **`./configure --prefix=$LFS/static`**, il vous faut en pratique exécuter **`./configure --prefix=/mnt/lfs/static`**.

Il est très important que cela soit fait quel que soit l'endroit où vous le lisez, que ce soit pour une commande à donner à un shell ou dans un fichier édité ou créé.

Une solution possible est de définir la variable d'environnement LFS. De cette façon, $LFS peut être tapé directement au lieu de le remplacer par /mnt/lfs. Cela peut être accomplit en exécutant la commande suivante:

```
export LFS=/mnt/lfs
```

A partir de ce moment, lorsqu'il vous sera demandé d'entrer une commande telle que **`./configure --prefix=$LFS/static`**,vous pourrez la taper littéralement. Votre shell fera le remplacement de $LFS par /mnt/lfs lors de l'analyse de la ligne de commande (c'est–à–dire après avoir appuyé sur la touche Entrée).

Si vous décidez d'utiliser cette méthode pour $LFS, n'oubliez pas de définir cette variable à chaque fois. Si la variable n'a pas été définie mais qu'elle est utilisée dans une commande, $LFS sera remplacée par rien, et la commande résultante sera tout de même exécutée. Une commande telle que **`echo "root:x:0:0:root:/root:/bin/bash" > $LFS/etc/passwd`** avec la variable non définie va effacer et re–créer le fichier /etc/passwd de votre système hôte. En clair : vous aurez perdu la base des utilisateurs sur votre machine !

Une bonne méthode pour s'assurer que la variable LFS est définie à chaque fois consiste à la définir dans le fichier /root/.bash_profile et le fichier /root/.bashrc afin qu'à chacune de vos connexions (ou sessions par **`su`**) avec le compte root, la variable LFS soit définie.

# A propos des SBU

SBU est un acronyme pour *Static Bash Units* (c'est−à−dire Unité Statique de Bash). Elles sont notre moyen d'identifier le temps que prends un package pour être compilé. Pourquoi n'utilisons−nous pas le temps normal comme tout le monde ?

Le plus gros problème est que l'indication de temps n'est pas précis. Un grand nombre de personnes installe LFS sur un grand nombre de systèmes différents, le temps que prend la compilation d'un package varie de manière trop importante. Un package peut prendre 20 minutes sur un système, mais trois jours sur un autre (ceci n'est pas une exagération). Donc, à la place, nous allons utiliser *Static Bash Unit* ou *SBU*.

Cela fonctionne ainsi: le tout premier package compilé dans ce livre est Bash dans le chapitre 5 et il sera lié statiquement. Le temps pris pour la compilation de ce package sera la base et composera un SBU. Tous les autres temps de compilation sont relatifs au temps qu'il a fallu pour compiler Bash. Par exemple, GCC−3.2 prends environ 9,5 SBUs et il est prouvé que ce nombre est assez régulier parmi les différents systèmes. Donc, multipliez 9,5 par le nombre de secondes que Bash a pris pour s'installer (la valeur de SBU) et vous obtiendrez une approximation assez proche du temps que GCC prendra pour être installé sur votre système.

Note: les SBUs ne fonctionnent pas sur les machines multi−processeurs (SMP). Tout cela n'est donc pas utilisable si vous êtes assez chanceux pour avoir une machine multi−processeur.

# Où stocker les logiciels téléchargés

Tout au long de ce document, nous supposerons que vous avez placé les packages téléchargés dans le répertoire $LFS/usr/src.

Peu importe l'endroit où vous enregistrez les packages. Malgré tout, nous recommandons de les conserver au moins sur la partition LFS. Ceci semble logique car vous aurez encore besoin d'y accéder après avoir redémarré sur votre système LFS. $LFS/usr/src est un endroit logique pour stocker les code source, mais ce n'est pas une obligation. Vous pourriez même créer un sous−répertoire sous $LFS/usrc/src pour stocker les archives tar. De cette façon, vous pouvez séparer les archives tar des répertoires temporaires créés pour la compilation et l'installation des différents packages. Mais, encore une fois, ceci dépend de vous.

Le prochain chapitre contient une liste de tous les packages devant être téléchargés. La partition LFS n'est pas encore créée, vous ne pouvez donc pas les stocker dedans. Sauvegardez−les ailleurs pour l'instant, et quand la partition LFS sera créée, déplacez−les.

# Comment installer les logiciels

Avant que vous ne commenciez à utiliser le livre LFS, nous aimerions vous faire remarquer que toutes les commandes employées supposent que vous utilisiez le shell bash. Si cela n'est pas le cas, les commandes peuvent fonctionner, mais nous ne pouvons le garantir. Si vous voulez vous simplifier la vie, utilisez bash.

Avant de pouvoir faire quelque chose d'un package, vous devrez d'abord le déballer. Souvent, les packages sont des archives tar qui ont été compressées avec gzip ou bzip2. Nous n'expliquerons pas à chaque fois comment déballer une archive. Nous allons vous l'expliquer une bonne fois pour toutes dans cette section.

Pour commencer, placez vous dans le répertoire $LFS/usr/src en lançant :

```
cd $LFS/usr/src
```

Si une archive est au format tar et gzip, elle est déballée grâce à l'une de ces deux commandes, suivant le nom du fichier :

```
tar -xvzf filename.tar.gz
tar -xvzf filename.tgz
```

Si une archive est au format tar et bzip2, elle est déballée par la commande :

```
bzcat filename.tar.bz2 | tar -xv
```

De nos jours, la plupart des programmes tar, mais pas tous, disposent du correctif les permettant d'utiliser les fichiers bzip2 directement. Ils utilisent soit le paramètre –I, soit le –y, soit le –j, qui fonctionne de la même manière que le paramètre –z concernant la gestion des fichiers gzip. Les commandes ci–dessus fonctionnent même si le correctif n'a pas été appliqué à tar.

Si une archive est simplement au format tar, elle est déballée par la commande :

```
tar -xvf filename.tar
```

Quand une archive est déballée, un nouveau répertoire est créé dans le répertoire courant (et ce livre suppose que les archives sont déballées dans le répertoire $LFS/usr/src). Placez vous dans ce nouveau répertoire pour la suite de la procédure d'installation. Encore une fois, à chaque fois que ce livre vous expliquera l'installation d'un package, ce sera à vous de déballer l'archive des sources et de vous placer dans le répertoire créé.

De temps en temps, il vous faudra travailler avec de simple fichiers tels que des fichiers de correctif. Ces fichiers vous seront généralement fournis compressés avec gzip ou bzip2. Avant de pouvoir les utiliser, il faudra les décompresser.

Si un fichier a été compressé avec gzip, il sera décompressé par :

```
gunzip filename.gz
```

Si un fichier a été compressé avec bzip2, il sera décompressé par :

```
bunzip2 filename.bz2
```

Après l'installation d'un package, deux choix s'offrent à vous : soit effacer le répertoire contenant les sources, soit le garder. Nous vous recommendons vivement de le supprimer. Si vous ne le faites pas et essayez d'utiliser de nouveau les sources plus tard (par exemple, dans le chapitre 6, ré–utiliser les sources compilés lors du chapitre 5), il est possible que cela ne fonctionne pas de la manière attendue. En effet, les sources utilisées au chapitre 5 auront des paramètres de configuration propres à la distribution hôte que vous avez utilisée, qui ne sont pas toujours applicables au système LFS après être rentré dans l'environnement chroot.

Même l'utilisation de la commande *make clean* ne garantit pas toujours le nettoyage complet des sources.

Pensez donc à vous éviter nombre d'ennuis, et supprimez simplement le répertoire contenant les sources après installation, mais gardez l'archive tar disponible pour quand vous en aurez encore besoin.

Il est cependant un cas d'exception à cette règle, c'est celui des sources du noyau. Gardez les dans un coin, car vous en aurez besoin plus tard pour compiler un noyau. Aucun package n'utilise les sources du noyau, donc celles−ci ne seront pas un obstacle. Si, néanmoins, vous manquez d'espace disque, vous pouvez les supprimer et les re−déballez plus tard lorsqu'elles seront nécessaires.

# Pour quelle plateforme?

LFS essaie d'être autant que possible indépendant de la plateforme. Ceci dit, les développements principaux de LFS se font sur la plateforme x86. Nous essayons d'inclure si possible des informations sur les différences avec d'autres plateformes comme PPC. Si vous découvrez un problème de compilation non lié à l'architecture x86, n'hésitez pas à demander de l'aide sur les listes de diffusion. Mieux encore, si vous trouvez une solution à un problème particulier dû à une certaine plateforme, veuillez nous le faire savoir sur la liste de diffusion lfs−dev. Nous pourrons par la suite l'inclure dans le livre après vérification de son bon fonctionnement.

# Comment demander de l'aide

Si vous rencontrez des problèmes en utilisant ce livre et que votre problème ne figure pas dans la FAQ, vous vous rendrez compte que les personnes sur IRC et sur les listes de discussion sont prêtes à vous aider. Un survol des listes de diffusion de LFS est disponible sur cette page Chapter 1 – Listes de diffusion et archives. Pour nous aider à diagnostiquer et résoudre votre problème, assurez–vous d'inclure autant d'informations utiles que possible dans votre demande d'aide.

## Information de base

A part une brève explication de votre problème, les informations essentielles à inclure dans votre requête sont:

- la version du livre que vous utilisez, c'est–à–dire 4.0,
- avec quel package ou section vous avez un problème,
- quel est le message d'erreur exact, ou quel est le symptôme,
- si vous vous êtes éloigné du livre ou non.

(Notez que dire que vous vous êtes éloigné du livre ne signifie pas que nous ne vous aiderons pas. Après tout, LFS est justement un ensemble de choix. Cela nous aidera simplement à voir les autres causes possibles à votre problème.)

## Problèmes de configuration

Quand quelque chose se passe mal pendant l'étape où le script configure est lancé, regardez dans les dernières lignes du fichier `config.log`. Il contient les erreurs possibles qui ont été rencontrées pendant la phase de configuration, et qui ne sont pas toujours affichées à l'écran. Incluez les lignes appropriées si vous décidez de demander de l'aide.

## Problèmes de compilation

Pour nous aider à trouver la cause du problème, à la fois les textes affichés à l'écran et le contenu de plusieurs fichiers sont utiles. Les informations affichées par à la fois le script ./configure et la commande make peuvent être utiles. N'incluez pas aveuglément l'ensemble, mais d'un autre côté n'en donnez pas trop peu. A titre d'exemple, voici l'affichage d'une commande make :

```
gcc -DALIASPATH=\"/mnt/lfs/usr/share/locale:.\"
-DLOCALEDIR=\"/mnt/lfs/usr/share/locale\" -DLIBDIR=\"/mnt/lfs/usr/lib\"
-DINCLUDEDIR=\"/mnt/lfs/usr/include\" -DHAVE_CONFIG_H -I. -I.
-g -O2 -c getopt1.c
gcc  -g -O2 -static -o make  ar.o arscan.o commands.o dir.o expand.o file.o
function.o getopt.o implicit.o job.o main.o misc.o read.o remake.o rule.o
signame.o variable.o vpath.o default.o remote-stub.o version.o opt1.o
```

```
-lutil job.o: In function `load_too_high':
/lfs/tmp/make-3.79.1/job.c:1565: undefined reference to `getloadavg'
collect2: ld returned 1 exit status
make[2]: *** [make] Error 1
make[2]: Leaving directory `/lfs/tmp/make-3.79.1'
make[1]: *** [all-recursive] Error 1
make[1]: Leaving directory `/lfs/tmp/make-3.79.1'
make: *** [all-recursive-am] Error 2
```

Dans ce genre de cas, nombreuses sont les personnes qui ne fournissent que la section du bas contenant le texte

```
make [2]: *** [make] Error 1
```

jusqu'en bas. Cela n'est pas suffisant pour émettre un diagnostique sur le problème, car cela nous signale que *quelque chose* s'est mal passé, mais pas *ce qui* s'est mal passé. La section complète, comme fournie ci–dessus, constitue ce qui doit être inclus pour nous être utile, car il inclut la commande qui a été exécutée avec son message d'erreur.

Un excellent article sur la façon de demander de l'aide sur Internet en général a été écrit par Eric S. Raymond. Il est disponible en ligne à l'adresse http://www.tuxedo.org/~esr/faqs/smart–questions.html. Lisez et suivez les astuces indiquées dans ce document et vous aurez plus de chances de recevoir une réponse pour commencer mais aussi d'avoir l'aide dont vous avez actuellement besoin.

# II. Deuxième partie – Installation du système LFS

***Table des matières***

*3. Packages qui doivent être téléchargés*
*4. Préparer une nouvelle partition*
*5. Préparer le système LFS*
*6. Installation des logiciels de base du système*
*7. Mise en place des scripts de démarrage*
*8. Rendre le système LFS démarrable*
*9. La fin*

# Chapitre 3. Packages qui doivent être téléchargés

# Introduction

Ci−dessous, se trouve une liste des packages à télécharger pour construire le système Linux de base. Les numéros de version affichés correspondent à des versions de logiciel qui fonctionnent, et qui sont la base du livre.

Toutes les URLs ci−dessous appartiennent au serveur LFS principal. Nous avons quelques miroirs FTP disponibles, à partir desquels vous pouvez aussi télécharger les fichiers. Les adresses des sites miroirs peuvent être trouvé au Chapitre 1 − Sites miroirrs.

L'archive FTP LFS contient seulement les versions des packages utilisées dans ce livre. Vous pouvez vérifier sur les sites de téléchargement officiels disponible sur Annexe A pour déterminer si il existe ou non un package pls récent. Si vous téléchargez un package plus récent, nous aimerions savoir si vous avez été capable d'installer le package sans problèmes en utilisant les instructions de ce livre.

# Packages nécessitant d'être téléchargés

Explorer avec FTP: ftp://ftp.linuxfromscratch.org/ Explorer avec HTTP: http://ftp.linuxfromscratch.org/
Vous pouvez soit télécharger une archive tar qui contient tous les packages
utilisés pour compiler un système LFS: Tous les packages LFS – 105.560 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/4.0/lfs–packages–4.0.tar
http://ftp.linuxfromscratch.org/lfs–packages/4.0/lfs–packages–4.0.tar
Soit télécharger individuellement les packages suivants: Bash (2.05a) – 1.400 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/4.0/bash–2.05a.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/4.0/bash–2.05a.tar.bz2 Binutils (2.13) – 9.651 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/4.0/binutils–2.13.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/4.0/binutils–2.13.tar.bz2 Bzip2 (1.0.2) – 610 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/4.0/bzip2–1.0.2.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/4.0/bzip2–1.0.2.tar.bz2 Diffutils (2.8.1) – 642 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/4.0/diffutils–2.8.1.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/4.0/diffutils–2.8.1.tar.bz2 Fileutils (4.1) – 1.217 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/4.0/fileutils–4.1.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/4.0/fileutils–4.1.tar.bz2 Correctif Fileutils (4.1) – 1 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/4.0/fileutils–4.1.patch.bz2
http://ftp.linuxfromscratch.org/lfs–packages/4.0/fileutils–4.1.patch.bz2 GCC (3.2) – 20.043 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/4.0/gcc–3.2.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/4.0/gcc–3.2.tar.bz2 Correctif GCC (3.2) – 4 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/4.0/gcc–3.2.patch.bz2
http://ftp.linuxfromscratch.org/lfs–packages/4.0/gcc–3.2.patch.bz2
Correctif nofixincludes GCC (3.2) – 1 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/4.0/gcc–3.2–nofixincludes–2.patch.bz2
http://ftp.linuxfromscratch.org/lfs–packages/4.0/gcc–3.2–nofixincludes–2.patch.bz2
Linux (2.4.19) – 25.432 Ko: ftp://ftp.linuxfromscratch.org/lfs–packages/4.0/linux–2.4.19.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/4.0/linux–2.4.19.tar.bz2 Grep (2.5) – 545 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/4.0/grep–2.5.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/4.0/grep–2.5.tar.bz2 Gzip (1.2.4a) – 179 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/4.0/gzip–1.2.4a.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/4.0/gzip–1.2.4a.tar.bz2 Correctif Gzip (1.2.4b) – 1 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/4.0/gzip–1.2.4b.patch.bz2
http://ftp.linuxfromscratch.org/lfs–packages/4.0/gzip–1.2.4b.patch.bz2 Make (3.79.1) – 794 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/4.0/make–3.79.1.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/4.0/make–3.79.1.tar.bz2 Sed (3.02) – 221 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/4.0/sed–3.02.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/4.0/sed–3.02.tar.bz2 Sh–utils (2.0) – 824 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/4.0/sh–utils–2.0.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/4.0/sh–utils–2.0.tar.bz2
Correctif Hostname Sh–utils (2.0–hostname) – 1 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/4.0/sh–utils–2.0–hostname.patch.bz2
http://ftp.linuxfromscratch.org/lfs–packages/4.0/sh–utils–2.0–hostname.patch.bz2
Correctif Sh–utils (2.0) – 1 Ko: ftp://ftp.linuxfromscratch.org/lfs–packages/4.0/sh–utils–2.0.patch.bz2
http://ftp.linuxfromscratch.org/lfs–packages/4.0/sh–utils–2.0.patch.bz2 Tar (1.13) – 730 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/4.0/tar–1.13.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/4.0/tar–1.13.tar.bz2 Correctif Tar (1.13) – 1 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/4.0/tar–1.13.patch.bz2
http://ftp.linuxfromscratch.org/lfs–packages/4.0/tar–1.13.patch.bz2 Textutils (2.1) – 1.847 Ko:

ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/textutils−2.1.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/4.0/textutils−2.1.tar.bz2 Gawk (3.1.1) − 1.420 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/gawk−3.1.1.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/4.0/gawk−3.1.1.tar.bz2 Correctif Gawk (3.1.1−2) − 1 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/gawk−3.1.1−2.patch.bz2
http://ftp.linuxfromscratch.org/lfs−packages/4.0/gawk−3.1.1−2.patch.bz2 Texinfo (4.2) − 1.175 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/texinfo−4.2.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/4.0/texinfo−4.2.tar.bz2 Patch (2.5.4) − 149 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/patch−2.5.4.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/4.0/patch−2.5.4.tar.bz2 MAKEDEV (1.7) − 8 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/MAKEDEV−1.7.bz2
http://ftp.linuxfromscratch.org/lfs−packages/4.0/MAKEDEV−1.7.bz2 Glibc (2.2.5) − 12.114 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/glibc−2.2.5.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/4.0/glibc−2.2.5.tar.bz2 Glibc−linuxthreads (2.2.5) − 164 KB:
ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/glibc−linuxthreads−2.2.5.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/4.0/glibc−linuxthreads−2.2.5.tar.bz2
Correctif Glibc (2.2.5−2) − 8 Ko: ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/glibc−2.2.5−2.patch.bz2
http://ftp.linuxfromscratch.org/lfs−packages/4.0/glibc−2.2.5−2.patch.bz2 Zlib (1.1.4) − 144 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/zlib−1.1.4.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/4.0/zlib−1.1.4.tar.bz2 Man−pages (1.52) − 569 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/man−pages−1.52.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/4.0/man−pages−1.52.tar.bz2 Ed (0.2) − 158 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/ed−0.2.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/4.0/ed−0.2.tar.bz2 Correctif Ed (0.2) − 1 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/ed−0.2.patch.bz2
http://ftp.linuxfromscratch.org/lfs−packages/4.0/ed−0.2.patch.bz2 Findutils (4.1) − 226 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/findutils−4.1.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/4.0/findutils−4.1.tar.bz2 Correctif Findutils (4.1) − 1 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/findutils−4.1.patch.bz2
http://ftp.linuxfromscratch.org/lfs−packages/4.0/findutils−4.1.patch.bz2 Ncurses (5.2) − 1.308 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/ncurses−5.2.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/4.0/ncurses−5.2.tar.bz2 Correctif Ncurses (5.2−2) − 1 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/ncurses−5.2−2.patch.bz2
http://ftp.linuxfromscratch.org/lfs−packages/4.0/ncurses−5.2−2.patch.bz2 Vim (6.1) − 2.823 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/vim−6.1.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/4.0/vim−6.1.tar.bz2 Correctif Vim (6.1) − 1 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/vim−6.1.patch.bz2
http://ftp.linuxfromscratch.org/lfs−packages/4.0/vim−6.1.patch.bz2 Bison (1.35) − 613 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/bison−1.35.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/4.0/bison−1.35.tar.bz2 Less (374) − 189 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/less−374.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/4.0/less−374.tar.bz2 Groff (1.18) − 1.739 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/groff−1.18.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/4.0/groff−1.18.tar.bz2 Man (1.5k) − 168 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/man−1.5k.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/4.0/man−1.5k.tar.bz2 Correctif Man (1.5k) − 1 KB
ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/man−1.5k.patch.bz2
http://ftp.linuxfromscratch.org/lfs−packages/4.0/man−1.5k.patch.bz2 Perl (5.8.0) − 8.416 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/perl−5.8.0.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/4.0/perl−5.8.0.tar.bz2 M4 (1.4) − 249 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/m4−1.4.tar.bz2

http://ftp.linuxfromscratch.org/lfs-packages/4.0/m4-1.4.tar.bz2 Autoconf (2.53) - 739 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/4.0/autoconf-2.53.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/4.0/autoconf-2.53.tar.bz2 Automake (1.6.3) - 465 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/4.0/automake-1.6.3.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/4.0/automake-1.6.3.tar.bz2 Flex (2.5.4a) - 278 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/4.0/flex-2.5.4a.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/4.0/flex-2.5.4a.tar.bz2 File (3.39) - 151 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/4.0/file-3.39.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/4.0/file-3.39.tar.bz2 Libtool (1.4.2) - 653 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/4.0/libtool-1.4.2.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/4.0/libtool-1.4.2.tar.bz2 Bin86 (0.16.3) - 113 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/4.0/bin86-0.16.3.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/4.0/bin86-0.16.3.tar.bz2 Gettext (0.11.5) - 2.489 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/4.0/gettext-0.11.5.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/4.0/gettext-0.11.5.tar.bz2 Kbd (1.06) - 559 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/4.0/kbd-1.06.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/4.0/kbd-1.06.tar.bz2 Correctif Kbd (1.06-3) - 3 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/4.0/kbd-1.06-3.patch.bz2
http://ftp.linuxfromscratch.org/lfs-packages/4.0/kbd-1.06-3.patch.bz2 E2fsprogs (1.27) - 1.176 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/4.0/e2fsprogs-1.27.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/4.0/e2fsprogs-1.27.tar.bz2 Lilo (22.2) - 292 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/4.0/lilo-22.2.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/4.0/lilo-22.2.tar.bz2 Modutils (2.4.19) - 213 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/4.0/modutils-2.4.19.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/4.0/modutils-2.4.19.tar.bz2 Procinfo (18) - 22 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/4.0/procinfo-18.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/4.0/procinfo-18.tar.bz2 Procps (2.0.7) - 153 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/4.0/procps-2.0.7.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/4.0/procps-2.0.7.tar.bz2 Psmisc (21) - 172 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/4.0/psmisc-21.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/4.0/psmisc-21.tar.bz2 Shadow (4.0.3) - 760 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/4.0/shadow-4.0.3.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/4.0/shadow-4.0.3.tar.bz2 Sysklogd (1.4.1) - 67 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/4.0/sysklogd-1.4.1.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/4.0/sysklogd-1.4.1.tar.bz2 Sysvinit (2.84) - 76 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/4.0/sysvinit-2.84.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/4.0/sysvinit-2.84.tar.bz2 Util-linux (2.11u) - 1.073 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/4.0/util-linux-2.11u.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/4.0/util-linux-2.11u.tar.bz2 Netkit-base (0.17) - 49 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/4.0/netkit-base-0.17.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/4.0/netkit-base-0.17.tar.bz2 Net-tools (1.60) - 194 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/4.0/net-tools-1.60.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/4.0/net-tools-1.60.tar.bz2
Scripts de démarrage LFS (1.10) - 27 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/4.0/lfs-bootscripts-1.10.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/4.0/lfs-bootscripts-1.10.tar.bz2
Taille totale de tous les packages: 105.560 Ko (103,88 Mo)

# Chapitre 4. Préparer une nouvelle partition

# Introduction

Dans ce chapitre, nous allons préparer la partition qui accueillera le système LFS. On créera tout d'abord cette partition, ensuite on y intègrera un système de fichiers puis enfin, on la montera.

# Créer une nouvelle partition

Tout d'abord, sachez qu'il est possible de construire un système LFS sur une seule partition, en l'occurence celle sur laquelle se trouve votre distribution originale. Ceci n'est pas recommandé si c'est la première fois que vous essayez LFS, mais peut être utile si vous vous trouvez à court d'espace disque. Dans le cas où cela vous tenterait, jettez un oeil au document *Installer LFS à côté d'un système existant sur la même partition* accessible à partir de ce lien http://hints.linuxfromscratch.org/hints/lfs_next_to_existing_systems.txt.

Avant de pouvoir construire notre nouveau système Linux, nous devons avoir une partition Linux vide dans laquelle nous pourrons créer notre nouveau système. Nous vous recommandons une partition d'une taille d'au moins 1 Go. Cela vous donne assez d'espace pour stocker toutes les archives et pour compiler tous les packages sans avoir à vous préoccuper de manquer d'espace disque temporaire nécessaire. Cependant, vous aurez probablement besoin de plus d'espace que cela, si vous avez l'intention d'installer des composants additionnels et/ou si vous décidez d'utiliser le système LFS comme système principal. Si vous disposez déjà d'une partition native Linux, vous pouvez passer cette sous−section.

Comme la mémoire de votre système ne peut contenir qu'un ensemble limité de données, à un certain moment, nous recommendons que de l'espace disque soit mis de côté pour les fichiers d'échanges (swap). Un fichier d'échange est un endroit où des objets en mémoire peuvent être stockés jusqu'à ce qu'ils soient appelés. L'espace disque peut être partagé entre votre système hôte et votre système LFS. Si vous avez déjà une partition swap, alors vous n'avez probablement pas besoin d'en créer une autre. Sinon, vous devez créer une partition swap avec le programme fdisk. Néanmoins, vous devez vous rappeller la désignation de la partition swap (tel que hda2) car cela sera nécessaire pour la création du fichier `/etc/fstab`.

Le programme cfdisk (ou tout autre programme de type fdisk) devra être démarré avec comme option le disque dur approprié (tel que /dev/hda, si une nouvelle partition doit être créée sur le disque maître primaire IDE). Il permet de créer une partition native Linux. Veuillez vous reporter à la documentation concernant le programme de type fdisk de votre choix (la lecture des pages de manuel ("man") est souvent un bon début) et lisez les procédures relatives à la création d'une partition native Linux et à l'écriture d'une table de partition.

Rappelez−vous de l'identification de votre nouvelle partition. Cela sera peut être quelque chose comme hda11. Dans ce livre, cette partition nouvellement créée sera, par la suite, appelée la partition LFS.

# Créer un système de fichiers sur la nouvelle partition

Une fois la partition créée, nous devons créer le nouveau système de fichiers sur cette partition. Le système de fichiers standard utilisé actuellement est le système de fichiers ext2, mais les systèmes de fichiers journalisés deviennent aussi de plus en plus populaires. Nous supposons que vous souhaitez créer un système de fichiers ext2. Néanmoins, des instructions de construction d'autres systèmes de fichiers peuvent être trouvées sur beyond.linuxfromscratch.org/view/cvs/postlfs/filesystems.html.

Pour créer un système de fichiers ext2, utilisez la commande mke2fs. La partition LFS est la seule option nécessaire à la commande et le système de fichiers est créé.

```
mke2fs /dev/xxx
```

Remplacer "xxx" par le nom de la partition (tel que hda11).

# Monter la nouvelle partition

Maintenant que le système de fichiers a été créé, il est prêt à être utilisé. Tout ce que nous avons à faire pour accéder à la partition (c'est–à–dire pour pouvoir y lire ou écrire des données), c'est de la monter. Si vous l'avez monté sous /mnt/lfs, vous pouvez accéder à cette partition en allant dans le répertoire /mnt/lfs. Nous considérerons dans ce livre que vous l'avez montée dans /mnt/lfs. Peu importe le répertoire que vous choisirez, du moment que vous vous en souvenez.

Créez le répertoire /mnt/lfs en exécutant :

```
mkdir –p /mnt/lfs
```

Maintenant, montez la partition LFS en exécutant:

```
mount /dev/xxx /mnt/lfs
```

Remplacez << xxx>> par le nom de votre partition (tel que hda11).

Ce répertoire (/mnt/lfs) est la variable LFS dont nous avons déjà parlé précédemment durant le chapitre 2. Désormais si vous voulez utiliser la variable d'environnement LFS, il vous faudra exécuter **`export LFS=/mnt/lfs`** maintenant.

Si vous décidez de créer plusieurs partitions pour LFS (disons $LFS et $LFS/usr), montez les ainsi:

```
mkdir –p /mnt/lfs &&
mount /dev/xxx /mnt/lfs &&
mkdir /mnt/lfs/usr &&
mount /dev/yyy /mnt/lfs/usr
```

Bien sur, remplacez /dev/xxx et /dev/yyy avec les désignations appropriées des partitions.

# Chapitre 5. Préparer le système LFS

# Introduction

Dans ce chapitre, nous allons compiler et installer un système Linux minimaliste. Ce système ne contiendra que les outils nécessaires pour être capable de commencer la construction du système LFS final dans le prochain chapitre.

Les fichiers compilés dans ce chapitre vont être installés sous le répertoire `$LFS/static`, de façon à les garder séparés des fichiers installés dans le chapitre suivant. Comme tous les packages compilés ici sont simplement temporaires, nous ne voulons pas polluer le futur système LFS.

La clef pour apprendre ce qui fait fonctionner un système Linux est de savoir exactement quelle est l'utilité de chaque package et pourquoi l'utilisateur ou le système en a besoin. Pour cette raison, une courte description du contenu de chaque package est fournie juste après les instructions d'installation.

Beaucoup de nos packages doivent être corrigés avant d'être compilés. Nous appliquons des correctifs que là et que quand c'est nécessaire. Donc, ne vous tracassez pas s'il semble que des instructions pour un correctif sont manquantes.

Pendant l'installation de certains packages, vous verrez, très probablement, toutes sortes de messages d'avertissement du compilateur qui défilent sur votre écran. Ceci est normal et peut être ignoré sans danger. Il s'agit seulement de messages d'avertissement −− principalement au sujet d'utilisation incorrecte, mais pas illégale, de la syntaxe de C ou de C++. Ceci est dû au fait que les standards C ont changé assez souvent et que quelques packages continuent à utiliser les anciens standards ; ce qui n'est pas un véritable problème.

Avant de commencer, assurez−vous que la variable LFS est correctement définie, si vous avez décidé de l'utiliser. Exécutez ce qui suit :

```
echo $LFS
```

Vérifiez que la sortie contient le bon répertoire vers le point de montage de la partition LFS (/mnt/lfs par exemple).

```
echo $LFS
```

# Pourquoi utiliser une édition de lien statique ?

(Merci à Plasmatic d'avoir posté, dans une des listes de diffusion, le texte sur lequel cette partie est principalement fondée.)

Lors de la création (compilation) d'un programme, plutôt que d'avoir à réécrire toutes les fonctions pour accéder au noyau, au matériel, aux fichiers, etc., à chaque fois que vous écrivez un nouveau programme, toutes ces fonctions de base sont conservées dans des bibliothèques. Glibc, que l'on installera plus tard, est une des principales bibliothèques, qui contient le code pour toutes les fonctions de base nécessaires aux programmes, telles que l'ouverture de fichiers, l'affichage d'informations à l'écran, et la récupération des saisies faites par les utilisateurs. A la compilation du programme, ces bibliothèques sont liées au nouveau programme, de façon à ce qu'il puisse utiliser toutes les fonctions contenues dans les bibliothèques.

Cependant, ces bibliothèques peuvent être assez volumineuses (par exemple, libc.a approche souvent les 2,5Mo), vous ne voudrez sans doute pas dupliquer chaque bibliothèque liée à votre programme. Imaginez que vous ayez une commande simple comme ls liée avec une bibliothèque de 2,5Mo! Au lieu de fusionner la bibliothèque et le programme, ce qui correspond à une édition de liens statique, mieux vaut la laisser dans un fichier indépendant et ne la charger qu'en cas de besoin. C'est cette édition de liens dynamiques qui permet de charger et décharger dynamiquement une bibliothèque selon les besoins du programme.

Comme ces libraires peuvent être très grosses (par exemple, libc.a approche souvent 2.5 Mo), alors vous ne voudriez pas qu'une copie séparée de chaque librairie soit attachée au progamme. Imaginez simplement, que si vous avec une simple commande comme ls, un supplément de 2.5 Mo lui est attachée. Au lieu de faire de la librairie une partie intégrante du programme, c'est à dire une liaison statique, la librairie est stockée dans un fichier séparé, qui n'est chargé seulement qu'au moment où le programme en a besoin. C'est ce qui s'appelle une liaison dynamique, puisque la librairie est chargée et déchargée dynamiquement, en fonction des besoins du programme.

Nous avons maintenant un fichier de 1Ko et un de 2.5Mo, mais somme toute nous n'avons pas économisé d'espace mémoire (excepté peut−être en mémoire vive jusqu'au moment où la librarie est chargée). L'avantage *réel* de l'édition de liens dynamique est que nous n'avons besoin que d'une seule copie de cette bibliothèque. Si `ls` et `rm` utilisent tous deux la même librairie, nous n'avons pas besoin de deux copies de cette librarie, puisqu'ils peuvent tous les deux obtenir le code à partir du même fichier. Même en mémoire, les deux programmes se partagent le même code, plutôt que de le dupliquer en mémoire. Du coup, nous n'économisons pas seulement l'espace disque, mais aussi la mémoire vive si précieuse.

Si la liaison dynamique a tous ces avantages, pourquoi utiliser alors exclusivement la liaison statique ? Et bien, c'est parce que lorsque vous passerez via chroot dans votre environnement LFS flambant neuf (mais plutôt incomplet), ces bibliothèques dynamiques ne seront pas disponibles car elles se situeront à un endroit de votre ancienne arborescence de répertoires (`/usr/lib` par exemple) qui ne sera pas accessible depuis votre racine LFS (`$LFS`).

Ainsi, pour que vos nouveaux programmes fonctionnent dans l'environnement chroot, vous devrez vous assurer que les bibliothèques soient liées statiquement lorsque vous les compilerez ; ce qui explique les options **`--enable-static-link`**, **`--disable-shared`**, et **`-static`** utilisées tout au long du chapitre 5. Au chapitre 6, la première chose que nous ferons sera de créer la principale bibliothèque du système, glibc. Cela fait, nous commencerons à reconstruire tous les programmes étudiés au chapitre 5, mais en les liant dynamiquement cette fois, de façon à profiter du gain d'espace proposé par cette méthode.

Et maintenant vous savez pourquoi utiliser cette mystérieuse option **`-static`**. Si vous essayez de ne pas l'utiliser, vous verrez très rapidement ce qui arrivera lors de l'exécution du chroot dans votre nouveau et imparfait système LFS.

Si vous voulez en savoir plus sur les librairies à liaison dynamique, consultez un ouvrage ou un site web sur la programmation, plus spécialement consacré à Linux.

# Création du répertoire $LFS/static

Comme expliqué dans l'introduction de ce chapitre, tout ce que nous installons dans ce chapitre va être installé sous le répertoire `$LFS/static`. De cette façon, cela ne polluera pas la partition LFS avec un ensemble de fichiers temporaires. Tout ce que nous devons faire est de créer ce répertoire pour que nous puissions commencer l'installation. Lancez simplement cette commande pour créer ce répertoire:

```
mkdir $LFS/static
```

Vous voudrez peut–être déplacer les packages que vous avez téléchargé durant le chapitre 3 dans le répertoire `$LFS/static`, peut–être aussi en créant un sous–répertoire `$LFS/static/src` pour les conserver dedans.

# Installer tous les logiciels en tant qu'utilisateur non privilégié

Quand vous êtes connecté en tant que root dans le chapitre 5, il est possible que certains fichiers de votre système hôte soient écrasés par ceux que vous allez construire dans le chapitre 5. Il existe beaucoup de raisons pour que ceci puisse arriver, par exemple parce que la variable d'environnement $LFS n'est pas renseignée. Ecraser des fichiers de votre système de base entraînerait certainement toutes sortes de problèmes, c'est donc une bonne idée d'être connecté en tant qu'utilisateur non privilégié pendant le chapitre 5. Pour être sûr que l'environnement soit aussi propre que possible, nous allons créer un nouvel utilisateur "lfs" qui pourra être utilisé lors de la construction de l'installation statique. Lancer cette commande en tant que rootcréera un nouvel utilisateur "lfs" :

```
useradd -s /bin/bash -m lfs &&
passwd lfs
```

Maintenant, nous devons donner les bonnes permissions au répertoire `$LFS/static`, pour que l'utilisateur "lfs" puisse y écrire :

```
chown -R lfs $LFS/static
```

Maintenant vous pouvez vous connecter en tant qu'utilisateur "lfs". Vous pouvez faire ceci de deux façons : soit de la façon normale par la console ou l'interface graphique, soit avec **`su - lfs`**. Lorsque vous travaillez en tant qu'utilisateur "lfs", taper la commande suivante pour configurer un bon environnement de travail :

```
cat > ~/.bash_profile << "EOF"
umask 022

LFS=/mnt/lfs
LC_ALL=POSIX
CC='gcc -s'
export LFS LC_ALL
EOF
source ~/.bash_profile
```

Ce profil assure que le masque umask est positionné à 022 pour que les nouveaux fichiers et répertoires soient créés avec les bonnes permissions. Il est préférable de conserver ce paramétrage pendant l'installation de LFS. De la même manière, les variables $LFS, $LC_ALL et $CC sont définies. $LFS a déjà été expliquée dans les chapitres précédents. $LC_ALL est une variable utilisée pour l'internationalisation.

Si votre distribution hôte utilise une version de glibc antérieure à 2.2.4, et a une variable LC_ALL définie à autre chose que "C" ou "POSIX", pour le travail effectué au chapitre 5, vous pouvez rencontrer des problèmes au chapitre 6, lorsque vous quittez l'environnement chroot et essayez d'y retourner. En indiquant "POSIX" ("C" est un alias pour "POSIX"), nous nous assurons que tout fonctionnera comme prévu dans l'environnement chroot.

$CC est une variable que nous initialisons pour empêcher les symboles de debuggage d'être compilés dans les packages statiques. En oubliant ces symboles lors de l'étape de l'édition de liens de la compilation, nous sauvons de l'espacedisque et diminuons le temps de construction.

# Installer Bash–2.05a

```
Estimation du temps de construction:           1 SBU
Estimation de l'espace disque requis:  24 Mo
```

## Installation de Bash

Avant d'installer Bash, vous devez vérifier que votre distribution dispose des fichiers `/usr/lib/libcurses.a` et `/usr/lib/libncurses.a`. Si votre distribution est un système LFS, tous les fichiers seront présents si vous avez suivi les instructions du livre que vous êtes en train de lire.

Si les deux fichiers sont manquants, vous devez installer le package de développement de Ncurses. Ce package est souvent dénommé *ncurses–dev*. Si ce package est déjà installé, ou si vous venez de l'installer, vérifiez à nouveau la présence des deux fichiers. Souvent, le fichier `libcurses.a` est (encore) manquant. S'il l'est, créez `libcurses.a` comme un lien symbolique en exécutant les commandes suivantes en tant qu'utilisateur root:

```
ln -s libncurses.a /usr/lib/libcurses.a
```

Maintenant, nous pouvons continuer. Installez Bash en exécutant les commandes suivantes:

```
./configure --enable-static-link \
    --prefix=$LFS/static --with-curses &&
make &&
make install
```

Si le **`make install`** se termine avec quelque chose du style

```
install-info: unknown option `--dir-file=/mnt/lfs/usr/info/dir'
usage: install-info [--version] [--help] [--debug] [--maxwidth=nnn]
        [--section regexp title] [--infodir=xxx] [--align=nnn]
        [--calign=nnn] [--quiet] [--menuentry=xxx]
        [--info-dir=xxx]
        [--keep-old] [--description=xxx] [--test]
        [--remove] [--] filename
make[1]: *** [install] Error 1
make[1]: Leaving directory `/mnt/lfs/usr/src/bash-2.05a/doc'
make: [install] Error 2 (ignored)
```

alors, cela veut dire que vous utilisez probablement Debian–2.2 (potato) et que vous avez une ancienne version du package texinfo. Cette erreur n'est pas gênante outre mesure: les pages info seront installées lorsque nous recompilerons bash dynamiquement au chapitre 6, donc vous pouvez ignorer ce message. Il a été rapporté que la version actuelle de Debian (3.0; aussi connu sous le nom de Woody) n'a pas ce problème.

## Détail de la commande

**`--enable-static-link:`** Cette option permet à Bash d'être lié statiquement.

**`--prefix=$LFS/static:`** Cette option installe tous les fichiers bash dans le répertoire $LFS/static, qui devient le répertoire /static lors d'un chroot ou d'un reboot dans LFS.

**`--with-curses:`** Ceci permet à bash d'être lié à la bibliothèque curses plutôt qu'à la bibliothèque termcap par défaut, laquelle est devenue obsolète.

Il n'est pas nécessaire au sens strict que le bash statique soit lié à libncurses (il peut être lié à la bibliothèque statique termcap dans un premier temps car nous réinstallerons Bash dans le chapitre 6 de toute façon, où nous utiliserons libncurses), cependant c'est un bon test pour être sûr que le package ncurses a été correctement installé. S'il ne l'a pas été, vous aurez quelques problèmes plus loin dans ce chapitre lorsqu'il s'agira d'installer le package Texinfo. Ce package, qui requiert ncurses et termcap, ne peut plus alors être utilisé.

Les caractères **&&** à la fin de chaque ligne permettent de n'exécuter la prochaine commande que si la commande précédente se termine en retournant 0, indiquant ainsi qu'elle s'est exécutée sans erreur. Dans le cas où toutes les commandes sont copiées et collées dans le shell, il est important de s'assurer que si ./configure échoue, make ne soit pas exécuté et, de même, que si make échoue, make install ne soit pas exécuté, et ainsi de suite.

# Contenu de Bash

Dernière vérification effectuée auprès de la version 2.05a.

## Programmes

bash, sh (lien vers bash) et bashbug

## Descriptions

### bash

bash est le Bourne−Again SHell, un interpréteur de commandes très largement répandu sur les systèmes Unix. bash est un programme qui lit depuis l'entrée standard, le clavier. L'utilisateur tape quelque chose et le programme évaluera ce qu'il a tapé et fera quelque chose avec, comme lancer un programme.

### bashbug

bashbug est un shell de script afin d'aider l'utilisateur à composer et à expédier par courrier électronique sous un format standard les rapports de bogues concernant bash.

### sh

sh est un lien symbolique vers le programme bash. Quand bash est invoqué en tant que sh, il essaye de simuler le comportement du démarrage des versions historiques de sh aussi similairement que possible, tout en restant conforme au standard POSIX aussi.

# Dépendances d'installation de Bash

Dernière vérification effectuée auprès de la version 2.05a.

Bash: bash, sh Binutils: ar, as, ld, ranlib, size Diffutils: cmp
Fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm Gcc: cc, cc1, collect2, cpp0, gcc
Grep: egrep, grep Make: make Gawk: awk Sed: sed
Sh−utils: basename, echo, expr, hostname, sleep, uname Texinfo: install−info Textutils: cat, tr, uniq

# Installer Binutils–2.13

```
Estimation du temps de construction:           2.05 SBU
Estimation de l'espace disque requis:  160 Mo
```

## Installation de Binutils

Ce package est connu pour son mauvais fonctionnement si vous changez les options d'optimisation qui sont définies par défaut (y compris les options –march et –mcpu). Binutils fonctionne bien mieux tel quel. Donc, si vous avez défini des variables d'environnement qui écrasent les optimisations par défaut, comme CFLAGS et CXXFLAGS, nous vous recommendons de les annuler ou de les modifier lors de la construction de Binutils. Vous avez été prévenu.

Installez Binutils en exécutant les commandes suivantes :

```
mkdir ../binutils-build &&
cd ../binutils-build &&
../binutils-2.13/configure --prefix=$LFS/static --disable-nls &&
make LDFLAGS=-all-static &&
make install
```

## Explication de la commande

**`mkdir ../binutils-build`:** Les instructions d'installation pour Binutils recommandent la création d'un répertoire de construction séparé plutôt que de compiler ce package dans le répertoire des sources. Donc, nous créons un répertoire binutils–build et travaillons depuis ce répertoire.

**`--disable-nls`:** Cette option désactive l'internationalisation (aussi connue comme i18n). Nous n'avons pas besoin de cela pour nos programmes statiques et NLS cause souvent quelques problèmes lors de l'édition de liens statiques.

**`LDFLAGS=-all-static`:** Mettre la variable LDFLAGS à la valeur –all–static permet à Binutils d'être lié statiquement.

## Contenu de Binutils

Dernière vérification effectuée auprès de la version 2.12.1.

### Programmes

addr2line, ar, as, gasp, gprof, ld, nm, objcopy, objdump, ranlib, readelf, size, strings et strip

### Descriptions

#### addr2line

addr2line traduit les adresses d'un programme en nom de fichiers et numéros de ligne. A partir d'une adresse et d'un exécutable, il utilise les informations de déboguage présentes dans l'exécutable pour trouver quel nom de fichier et quel numéro de ligne sont associés à une adresse donnée.

### ar

Le programme ar crée, modifie et extrait des données depuis des archives. Une archive est un fichier contenant un ensemble d'autres fichiers dans une structure qui permet de retrouver les fichiers individuels d'origine (appelés membres de l'archive).

### as

as a d'abord été prévu pour assembler la sortie du compilateur GNU C, gcc, pour qu'elle puisse être utilisée par l'éditeur de liens ld.

### gasp

gasp est le préprocesseur de macros assembleur.

### gprof

gprof affiche les données de profil sous forme de graphe d'appel.

### ld

ld combine plusieurs fichiers objets et archives, modifie l'emplacement de leurs données et lie les références des symboles. Souvent, la dernière étape dans la construction d'un nouveau programme compilé à exécuter est un appel à ld.

### nm

nm liste les symboles des fichiers objet.

### objcopy

L'utilitaire objcopy copie le contenu d'un fichier objet dans un autre. objcopy utilise la bibliothèque GNU BFD pour lire et écrire les fichiers objet. Il peut écrire le fichier objet destination dans un format différent de celui du fichier objet source.

### objdump

objdump affiche des informations sur un ou plusieurs fichiers objet. Les options déterminent quelles informations spécifiques sont à afficher. Ces informations sont essentiellement utiles pour les programmeurs qui travaillent sur des outils de compilation, à l'inverse des programmeurs qui veulent juste compiler leur programme pour qu'il fonctionne.

### ranlib

ranlib génère un index du contenu d'une archive, et le stocke dans l'archive. L'index liste chaque symbole défini par un membre d'une archive, qui est un fichier objet re−localisable.

### readelf

readelf affiche des informations sur les binaires de type elf.

#### size

size liste les tailles des sections, et la taille totale, pour chaque fichier objet de sa liste d'arguments. Par défaut, une ligne est générée en sortie pour chaque fichier objet ou chaque module dans une archive.

#### strings

Pour chaque fichier donné, strings affiche les séquences de caractères imprimables qui ont au moins une longueur de quatre caractères (ou le nombre spécifié en option du programme) et qui sont suivies d'un caractère non imprimable. Par défaut, il affiche seulement les chaînes des parties initialisées et chargées des fichiers objet. Pour les autres types de fichier, il affiche les chaînes de tout le fichier.

strings est principalement utile pour déterminer le contenu des fichiers binaires.

#### strip

strip débarrasse les fichiers objet de tous leurs symboles, ou de certains en particulier. La liste des fichiers objet peut inclure des archives. Au moins un fichier objet doit être fourni. strip modifie les fichiers fournis en argument, plutôt que d'écrire les copies modifiées sous d'autres noms.

### Bibliothèques

libbfd.[a,so] et libopcodes.[a,so]

### Descriptions

#### libbfd

libbfd est une bibliothèqes de description des fichiers binaires.

#### libopcodes

libopcodes est une bibliothèque native pour traiter les opcodes et est utilisé dans le but de construire des utilitaires comme ojbdump. Les opcodes correspondent à une version texte facilement lisible d'instructions processeurs.

## Dépendances d'installation de Binutils

Dernière vérification effectuée auprès de la version 2.11.2.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh
Binutils: ar, as, ld, nm, ranlib, strip Diffutils: cmp
Fileutils: chmod, cp, ln, ls, mkdir, mv, rm, rmdir, touch Flex: flex
Gcc: cc, cc1, collect2, cpp0, gcc Glibc: ldconfig Grep: egrep, fgrep, grep M4: m4 Make: make
Gawk: gawk Sed: sed Sh−utils: basename, echo, expr, hostname, sleep, true, uname
Texinfo: install−info, makeinfo Textutils: cat, sort, tr, uniq

# Installer Bzip2–1.0.2

```
Estimation du temps de construction:           0.07 SBU
Estimation de l'espace disque requis:  6 Mo
```

## Installation de Bzip2

Installez Bzip2 en lançant les commandes suivantes:

```
make CC="gcc -static" &&
make PREFIX=$LFS/static install
```

Bien que cela ne fasse pas partie du système LFS de base, il est utile de mentionner qu'un correctif pour Tar peut être téléchargé pour permettre à celui–ci d'utiliser facilement bzip2/bunzip2 pour les compressions et décompressions. Avec un tar de base, vous devez utiliser des commandes comme **`bzcat fichier.tar.bz | tar -xv`** ou **`tar --use-compress-prog=bunzip2 -xvf fichier.tar.bz2`** pour utiliser bzip2 et bunzip2 avec tar. Ce correctif apporte l'option **`-j`** pour décompresser une archive bzip2 avec la commande **`tar -xvfj fichier.tar.bz2`**. Appliquer ce correctif sera décrit plus tard quand le package tar sera installé.

## Explication des commandes

**`make CC="gcc -static":`** C'est la méthode que nous utilisons pour dire à gcc que nous voulons lier bzip2 statiquement.

## Contenu de Bzip2

Dernière vérification effectuée auprès de la version 1.0.2.

### Programmes

bunzip2 (lien vers bzip2), bzcat (lien vers bzip2), bzcmp, bzdiff, bzegrep, bzfgrep, bzgrep, bzip2, bzip2recover, bzless et bzmore

### Descriptions

#### bunzip2

bunzip2 décompresse les fichiers compressés avec bzip2.

#### bzcat

bzcat (ou bzip2 –dc) décompresse les fichiers spécifiés sur la sortie standard.

#### bzcmp, bzdiff

bzcmp et bzdiff sont utilisés pour appeler les programmes cmp et diff avec des fichiers compressés avec bzip2.

#### bzegrep, bzfgrep, bzgrep

bzegrep, bzfgrep et bzgrep appellent respectivement egrep, fgrep, or grep sur des fichiers compressés avec bzip2.

#### bzip2

bzip2 compresse les fichiers grâce à l'algorithme Burrows−Wheeler et au codage Huffman. La compression est généralement bien meilleure à celle obtenue avec des compresseurs plus conventionnels basés sur LZ77/LZ78, et approche les performances de la famille PPM des compresseurs statistiques.

#### bzip2recover

bzip2recover recupère les données de fichiers bzip2 endommagés.

#### bzless

bzless est un filtre permettant d'examiner des fichiers textes compressés, un écran à la fois, sur un terminal, de la même manière que less.

#### bzmore

bzmore est un filtre permettant d'examiner des fichiers textes compressés, un écran à la fois, sur un terminal, de la même manière que more.

### Bibliothèques

libbz2.a, libbz2.so (lien vers libbz2.so.1.0), libbz2.so.1.0 (lien vers libbz2.so.1.0.2) et libbz2.so.1.0.2

#### libbz2

libbz2 est la bibliothèque implémentant l'algorithme de compression Burrows−Wheeler.

## Dépendances d'installation de Bzip2

Dernière vérification effectuée auprès de la version 1.0.1.

Bash: sh Binutils: ar, as, ld, ranlib Fileutils: cp, ln, rm Gcc: cc1, collect2, cpp0, gcc Make: make

# Installer Diffutils–2.8.1

```
Estimation du temps de construction:           0.39 SBU
Estimation de l'espace disque requis:  10 Mo
```

## Installation de Diffutils

Installez Diffutils en lançant les commandes suivantes:

```
LDFLAGS=-static CPPFLAGS=-Dre_max_failures=re_max_failures2 \
    ./configure --prefix=$LFS/static --disable-nls &&
make &&
make install
```

## Explication des commandes

**`CPPFLAGS=-Dre_max_failures=re_max_failures2 :`** La variable CPPFLAGS est lue par le programme cpp (préprocesseur C). La valeur de cette variable indique au préprocesseur de remplacer chaque instance de re_max_failures trouvée par re_max_failures2 avant de passer le fichier source au compilateur lui−même pour la compilation. Ce package a des problèmes quand on fait l'édition de liens statiques sur certains systèmes qui utilisent une vieille version de Glibc et cette construction résout ce problème.

## Contenu de Diffutils

Dernière vérification effectuée auprès de la version 2.8.1.

### Programmes

cmp, diff, diff3 et sdiff

### Descriptions

#### cmp et diff

cmp et diff comparent tous les deux deux fichiers et indiquent les différences. Tous les deux disposent d'options particulières pour comparer les fichiers dans des situations différentes.

#### diff3

La différence entre diff et diff3 est que diff compare 2 fichiers, alors que diff3 en compare 3.

#### sdiff

sdiff joint deux fichiers et affiche interactivement le résultat.

## Dépendances d'installation de Diffutils

Dernière vérification effectuée auprès de la version 2.7.

Bash: sh Binutils: ld, as Diffutils: cmp Fileutils: chmod, cp, install, mv, rm
Gcc: cc1, collect2, cpp0, gcc Grep: egrep, grep Make: make Sed: sed Sh−utils: date, hostname
Textutils: cat, tr

# Installer Fileutils–4.1

```
Estimation du temps de construction:           0.94 SBU
Estimation de l'espace disque requis:  40 Mo
```

## Installation de Fileutils

Les programmes d'un package Fileutils lié statiquement peuvent causer des erreurs de segmentation sur certains systèmes, si une version Glibc–2.2.3 ou supérieure a été installée par votre distribution. Cela semble apparaître souvent sur des machines basées sur des processeurs AMD, cependant, il existe quelques cas de systèmes Intel qui sont également affectés. Si votre système appartient à cette catégorie, appliquez le correctif.

Noter que, dans certains cas, utiliser ce correctif aboutira à une impossibilité de compiler, même si votre système dispose d'un processeur AMD et d'une version Glibc–2.2.3 ou plus récente. Dans ce cas, vous devrez besoin de supprimer le répertoire fileutils–4.1 et de déballer à nouveau l'archive tar. Nous croyons que cela peut être le cas si votre distribution a modifié Glibc–2.2.3, mais aucun détail n'est disponible pour l'instant.

Pour permettre au package de compiler correctement sur des machines AMD/Glibc–2.2.3, lancez la commande suivante. *N'essayez pas* cette correction si vous n'avez pas Glibc–2.2.3 installé. Cela créerait toutes sortes de problèmes de compilation.

```
patch -Np1 -i ../fileutils-4.1.patch
```

Installez Fileutils en lançant les commandes suivantes:

```
LDFLAGS=-static \
    ./configure --disable-nls --prefix=$LFS/static &&
make &&
make install
```

Une fois Fileutils installé, vous pouvez tester si le problème d'erreur de segmentation a été évité en lançant **`$LFS/static/bin/ls`**. Si cela fonctionne, tout va bien. Sinon, vous devez refaire l'installation en utilisant le correctif si vous ne les aviez pas utilisées, ou sans les commandes sed si vous les aviez utilisées.

## Explication de la commande

**`patch -Np1 -i ../fileutils-4.1.patch:`** Cela est utilisé pour corriger un problème lors de la construction statique de fileutils à partir de systèmes glibc 2.2.3. Si ceci n'est pas fait, alors il y a un risque que tous les programmes de fileutils causent des erreurs de segmentation, une fois entré dans chroot au chapitre 6.

## Contenu de Fileutils

Dernière vérification effectuée auprès de la version 4.1.

### Programmes

chgrp, chmod, chown, cp, dd, df, dir, dircolors, du, install, ln, ls, mkdir, mkfifo, mknod, mv, rm, rmdir, shred, sync, touch et vdir

## Descriptions

### chgrp

chgrp remplace le groupe des propriétaires des fichiers spécifiés par celui dont le nom ou le numéro est passé en paramètre.

### chmod

chmod change les permissions des fichiers spécifiés en fonction du mode, ce mode pouvant être une représentation symbolique des changements à effectuer, ou une représentation octale d'un nombre représentant le champ de bits des nouvelles permissions.

### chown

chown change le propriétaire et/ou le groupe de propriétaires des fichiers spécifiés.

### cp

cp copie des fichiers d'un répertoire à un autre.

### dd

dd copie un fichier (de l'entrée standard vers la sortie standard, par défaut) en fonction d'une taille de bloc et des options de conversion du fichier données par l'utilisateur.

### df

df affiche l'espace disque disponible du système de fichiers contenant les fichiers passés en argument. Sans argument, df indiquera les quantités correspondant à tous les systèmes de fichiers montés.

### dir, ls et vdir

dir et vdir sont des versions de ls avec des formats d'affichage différents. Ces programmes listent tous les fichiers ou répertoires donnés en argument. Le contenu des répertoires est trié alphabétiquement. Pour ls, les fichiers sont affichés en colonnes par défaut et triés verticalement si la sortie standard est un terminal; ils sont affichés ligne par ligne dans le cas contraire. Pour dir, les fichiers sont par défaut affichés en colonnes et triés verticalement. Pour vdir, l'affichage par défaut est le format long.

### dircolors

dircolors affiche une liste de commandes permettant de définir la variable d'environnement LS_COLOR. La variable LS_COLOR est utilisée pour changer les couleurs d'affichage par défaut de la commande ls et d'autres utilitaires.

### du

du affiche l'espace disque utilisé par chaque fichier et répertoire listés en argument sur la ligne de commande et pour chacun de leurs sous−répertoires.

#### install

install copie les fichiers puis positionne leurs permissions et, si possible, leur propriétaire et leur groupe.

#### ln

ln crée des liens matériels ou symboliques entre fichiers.

#### mkdir

mkdir crée un répertoire correspondant à chacun des noms mentionnés en argument.

#### mkfifo

mkfifo crée un fichier FIFO avec les noms passés en argument.

#### mknod

mknod crée un fichier FIFO, fichier spécial en mode caractère ou fichier spécial en mode bloc, avec le nom passé en argument.

#### mv

mv déplace des fichiers d'un répertoire à un autre ou les renomme, selon les arguments qui lui sont passés.

#### rm

rm efface des fichiers ou des répertoires.

#### rmdir

rmdir efface des répertoires, s'ils sont vides.

#### shred

shred efface un fichier de façon sûre, en écrasant son contenu afin qu'il ne puisse pas être récupéré.

#### sync

sync force l'écriture des blocs modifiés et met à jour le super bloc.

#### touch

touch remplace le temps d'accès et de modification des fichiers passés en argument par l'heure et la date courantes. S'ils n'existent pas, des fichiers vides sont créés.

## Dépendances d'installation de Fileutils

Dernière vérification effectuée auprès de la version 4.1.

Bash: sh Binutils: ar, as, ld, ranlib Diffutils: cmp
Fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm, rmdir Gettext: msgfmt, xgettext
Gcc: cc, cc1, collect2, cpp0, gcc Grep: egrep, fgrep, grep Make: make Perl: perl Sed: sed
Sh−utils: basename, echo, expr, hostname, sleep, uname Texinfo: install−info Textutils: cat, tr

# Installer Findutils–4.1

```
Estimation du temps de construction:           0.12 SBU
Estimation de l'espace disque requis:  8 Mo
```

## Installer Findutils

Ce package nécessite que son correctif lui soit appliqué avant de pouvoir l'installer. Assurez–vous qu'il est déballé avant de lancer les commandes d'installation.

Installez Findutils en lançant les commandes suivantes:

```
patch -Np1 -i ../findutils-4.1.patch &&
CPPFLAGS=-Dre_max_failures=re_max_failures2 \
    ./configure --prefix=$LFS/static &&
make LDFLAGS=-static &&
make install
```

## Explications des commandes

**`patch -Np1 -i ../findutils-4.1.patch`**: Ce correctif permet d'éviter des erreurs de compilation en supprimant des conflits de variables et en changeant des mauvaises syntaxes.

## Contenu du package Findutils

Dernière vérification effectuée auprès de la version 4.1.

### Programmes

bigram, code, find, frcode, locate, updatedb et xargs

### Descriptions

#### bigram

bigram est utilisé avec la commande code pour produire des bases de données de localisation ancienne formule. Pour en savoir plus sur ces trois derniers programmes, lire les pages de manuel updatedb.5.

#### code

code est l'ancêtre de frcode. Il était utilisé dans les anciennes bases de données de localisation.

#### find

Le programme find recherche des fichiers dans une arborescence selon certains critères. Si aucun critère n'est défini, il liste tous les fichiers du répertoire courant et de ses sous–répertoires.

#### frcode

frcode est lancé par updatedb pour compresser la liste des fichiers de façon à réduire la taille de la base de données d'un facteur 4 à 5.

### locate

locate scanne une base de données contenant tous les fichiers et répertoires d'un système de fichiers. Ce programme liste les fichiers et répertoires de cette base de données selon certains critères de concordance. Si un utilisateur recherche un fichier, ce programme scannera la base de données pour lui donner l'emplacement exact où ces fichiers sont situés. Ceci n'a de sens que si la base de données est régulièrement mise à jour, autrement cela ne fournirait que des informations dépassées.

### updatedb

Le programme updatedb met à jour la base de données de localisation. Il parcourt entièrement les systèmes de fichiers (sans contre indications, il parcourt tous les systèmes de fichiers montés) et met à jour la base de données qui sera ensuite utilisée par le programme locate pour retrouver ces informations. Lancer cette commande quotidiennement est une bonne pratique pour retrouver des informations à jour au moment opportun.

### xargs

La commande xargs applique une commande à une liste de fichiers. S'il faut appliquer la même commande à de multiples fichiers, une liste peut être créée pour contenir la liste de tous ces fichiers (un par ligne) et xargs exécuté une commande sur chacun d'eux.

## Dépendances d'installation de Findutils

Dernière vérification effectuée auprès de la version 4.1.

Bash: sh Binutils: ar, as, ld, ranlib Diffutils: cmp Fileutils: chmod, cp, install, mv, rm
Grep: egrep, grep Gcc: cc1, collect2, cpp0, gcc Make: make Patch: patch Sed: sed
Sh−utils: basename, date, echo, hostname Textutils: cat, tr

# Installer Gawk–3.1.1

```
Estimation du temps de construction:           0.39 SBU
Estimation de l'espace disque requis:  17 Mo
```

# Installation de Gawk

Installez Gawk en lançant les commandes suivantes:

```
CPPFLAGS=-Dre_max_failures=re_max_failures2 \
    ./configure --prefix=$LFS/static --disable-nls &&
make LDFLAGS=-static &&
make install
```

# Contenu de Gawk

Dernière vérification effectuée auprès de la version 3.1.1.

## Programmes

awk, gawk, gawk–3.1.1, grcat, igawk, pgawk, pgawk–3.1.1, pwcat

## Descriptions

### awk

awk est un lien symbolique vers gawk.

### gawk, gawk–3.1.1

gawk est l'implémentation GNU de awk, un langage basé sur le traitement et la reconnaissance de chaînes de caractères.

### grcat

grcat concatène la base de données des groupes, /etc/group.

### igawk

igawk est un script shell qui donne à gawk la possibilité d'inclure des fichiers.

### pgawk, pgawk–3.1.1

pgawk est la version de débuggage de gawk.

### pwcat

pwcat concatène la base de données des utilisateurs, /etc/passwd.

# Dépendances d'installation de Gawk

Dernière vérification effectuée auprès de la version 3.1.0.

(Aucune dépendance encore vérifiée)

# Installer GCC–3.2

```
Estimation du temps de construction:           9.48 SBU
Estimation de l'espace disque requis:  326 Mo
```

## Installation de GCC

Ce package nécessite que son correctif lui soit appliqué avant que vous ne puissiez l'installer. Assurez–vous qu'il est déballé avant de lancer les commandes d'installation.

Ce package est connu pour se comporter bizarrement si vous avez changé les options d'optimisation par défaut (y compris les options –march et –mcpu). GCC se comporte mieux sans. Donc, si vous avez défini des variables d'environnement qui écrasent les optimisations par défaut, comme CFLAGS et CXXFLAGS, nous vous recommandons de les annuler ou de les modifier lors de la construction de GCC. Vous avez été prévenu.

Installez GCC en lançant les commandes suivantes :

```
patch -Np1 -i ../gcc-3.2.patch &&
patch -Np1 -i ../gcc-3.2-nofixincludes-2.patch &&
mkdir ../gcc-build &&
cd ../gcc-build &&
../gcc-3.2/configure --prefix=/static --enable-languages=c \
   --disable-nls --disable-shared &&
echo "#define HAVE_GAS_HIDDEN 1" >> gcc/auto-host.h &&
make BOOT_LDFLAGS=-static bootstrap &&
make prefix=$LFS/static install &&
ln -s gcc $LFS/static/bin/cc
```

## Explication des commandes

**`patch -Np1 -i ../gcc-3.2.patch` :** Ce correctif corrige quelques problèmes. Voir http://www.zipworld.com.au/~gschafer/lfs–tweaks.html pour plus d'informations.

**`patch -Np1 -i ../gcc-3.2-nofixincludes-2.patch` :** Ceci empêche le script fixincludes d'être exécuté.

**`--prefix=/static` :** Ce n'est PAS une erreur de frappe. GCC garde en dur certains chemins lors de la compilation et nous devons donc passer /static comme préfixe lors du ./configure. Nous donnons le véritable préfixe d'installation plus tard lors de la commande make install.

**`--enable-languages=c` :** Ceci crée le compilateur C. Le compilateur C++ sera construit lors du chapitre 6, quand nous reconstruirons GCC. D'autres compilateurs sont également disponibles. Si ils vous sont nécessaires, le paramètre ––enable–languages peut être omis.

**`echo "#define HAVE_GAS_HIDDEN 1"` :** Ceci définit la directive assembleur .hidden pour que nous ne construisons pas une Glibc erronée par la suite.

**`make BOOT_LDFLAGS=-static` :** Ceci est équivalent à make LDFLAGS=–static que nous utilisons pour d'autres packages pour les compiler statiquement.

**`ln -s gcc $LFS/static/bin/cc` :** Ceci crée le lien symbolique $LFS/static/bin/gcc, dont certains packages ont besoin.

# Contenu de GCC

Dernière vérification effectuée auprès de la version 3.1.

## Programmes

c++, c++filt, cc (lien vers gcc), cc1, cc1plus, collect2, cpp, cpp0, g++, gcc, gccbug, gcov, tradcpp0

## Descriptions

### cc, cc1, cc1plus, gcc

Ce sont les compilateurs C. Un compilateur transforme le code source au format texte dans un format que l'ordinateur comprends. Après qu'un code source ait été compilé en un fichier objet, un éditeur de lien va créer un fichier exécutable à partir d'un ou plusieurs de ces fichiers générés par le compilateur.

### c++, cc1plus, g++

Ce sont les compilateurs C++, l'équivalent de cc et gcc, etc...

### c++filt

Le langage C++ procure la surcharge de fonctions, c'est−à−dire qu'il est possible d'écrire plusieurs fonctions possédant le même nom, à condition que chaque fonction utilise des paramètres de différents types. Tous les noms des fonctions C++ sont codés avec un label assembleur bas niveau (ce procédé est aussi connu sous le nom de mangling). Le programme c++filt réalise l'opération inverse; il décode les noms bas niveau en des noms de niveau utilisateur, de manière à ce que l'éditeur de liens ne confonde pas les fonctions surchargées.

### collect2

collect2 assiste à la compilation de constructeurs.

### cpp, cpp0

cpp réalise un travail préliminaire sur un fichier source, comme inclure le contenu des fichiers d'entêtes dans le code source. Il suffit d'insérer une ligne comme #include <filename> dans votre fichier source. Le préprocesseur va insérer le contenu de ce fichier dans le fichier source.

### gccbug

gccbug est un script shell, utilisé pour simplifier la création de rapports de bugs.

### gcov

gcov analyse les programmes pour aider à créer des codes plus efficaces, plus rapides avec une optimisation.

### tradcpp0

Aucune description actuellement disponible.

## Bibliothèques

libgcc.a, libgcc_eh.a, libgcc_s.so, libiberty.a, libstdc++.[a,so], libsupc++.a

## Descriptions

### libgcc, libgcc_eh, libgcc_s

Fichiers de support pour gcc.

### libiberty

libiberty est une collection de sous−routines utilisées par différents programmes GNU, comme getopt, obstack, strerror, strtol et strtoul.

### libstdc++

libstdc++ est la bibliothèque C++. Elle est utilisée par les programmes C++ et contient des fonctions couramment utilisées par eux. De cette façon, le développeur n'a pas besoin d'écrire certaines fonctions (comme écrire une ligne de texte à l'écran) à partir de rien, à chaque fois qu'il crée un programme.

### libsupc++

libsup++ procure une aide au langage de programmation c++. Entre autres choses, libsup++ contient des routines pour la gestion des exceptions.

# Dépendances d'installation de GCC

Dernière vérification effectuée auprès de la version 2.95.3.

Bash: sh Binutils: ar, as, ld, nm, ranlib Diffutils: cmp
Fileutils: chmod, cp, ln, ls, mkdir, mv, rm, touch Find: find Gcc: cc, cc1, collect2, cpp0, gcc
Grep: egrep, grep Make: make Patch: patch Sed: sed
Sh−utils: basename, dirname, echo, expr, hostname, sleep, true, uname Tar: tar
Texinfo: install−info, makeinfo Textutils: cat, tail, tr

# Installer Grep–2.5

```
Estimation du temps de construction:           0.26 SBU
Estimation de l'espace disque requis:  5 Mo
```

## Installation de Grep

Installez Grep en lançant les commandes suivantes:

```
LDFLAGS=-static CPPFLAGS=-Dre_max_failures=re_max_failures2 \
    ./configure --prefix=$LFS/static --disable-nls \
    --disable-perl-regexp &&
make &&
make install
```

## Explication des commandes

**`--disable-perl-regexp` :** Cette option de configuration assure que grep n'est pas lié avec la bibliothèque PCRE, qui n'est souvent disponible qu'en tant que bibliothèque partagée dans les distributions. Ne pas utiliser cette option pourrait entraîner des erreurs de compilation.

## Contenu de Grep

Dernière vérification effectuée auprès de la version 2.5.

### Programmes

egrep (lien vers grep), fgrep (lien vers grep) et grep

### Descriptions

#### egrep

egrep affiche les lignes des fichiers correspondant au motif d'une expression régulière étendue.

#### fgrep

fgrep affiche les lignes des fichiers contenant une expression littérale.

#### grep

grep affiche les lignes des fichiers correspondant au motif d'une expression régulière basique.

## Dépendances d'installation de Grep

Dernière vérification effectuée auprès de la version 2.4.2.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh Binutils: as, ld Diffutils: cmp Fileutils: chmod, install, ls, mkdir, mv, rm Gettext: msgfmt, xgettext
Gcc: cc, cc1, collect2, cpp0, gcc Glibc: getconf Grep: egrep, fgrep, grep M4: m4 Make: make

Gawk: gawk Sed: sed Sh−utils: basename, echo, expr, hostname, sleep, uname
Texinfo: install−info, makeinfo Textutils: cat, tr

# Installer Gzip–1.2.4a

```
Estimation du temps de construction:           0.04 SBU
Estimation de l'espace disque requis:  2 Mo
```

# Installation de Gzip

Installez Gzip en lançant les commandes suivantes:

```
./configure --prefix=$LFS/static &&
make LDFLAGS=-static &&
make install
```

# Contenu de Gzip

Dernière vérification effectuée auprès de la version 1.2.4a.

## Programmes

gunzip (lien vers gzip), gzexe, gzip, uncompress (lien vers gunzip), zcat (lien vers gzip), zcmp, zdiff, zforce, zgrep, zmore et znew

## Description

### gunzip, uncompress

gunzip et uncompress décompressent les fichiers qui ont été compressés par gzip.

### gzexe

gzexe vous permet de compresser des exécutables et de les décompresser puis de les lancer automatiquement (avec cependant une baisse des performances).

### gzip

gzip réduit la taille des fichiers en utilisant l'algorithme de compression Lempel–Ziv (LZ77).

### zcat

zcat décompresse, et écrit sur la sortie standard, une liste de fichiers donnée en ligne de commande ou sur l'entrée standard en lecture.

### zcmp

zcmp a le même fonctionnement que le programme cmp mais sur des fichiers compressés.

### zdiff

zdiff a le même fonctionnement que le programme diff mais sur des fichiers compressés.

### zforce

zforce force l'extension .gz sur les fichiers compressés par gzip de façon à ce que gzip ne les compresse pas une deuxième fois. Ceci peut être utile pour des fichiers dont le nom a été tronqué après un transfert de fichier.

### zgrep

zgrep a le même fonctionnement que le programme grep mais sur des fichiers compressés.

### zmore

zmore est un programme permettant d'avoir le comportement du programme more aussi bien sur des fichiers normaux que compressés.

### znew

znew convertit les fichiers de format .Z (compress) en .gz (gzip).

## Dépendances d'installation de Gzip

Dernière vérification effectuée auprès de la version 1.2.4a.

Bash: sh Binutils: as, ld, nm Fileutils: chmod, cp, install, ln, mv, rm
Gcc: cc1, collect2, cpp, cpp0, gcc Grep: egrep, grep Make: make Sed: sed Sh−utils: hostname
Textutils: cat, tr

# Installer Make–3.79.1

```
Estimation du temps de construction:           0.26 SBU
Estimation de l'espace disque requis:  8 Mo
```

## Installation de Make

Installez Make en lançant les commandes suivantes :

```
./configure --prefix=$LFS/static --disable-nls &&
make LDFLAGS=-static &&
make install
```

Lors de la phase du 'make install', vous verrez ce message d'avertissement :

```
chgrp: changing group of `/mnt/lfs/static/bin/make':
Operation not permitted /mnt/lfs/static/bin/make needs to be owned by
group kmem and setgid; otherwise the `-l' option will probably not work.
You may need special privileges to complete the installation of
/mnt/lfs/static/bin/make.
```

Vous pouvez en toute sécurité ignorer ce message. make n'a pas besoin d'appartenir au groupe kmem et d'utiliser setgid pour que l'option *–l* fonctionne. (Cette option demande à make de ne démarrer aucun nouveau travail quand une certaine charge sur le système a été atteinte.)

## Contenu de Make

Dernière vérification effectuée auprès de la version 3.79.1.

### Programmes

make

### Descriptions

#### make

make détermine automatiquement quelles parties d'un gros programme ont besoin d'être recompilées, et exécute les commandes pour les recompiler.

## Dépendances d'installation de Make

Dernière vérification effectuée auprès de la version 3.79.1.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh Binutils: as, ld Diffutils: cmp
Fileutils: chgrp, chmod, install, ls, mv, rm Gcc: cc, cc1, collect2, cpp0, gcc Glibc: getconf
Grep: egrep, fgrep, grep M4: m4 Make: make Gawk: gawk Sed: sed
Sh–utils: basename, echo, expr, hostname, sleep, uname Texinfo: install–info, makeinfo
Textutils: cat, tr

# Installer Patch–2.5.4

```
Estimation du temps de construction:           0.10 SBU
Estimation de l'espace disque requis:  3 Mo
```

## Installation de Patch

Installez Patch en lançant les commandes suivantes:

```
CPPFLAGS=-D_GNU_SOURCE \
    ./configure --prefix=$LFS/static &&
make LDFLAGS=-static &&
make install
```

## Explication des commandes

**`CPPFLAGS=-D_GNU_SOURCE :`** Cette option corrige des problèmes d'installation du package sur des plateformes PPC et m68k (autant que nous le sachions). Cela ne pose pas de problèmes sur les autres plateformes, comme les x86, donc nous positionnons cette variable par défaut.

## Contenu de Patch

Dernière vérification effectuée auprès de la version 2.5.4.

### Programmes

patch

### Descriptions

#### patch

Le programme patch modifie un fichier en fonction d'un fichier de patch, aussi appelé correctif. Ce correctif est habituellement une liste créée par le programme diff et contient les instructions pour modifier le fichier original. Patch est très souvent utilisé pour la modification de code source car il fait économiser du temps et de la taille mémoire. Imaginez un package faisant 1 Mo. La version suivante de ce package n'a modifié que deux fichiers de la précédente version. On peut alors le considérer entièrement comme un nouveau package d'1 Mo ou juste sous la forme d'un correctif d'1 Ko qui viendra mettre à jour la précédente version pour la rendre identique à la dernière version réalisée. Ainsi, si ma première version avait déjà été téléchargée, un correctif évite un deuxième téléchargement conséquent.

## Dépendances d'installation de Patch

Dernière vérification effectuée auprès de la version 2.5.4.

Bash: sh Binutils: as, ld Diffutils: cmp Fileutils: chmod, install, mv, rm
Gcc: cc, cc1, collect2, cpp0, gcc Glibc: getconf Grep: egrep, grep Make: make Sed: sed
Sh–utils: echo, expr, hostname, uname Textutils: cat, tr

# Installer Sed–3.02

```
Estimation du temps de construction:           0.09 SBU
Estimation de l'espace disque requis:  2 Mo
```

# Installation de Sed

Installez Sed en lançant les commandes suivantes :

```
CPPFLAGS=-Dre_max_failures=re_max_failures2 \
    ./configure --prefix=$LFS/static &&
make LDFLAGS=-static &&
make install
```

# Contenu de Sed

Dernière vérification effectuée auprès de la version 3.02.

## Programmes

sed

## Descriptions

### sed

sed est un éditeur de flux. Un éditeur de flux est utilisé pour exécuter des transformations de texte basiques sur un flux d'entrée (un fichier ou l'entrée reçue d'un tuyau).

# Dépendances d'installation de Sed

Dernière vérification effectuée auprès de la version 3.02.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh Binutils: ar, as, ld, ranlib
Diffutils: cmp Fileutils: chmod, install, ls, mv, rm Gcc: cc1, collect2, cpp0, gcc Glibc: getconf
Grep: egrep, fgrep, grep M4: m4 Make: make Gawk: gawk Sed: sed
Sh–utils: echo, expr, hostname, sleep Texinfo: install–info, makeinfo Textutils: cat, tr

# Installer Sh–utils–2.0

```
Estimation du temps de construction:           0.47 SBU
Estimation de l'espace disque requis:  42 Mo
```

## Installation de Sh–utils

Avant que Sh–utils ne soit installé, il peut être nécessaire d'appliquer le correctif pour sh–utils. Ce correctif permet d'éviter un conflit sur des noms de variables avec certaines versions de Glibc (généralement glibc–2.1.x) lors de la compilation de sh–utils en statique. Il est de toute façon sain d'appliquer le correctif, même si vous utilisez une version différente de glibc. Donc, dans le cas où vous n'êtes pas sûr, il est préférable de l'appliquer.

```
patch -Np1 -i ../sh-utils-2.0.patch
```

Installez Sh–utils en lançant les commandes suivantes :

```
./configure --prefix=$LFS/static \
    --disable-nls &&
make LDFLAGS=-static &&
make install
```

Durant l'étape "make install", vous allez voir le message d'avertissement suivant :

```
WARNING: insufficient access; not installing su
NOTE: to install su, run 'make install-root' as root
```

Vous pouvez ignorer en toute sécurité ce message. Vous auriez besoin d'être connecté en tant que root pour pouvoir installer su de la façon que Sh–utils le souhaite, c'est–à–dire en suid root. Parce que nous n'avons pas besoin de su lors du chapitre 6, et que de toute façon su sera correctement installé quand nous réinstallerons Sh–utils lors du chapitre 6, vous pouvez faire comme si vous n'aviez pas vu ce message.

## Contenu de Sh–utils

Dernière vérification effectuée auprès de la version 2.0.

### Programmes

basename, chroot, date, dirname, echo, env, expr, factor, false, groups, hostid, id, logname, nice, nohup, pathchk, pinky, printenv, printf, pwd, seq, sleep, stty, su, tee, test, true, tty, uname, uptime, users, who, whoami et yes

### Descriptions

#### basename

basename enlève le répertoire et les suffixes d'un nom de fichier.

### chroot

chroot exécute une commande ou un shell interactif depuis un répertoire racine particulier.

### date

date affiche le temps courant dans le format spécifié, ou met à jour la date système.

### dirname

dirname enlève les suffixes ne faisant pas partie du répertoire d'un nom de fichier.

### echo

echo affiche une ligne de texte.

### env

env exécute un programme dans un environnement modifié.

### expr

expr évalue des expressions.

### factor

factor affiche les facteurs premiers des nombres entiers spécifiés.

### false

false sort toujours avec un code de statut indiquant une erreur.

### groups

groups affiche les groupes auquels appartient l'utilisateur.

### hostid

hostid affiche l'identifiant numérique (en hexadécimal) pour l'hôte courant.

### id

id affiche les identifiants effectifs d'utilisateur et de groupe de l'utilisateur courant ou d'un utilisateur donné.

### logname

logname affiche le nom de login de l'utilisateur.

### nice

nice exécute un programme avec une priorité de scheduling modifiée.

### nohup

nohup lance une commande qui ne tiendra pas compte du signal hangups, avec une sortie vers un fichier de trace.

### pathchk

pathchk vérifie si les noms de fichiers sont valides et portables.

### pinky

pinky est une commande finger allégée qui permet de retrouver des informations sur certains utilisateurs.

### printenv

printenv affiche tout ou partie des variables d'environnement.

### printf

printf formate et affiche des données (identique à la fonction printf en C).

### pwd

pwd affiche le nom du répertoire de travail courant.

### seq

seq affiche des nombres dans un certain ordre avec un incrément à spécifier.

### sleep

sleep stoppe une tâche pendant un certain laps de temps.

### stty

stty change et affiche les caractéristiques d'une ligne de terminal.

### su

su exécute un shell en substituant l'identifiant d'utilisateur et de groupe.

### tee

tee lit l'entrée standard et écrit sur la sortie standard et un fichier.

### test

test vérifie les types de fichiers et compare les valeurs.

### true

true sort toujours avec un code de statut indiquant le succès.

### tty

tty affiche le nom de fichier du terminal connecté à l'entrée standard.

### uname

uname affiche des informations système.

### uptime

uptime donne le temps depuis lequel le système fonctionne.

### users

users affiche les noms des utilisateurs actuellement connectés sur la machine hôte.

### who

who affiche qui est connecté.

### whoami

whoami affiche le nom de l'utilisateur associé à l'identifiant utilisateur effectif courant.

### yes

yes affiche 'y' ou une chaîne de caractères donnée répétitivement jusqu'à ce que la tâche soit tuée.

## Dépendances d'installation de Sh−utils

Dernière vérification effectuée auprès de la version 2.0.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh Binutils: ar, as, ld, ranlib
Diffutils: cmp Fileutils: chmod, chown, install, ls, mv, rm Gettext: msgfmt, xgettext
Gcc: cc, cc1, collect2, cpp0, gcc Glibc: getconf Grep: egrep, fgrep, grep M4: m4 Make: make
Gawk: gawk Perl: perl Sed: sed Sh−utils: basename, echo, expr, hostname, sleep, uname Tar: tar
Texinfo: install−info, makeinfo Textutils: cat, tr

# Installer Tar–1.13

```
Estimation du temps de construction:           0.25 SBU
Estimation de l'espace disque requis:  10 Mo
```

## Installation de Tar

Pour pouvoir utiliser directement les fichiers bzip2 avec tar, utiliser le correctif tar disponible depuis le site FTP LFS. Ce correctif ajoutera à Tar l'option –j, qui fonctionne de manière identique à l'option –z de tar (qui permet d'utiliser directement les fichiers gzip).

Installez le correctif en lançant la commande suivante :

```
patch -Np1 -i ../tar-1.13.patch
```

Installez Tar en lançant les commandes suivantes :

```
./configure --prefix=$LFS/static --disable-nls &&
make LDFLAGS=-static &&
make install
```

## Contenu de Tar

Dernière vérification effectuée auprès de la version 1.13.

### Programmes

rmt et tar

### Descriptions

#### rmt

rmt est un programme utilisé par les programmes distants dump et restore pour manipuler un lecteur de bandes magnétiques via une connection de communication entre processus.

#### tar

tar est un programme de sauvegarde destiné à stocker et à extraire des fichiers depuis un format d'archive connu sous le nom de fichier tar.

## Dépendances d'installation de Tar

Dernière vérification effectuée auprès de la version 1.13.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh Binutils: ar, as, ld, ranlib
Diffutils: cmp Fileutils: chmod, install, ls, mv, rm Gettext: msgfmt, xgettext
Gcc: cc, cc1, collect2, cpp0, gcc Glibc: getconf Grep: egrep, fgrep, grep M4: m4 Make: make
Gawk: gawk Net–tools: hostname Patch: patch Sed: sed Sh–utils: basename, echo, expr, sleep, uname
Texinfo: install–info, makeinfo Textutils: cat, tr

# Installer Texinfo–4.2

```
Estimation du temps de construction:           0.47 SBU
Estimation de l'espace disque requis:  19 Mo
```

# Installation de Texinfo

Installez Texinfo en lançant les commandes suivantes:

```
LDFLAGS=-static ./configure --prefix=$LFS/static \
    --disable-nls &&
make &&
make install
```

# Contenu de Texinfo

Dernière vérification effectuée auprès de la version 4.2.

## Programmes

info, infokey, install–info, makeinfo, texi2dvi et texindex

## Descriptions

### info

Le programme info lit les documents de type Info, habituellement stockés dans le répertoire /usr/doc/info. Les documents info sont identiques aux pages de manuel, mais ils tendent à être plus détaillés et n'expliquent pas seulement les différentes options d'un programme.

### infokey

infokey compile un fichier source contenant des personnalisations Info en un format binaire.

### install–info

Le programme install–info met à jour les documents info. Lorsque le programme info est invoqué, il présente une liste des sujets disponibles (c'est–à–dire disponibles dans les documents info). Le programme install–info est utilisé pour tenir à jour cette liste. Si des fichiers info sont effacés manuellement, vous devez aussi supprimer les entrées correspondantes dans le fichier d'index. Ce programme le fait et fait aussi l'inverse lorsque des documents info sont ajoutés.

### makeinfo

Le programme makeinfo traduit des documents sources Texinfo en d'autres formats. Les formats disponibles sont : fichiers info, texte et HTML.

### texi2dvi

Le programme texi2dvi affiche les documents Texinfo.

### texindex

Le programme texindex est utilisé pour trier les fichiers d'index Texinfo.

## Dépendances d'installation de Texinfo

Dernière vérification effectuée auprès de la version 4.0.

Bash: sh Binutils: ar, as, ld, ranlib Diffutils: cmp Fileutils: chmod, install, ln, ls, mkdir, mv, rm
Gcc: cc1, collect2, cpp0, gcc Grep: egrep, fgrep, grep Make: make Sed: sed
Sh−utils: basename, echo, expr, hostname, sleep Texinfo: makeinfo Textutils: cat, tr

# Installer Textutils–2.1

```
Estimation du temps de construction:           0.95 SBU
Estimation de l'espace disque requis:  49 Mo
```

# Installation de Textutils

Installez Textutils en lançant les commandes suivantes:

```
CPPFLAGS=-Dre_max_failures=re_max_failures2 \
    ./configure --prefix=$LFS/static \
    --disable-nls &&
make LDFLAGS=-static &&
make install
```

# Contenu de Textutils

Dernière vérification effectuée auprès de la version 2.0.

## Programmes

cat, cksum, comm, csplit, cut, expand, fmt, fold, head, join, md5sum, nl, od, paste, pr, ptx, sort, split, sum, tac, tail, tr, tsort, unexpand, uniq et wc

## Descriptions

### cat

cat concatène un ou plusieurs fichiers ou entrée standard vers la sortie standard.

### cksum

cksum affiche la somme de contrôle cyclique et le compte d'octets des fichiers spécifiés.

### comm

comm compare deux fichiers triés ligne par ligne.

### csplit

csplit découpe un fichier en plusieurs sections nommées xx01, xx02, ..., et affiche le compte d'octets de chaque section sur la sortie standard.

### cut

cut sélectionne certaines colonnes d'un ou plusieurs fichiers et les affiche sur la sortie standard.

### expand

expand convertit les tabulations d'un fichier en espaces, il affiche le résultat sur la sortie standard.

### fmt

fmt reformate chaque paragraphe d'un ou plusieurs fichiers et affiche le résultat sur la sortie standard.

### fold

fold enregistre les lignes entrées dans des fichiers spécifiés (sur l'entrée standard par défaut), et écrit sur la sortie standard.

### head

head affiche les xx premières lignes (10 par défaut) des fichiers spécifiés sur la sortie standard.

### join

join fusionne les lignes de deux fichiers ayant un champ commun.

### md5sum

md5sum affiche ou vérifie la somme de contrôle MD5.

### nl

nl affiche les fichiers spécifiés sur la sortie standard en ajoutant le numéro de ligne.

### od

od affiche le contenu d'un fichier en octal par défaut sur la sortie standard.

### paste

paste affiche des lignes regroupant les lignes correspondantes de chaque fichier, séparées par des tabulations, sur la sortie standard.

### pr

pr met en page des fichiers pour l'impression.

### ptx

ptx produit un index permuté du contenu d'un fichier.

### sort

sort écrit la concaténation triée de fichiers sur la sortie standard.

### split

split découpe un fichier en sections de taille préfixée nommées PREFIXaa, PREFIXab, ...

### sum

sum affiche la somme de contrôle et le compte de blocs pour chaque fichier spécifié.

### tac

tac écrit les fichiers spécifiés sur la sortie standard, dans l'ordre inverse.

### tail

tail affiche les xx dernières lignes (10 par défaut) des fichiers spécifiés sur la sortie standard.

### tr

tr transpose ou élimine des caractères de l'entrée standard et affiche le résultat sur la sortie standard.

### tsort

tsort affiche une liste ordonnée cohérente à partir de fichiers partiellement ordonnés.

### unexpand

unexpand convertit les espaces en tabulations pour les fichiers spécifiés, et affiche le résultat sur la sortie standard.

### uniq

uniq efface les lignes dupliquées d'un fichier trié.

### wc

wc affiche le nombre de lignes, de mots et d'octets pour tous les fichiers spécifiés, et un total du nombre de lignes si plusieurs fichiers ont été spécifiés.

## Dépendances d'installation de Textutils

Dernière vérification effectuée auprès de la version 2.0.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh Binutils: ar, as, ld, ranlib
Diffutils: cmp Fileutils: chmod, install, ls, mv, rm Gettext: msgfmt, xgettext
Gcc: cc, cc1, collect2, cpp0, gcc Glibc: getconf Grep: egrep, fgrep, grep M4: m4 Make: make
Gawk: gawk Net−tools: hostname Perl: perl Sed: sed Sh−utils: basename, echo, expr, sleep, uname
Tar: tar Texinfo: install−info, makeinfo Textutils: cat, tr

# Installer Util–linux–2.11u

```
Estimation du temps de construction:           0.09 SBU
Estimation de l'espace disque requis:  9 Mo
```

## Installation de Util–linux

Nous avons seulement besoin des programmes mount et umount pour le moment, donc nous ne compilerons pas le package entier.

Installez Util–linux en lançant les commandes suivantes:

```
./configure &&
make -C lib &&
make -C mount LDFLAGS=-static mount umount &&
cp mount/{mount,umount} $LFS/static/bin
```

## Dépendances d'installation de Util–linux

Dernière vérification effectuée auprès de la version 2.11n.

Bash: sh Binutils: as, ld Diffutils: cmp Fileutils: chgrp, chmod, cp, install, ln, mv, rm
Gettext: msgfmt, xgettext Gcc: cc, cc1, collect2, cpp, cpp0 Glibc: rpcgen Grep: grep Make: make
Sed: sed Sh–utils: uname, whoami textutils: cat

# Chapitre 6. Installation des logiciels de base du système

# Introduction

Dans ce chapitre, nous entrons le site de construction et démarrons la construction de notre système LFS. C'est−à−dire, nous entrons avec chroot dans le mini système Linux temporaire, et lançons l'installation de tous les packages un par un.

L'installation de tous les logiciels est plutôt simple et vous allez probablement penser qu'il est tellement plus facile et plus court de donner les instructions génériques d'installation pour un package et de seulement expliquer en détail l'installation des packages nécessitant une méthode alternative. Même si nous sommes d'accord avec ceci, nous avons choisi de donner les instructions complètes pour chaque package, simplement pour éviter toute confusion ou erreur.

Si vous comptez utiliser une optimisation de compilation pour les packages installés dans ce chapitre, jetez un oeil aux astuces d'optimisation dans http://hints.linuxfromscratch.org/hints/optimization.txt. Ces options peuvent faire qu'un programme s'exécute plus vite, mais peuvent également causer des problèmes de compilation. Si vous rencontrez des problèmes après avoir utilisé les optimisations, essayez toujours sans optimisation pour voir si le problème persiste.

S'il vous plait, n'installez pas plus d'un package à la fois, même si cela peut vous sauver du temps (spécialement avec des machines bi−processeur). Si vous installez le mauvais package au mauvais moment, cela peut inscrire en dur le chemin du répertoire `$LFS/static` dans les programmes, ce qui empêchera certains programmes de fonctionner correctement une fois le répertoire static supprimé. L'ordre dans lequel les packages sont installés dans ce chapitre doit être suivi scrupuleusement pour s'assurer qu'aucun chemin vers static n'est inscrit en dur.

# A propos des symboles de débogage

La plupart des programmes et des bibliothèques sont compilés, par défaut, avec des symboles de débogage (avec l'option –g de gcc).

Lors du débogage d'un programme ou d'une librairie qui a été compilée avec les informations de débogage inclues, le débogueur ne vous donnera pas seulement les adresses mémoires, mais aussi les noms des routines et des variables.

Mais l'inclusion de ces symboles de débogage grossit le programme ou la bibliothèque de manière significative. Pour avoir une idée de la quantité d'espace que ces symboles occupent, regarder ceci :

- Un binaire Bash dynamique avec symboles de débogage: 1200 Ko.
- Un binaire Bash dynamique sans symboles de débogage: 478 Ko.
- Fichiers glibc et gcc (/lib et /usr/lib) avec symboles de débogage: 87 Mo.
- Fichiers glibc et gcc sans symboles de débogage: 16 Mo.

Les tailles peuvent varier un peu selon le compilateur utilisé et la version de labibliothèque C utilisée. Mais lors d'une comparaison entre des programmes avecles symboles de débogage et ceux sans, la différence sera généralement d'unfacteur 2 à 5.

Comme la plupart des gens n'utiliseront probablement jamais un débogueur sur leurs logiciels, beaucoup de place disque peut être gagnée en supprimant ces symboles.

Pour enlever les symboles de débogage d'un binaire (qui doit être de typea.out ou ELF), exécutez **`strip --strip-debug filename`**. Les jokers peuvent être utilisés pour supprimer pour traiter plusieurs fichiers (utilisez quelque chose comme **`strip --strip-debug $LFS/usr/bin/*`**).

Pour vous faciliter l'opération, le chapitre 9 comprend une commande unique pour supprimer tous les symboles de débogage des programmes et bibliothèques de votre système. Vous obtiendrez des informations supplémentaires dans l'astuce http://hints.linuxfromscratch.org/hints/optimization.txt.

# Entrée dans l'environnement "chrooté"

C'est le moment d'entrer dans l'environnement chroot afin d'installer le reste des programmes nécessaires. Avant de lancer la commande chroot, vous devez être *root* car seul l'utilisateur *root* peut utiliser la commande **`chroot`**.

Devenez *root* et lancez la commande suivante pour entrer dans l'environnement chroot:

```
chroot $LFS /static/bin/env -i \
    HOME=/root TERM=$TERM PS1='\u:\w\$ ' \
    PATH=/bin:/usr/bin:/sbin:/usr/sbin:/static/bin \
    /static/bin/bash --login
```

L'option **`-i`** donné à la commande **`env`** efface toutes les variables de l'environnement chroot. Après cela, seules les variables HOME, TERM, PS1 et PATH sont initialisées de nouveau. La commande TERM=$TERM initialise la variable TERM à l'intérieur de chroot à la même valeur que celle à l'extérieur de chroot; cette variable est utilisée par des programmes comme vim et less pour fonctionner correctement. Si vous avez besoin d'autres variables, telles que CFLAGS ou CXXFLAGS, initialisez–les de même.

Dès maintenant, nous n'avons plus besoin d'utiliser la variable LFS –– car le shell pense que `/mnt/lfs` est réellement `/`.

Assurez–vous que les commandes suivantes dans le reste de ce chapitre et les suivants sont exécutées dans l'environnement chroot. Si vous quittez l'environnement chroot (en redémarrant l'ordinateur par exemple), vous devez vous rappeller de retourner dans l'environnement chroot et de monter $LFS/proc de nouveau (comme indiqué plus tard) avant de continuer dans le livre.

Notez que l'invite de commande de bash contiendra "I have no name!" (je n'ai pas de nom!) ce qui est normal puisque Glibc n'a pas encore été installé.

# Changer de propriétaire

La première chose que nous allons faire, maintenant que nous sommes *root* est de changer le propriétaire des fichiers et répertoires installés dans le chapitre 5 pour l'utilisateur root; parce que, si plus tard nous ne supprimons pas le répertoire `/static` et commençons à ajouter de nouveaux utilisateurs, un de ces utilisateurs pourraient posséder les programmes liés statiquement, ce qui n'est pas une bonne idée.

Lancez la commande suivante pour que root soit le propriétaire de tous les programmes liés statiquement :

```
chown -R 0:0 /static
```

Cette commande utilise 0:0 au lieu de root.root, car il n'existe aucun moyen de "résoudre" le nom root, puisque glibc n'a pas été encore installée.

# Création des répertoires

Créons maintenant la hiérarchie de répertoires sur notre système de fichiers LFS. Lancer les commandes suivantes pour créer une hiérarchie plus ou moins standard des répertoires:

```
mkdir -p /{bin,boot,dev/pts,etc/opt,home,lib,mnt,proc} &&
mkdir -p /{root,sbin,tmp,usr/local,var,opt} &&
for dirname in /usr /usr/local
    do
    mkdir $dirname/{bin,etc,include,lib,sbin,share,src}
    ln -s share/{man,doc,info} $dirname
    mkdir $dirname/share/{dict,doc,info,locale,man}
    mkdir $dirname/share/{nls,misc,terminfo,zoneinfo}
    mkdir $dirname/share/man/man{1,2,3,4,5,6,7,8}
done &&
mkdir /var/{lock,log,mail,run,spool} &&
mkdir -p /var/{tmp,opt,cache,lib/misc,local} &&
mkdir /opt/{bin,doc,include,info} &&
mkdir -p /opt/{lib,man/man{1,2,3,4,5,6,7,8}} &&
ln -s ../var/tmp /usr
```

Par défaut, les répertoires sont créés avec le mode de permission 755, ce qui n'est pas souhaité pour tous les répertoires. Nous allons réaliser deux changements: un pour le répertoire principal de root et un autre pour les répertoires des fichiers temporaires.

```
chmod 0750 root &&
chmod 1777 tmp var/tmp
```

Le premier changement de permission nous aussre que personne ne pourra entrer dans le répertoire `/root` (de façon identique à un utilisateur pour son répertoire principal). Le deuxième changement nous assure que tout utilisateur peut écrire dans `/tmp` et `/var/tmp`, mais ne peut pas supprimer des fichiers d'autres utilisateurs. Cette dernière permission est assuré par le "sticky bit", bit 1 du masque 1777.

Maintenant que les répertoires sont créés, déplacez les fichiers tar des sources téléchargés au chapitre 3 dans un sous−répertoire de `$LFS/usr/src` (vous devrez créer ce répertoire vous−même).

# Notes de conformité FHS

Nous avons basé notre arbre de répertoires sur le standard FHS (disponible sur http://www.pathname.com/fhs/). En dehors de cette arborescence, ce standard stipule l'existence de `/usr/local/games` et `/usr/share/games`, mais nous ne voyons pas l'intérêt de ceux−ci pour un système de base. Néanmoins, rendez votre système compatible FHS si vous le souhaitez. En ce qui concerne la structure du sous−répertoire `/usr/local/share`, le FHS est inprécis, donc nous créons ici les répertoires dont nous pensons avoir besoin.

# Monter le système de fichier proc

Pour que certans programmes fonctionnent correctement, le système de fichiers proc doit être monté et disponible aussi depuis l'environnement chrooté. Un système de fichiers peut être monté autant de fois et à autant d'emplacements différents que vous le souhaitez. Donc, ce n'est pas un problème que le système de fichiers proc soit déjà monté sur votre système hôte, surtout parce que proc est un système de fichiers virtuel.

Le système de fichiers proc est monté sous `/proc` en lançant la commande suivante.

```
mount proc /proc –t proc
```

Vous pouvez obtenir des messages d'avertissement de la commande mount, tels que:

```
warning: can't open /etc/fstab: No such file or directory
not enough memory
```

Ignorez–les, ils sont dûs au fait que le système n'est pas encore installé complètement et que quelques fichiers manquent. Le montage lui–même sera fait avec succès et c'est tout ce dont nous avons besoin à ce moment.

# Créer le lien symbolique /etc/mtab

La prochaine étape est de créer un lien symbolique de /etc/mtab sur /proc/mounts. Ceci est fait en lançant la commande suivante

```
ln -sf /proc/mounts /etc/mtab
```

La création de ce lien symbolique permet d'éviter des problèmes survenant si / est monté en lecture seule et que l'information dans /etc/mtab n'est plus à jour. En créant ce lien symbolique sur /proc/mounts, nous nous assurons que /etc/mtab sra toujours à jour.

Notez qu'utiliser ce lien symbolique nécessite que le support du système de fichiers /proc soit compilé dans votre noyau. Ceci est fait par défaut, et ne doit pas être changé sauf si vous savez *réellement* ce que vous êtes en train de faire car bien plus de choses dépendent de la présence de /proc. Bref, assurez−vous de la présence du support du système de fichiers /proc dans votre noyau.

# Création des liens symboliques bash et sh

Certains programmes ont enregistré en dur des chemins vers des programmes encore non existants. Pour satisfaire ces programmes, nous créons les liens symboliques `/bin/bash` et `/bin/sh` tous deux pointant vers le programme statique `bash`.

Créez les liens symboliques `/bin/bash` et `/bin/sh` en lançant les commandes suivantes:

```
ln -s /static/bin/bash /bin/bash &&
ln -s bash /bin/sh
```

# Créer les fichiers passwd et group

Pour que root puisse se connecter et pour que le nom "root" soit reconnue, il doit exister les entrées adéquates dans les fichiers `/etc/passwd` et `/etc/group`.

Créez le fichier `/etc/passwd` en lançant la commande suivante:

```
echo "root:x:0:0:root:/root:/bin/bash" > /etc/passwd
```

Le mot de passe actuel pour root (le "x" ici sert juste à remplir la case) sera initialisé plus tard.

Créez le fichier `/etc/group` en lançant la commande suivante:

```
cat > /etc/group << "EOF"
root:x:0:
bin:x:1:
sys:x:2:
kmem:x:3:
tty:x:4:
tape:x:5:
daemon:x:6:
floppy:x:7:
disk:x:8:
lp:x:9:
dialout:x:10:
audio:x:11:
EOF
```

Les groupes créés ne font aprtie d'aucun standard −− ce sont les groupes que le script MAKEDEV utilise dans la section suivante. En plus du groupe "root", le LSB recommende seulement un groupe "bin", avec un GID de 1. Tous les autres noms de groupe et GIDs peuvent être choisis librement par l'utilisateur, car les packages bien écrits ne dépendent pas du numéro GID mais utilise le nom du groupe.

# Créer les périphériques (Makedev–1.7)

```
Estimation du temps de construction:           0.07 SBU
Estimation de l'espace disque requis:  50 Ko
```

## Créer les périphériques

(Notez que déballer le fichier MAKEDEV–1.7.bz2 ne créera pas de répertoire dans lequel vous pourriez entrer, car ce fichier ne contient qu'un script.)

Préparez la création des fichiers périphériques en lançant les commandes suivantes:

```
cp MAKEDEV-1.7 /dev/MAKEDEV &&
cd /dev &&
chmod 754 MAKEDEV &&
```

La plupart de gens voudront maintenant créer les ficheirs périphériques en lançant la commande:

```
./MAKEDEV -v generic
```

Mais si vous comptez utiliser devpts, alors lancez ceci à la place:

```
./MAKEDEV -v generic-nopty
```

Notez que si vous n'êtes pas sûr, il vaut mieux utiliser la commande **`./MAKEDEV -v generic`** qui assure que vous disposez des périphériques dont vous avez besoin. Si vous êtes toutefois sûr que vous allez utiliser devpts, l'autre commande assure que vous ne créiez pas un jeu de périphériques dont vous n'avez pas besoin.

MAKEDEV créera hda[1–20] à hdh[1–20] et d'autres périphériques de la sorte, mais gardez à l'esprit que vous ne pourrez pas tous les utiliser à cause des limitations du noyau sur le nombre maximum de partitions.

## Explication des commandes

**`./MAKEDEV -v generic:`** Ceci crée les périphériques. Normalement, ces périphériques sont tous ceux dont vous avez besoin. Il est possible que certains périphériques spéciaux nécessaires pour votre configuration matérielle vous manquent. Créez–les avec ./MAKEDEV –v <device>. L'option **`generic-nopty`** crée les mêmes périphériques que **`generic`**, mais ne s'occupe pas de ceux qui ne vous sont pas nécessaires si vous utilisez devpts.

## Contenu de MAKEDEV

Dernière vérification effectuée auprès de la version 1.5.

### Programmes

MAKEDEV

### Descriptions

### MAKEDEV

`MAKEDEV` est un script qui crée les fichiers périphériques statiques nécessaires, qui se trouvent généralement dans le répertoire `/dev`. Plus d'informations sur les fichiers périphériques sont disponibles à partir de la hiérarchie des sources du noyau dans le fichier `Documentation/devices.txt`.

## Dépendances d'installation de MAKEDEV

Dernière vérification effectuée auprès de la version 1.5.

Bash: sh Fileutils: chmod, chown, cp, ln, mknod, mv, rm Grep: grep Sh−utils: expr, id

# Installer Linux–2.4.19

```
Estimation du temps de construction:           0.02
Estimation de l'espace disque requis:  142 Mo
```

## Installation du noyau Linux

Nous ne compilerons pas encore un nouveau noyau –– nous le ferons quand nous aurons fini l'installation de tous les packages. Mais comme certains packages ont besoin des fichiers d'entête du noyau, nous allons déballer l'archive du noyau maintenant, le configurer et copier les fichiers d'entête où ils pourront être trouvés par ces packages.

Les entêtes du noyau sont copiées en lançant les commandes suivantes:

```
ln -s /static/bin/pwd /bin/pwd &&
make mrproper &&
make include/linux/version.h &&
make symlinks &&
mkdir /usr/include/asm &&
cp include/asm/* /usr/include/asm &&
cp -R include/asm-generic /usr/include &&
cp -R include/linux /usr/include &&
touch /usr/include/linux/autoconf.h &&
rm /bin/pwd
```

## Explication des commandes

**`ln -s /static/bin/pwd /bin/pwd`**: Les sources du noyau enregistrent en dur le chemin vers la commande `pwd`, c'est–à–dire `/bin/pwd` donc nous créons un lien symbolique temporaire pour gérer cela.

**`make mrproper`:** Ceci s'assure que l'arborescence du noyau est parfaitement propre. Nous faisons ceci parce que l'équipe de développement du noyau recommande que ceci soit fait avant *chaque* compilation du noyau, et que nous ne devons pas supposer que l'arborescence des sources soit propre après l'avoir déballé.

**`make include/linux/version.h`** et **`make symlinks`**: Ceci crée le fichier `include/linux/version.h`, ainsi que le lien symbolique spécifique à la plateforme `include/asm`.

**`mkdir /usr/include/asm`**, **`cp include/asm/* /usr/include/asm`** et **`cp -R include/asm-generic /usr/include`**: Ces commandes copient les fichiers d'entête assembleur du noyau dans `/usr/include/asm` et `/usr/include/asm-generic`.

**`cp -R include/linux /usr/include`**: Cette commande copie les fichiers d'entête pour la cross–compilation du noyau dans `/usr/include`.

**`touch /usr/include/linux/autoconf.h`**: Quelque fichiers d'entête du noyau incluent le fichier `autoconf.h`, mais en dehors de la hiérarchie des fichiers source du noyau Linux, ce fichier n'apporte rien donc nous en créons un vide de manière à ne pas avoir d'erreurs de compilation lorsqu'il est une dépendance pour un autre fichier d'entête du noyau.

## Pourquoi nous copions les entêtes du noyau et pourquoi nous ne créons pas de liens

Auparavant, une pratique commune consistait à créer des liens symboliques pour les répertoires `/usr/include/{linux,asm}` vers respectivement `/usr/src/linux/include/{linux,asm}`. Ceci est une *mauvaise* idée d'après cet extrait d'un message de Linus Torvalds sur la liste de diffusion du noyau Linux:

```
Je suggère que les personnes qui compilent des noyaux devraient:

 - ne pas créer un seul lien symbolique (sauf celui créé lors de la
construction du noyau, "linux/include/asm" qui est utilisé pour la compilation
du noyau lui-même)

Et oui, c'est ce que je fais. Mon répertoire /usr/src/linux a toujours les
anciens entêtes du noyau 2.2.13, même si je n'ai pas lancé cette version du
noyau depuis un _loong_ moment. Mais glibc a été compilé avec, donc ces
entêtes correspondent aux objets de la bibliothèque.

Et cela correspond à l'environnement suggéré depuis au moins les cinq dernières
années. Je ne sais pas pourquoi l'idée du lien symbolique est toujours vivante,
comme un mauvais zombie. Pratiquement toutes les distributions conservent
l'idée du lien et tout le monde se souvient que les sources du noyau doivent
aller sous "/usr/src/linux" même si ce n'est plus vrai depuis _trèès_
longtemps.
```

La partie importante là–dedans correspond au moment où il indique que les entêtes doivent être *ceux avec lesquels glibc a été compilé*. Ces entêtes doivent rester accessibles et en les copiant, nous nous assurons de suivre ces recommandations. Notez aussi que tant que ces liens symboliques ne sont pas créés, il est tout à fait correct d'avoir les sources du noyau sous `/usr/src/linux`.

La partie essentielle se trouve là où Linus indique que les fichiers d'entête doivent être *ceux avec lesquels gblic a été compilé*. Ces entêtes doivent être utilisés plus tard lorsque vous compilerez d'autres packages, car ce sont eux qui représentent les fichiers de bibliothèques. En copiant les entêtes, nous nous assurons qu'ils restent disponibles plus tard lors d'une mise à jour du noyau.

Notez qu'il est parfaitement normal d'avoir les sources du noyau dans `/usr/src/linux`, aussi longtemps que vous n'avez pas les liens symboliques `/usr/include/{linux,asm}`.

# Contenu de Kernel

Dernière vérification effectuée auprès de la version 2.4.18.

## Programmes

Le noyau linux et les entêtes du noyau linux

## Descriptions

### Noyau linux

Le noyau Linux est au coeur de chaque système Linux. C'est lui qui fait tourner Linux. Quand vous allumez votre ordinateur et démarrez un système Linux, la toute première partie de logiciel Linux qui est chargée est le noyau. Le noyau initialise les composants matériels du système comme les ports série, les ports parallèles, les cartes son, les cartes réseau, les contrôleurs IDE, les contrôleurs SCSI et beaucoup d'autres choses. En bref, le noyau rend le matériel disponible pour que le logiciel puisse fonctionner.

### En−têtes du noyau linux

Nous copions ces fichiers dans /usr/include/{linux,asm} lors du chapitre 5. Ils doivent correspondre à ceux avec lesquels glibc a été compilé et ne doivent *pas* être remplacé lors d'une mise à jour du kernel. Ils sont essentiels pour compiler un grand nombre de logiciels.

## Dépendances d'installation de Linux

Dernière vérification effectuée auprès de la version 2.4.17.

Bash: sh Binutils: ar, as, ld, nm, objcopy Fileutils: cp, ln, mkdir, mv, rm, touch
Findutils: find, xargs Gcc: cc1, collect2, cpp0, gcc Grep: grep Gzip: gzip Make: make Gawk: awk
Modutils: depmod, genksyms Net−tools: dnsdomainname, hostname Sed: sed
Sh−utils: basename, date, expr, pwd, stty, uname, whoami, yes Textutils: cat, md5sum, sort, tail

# Installer Man−pages−1.52

```
Estimation du temps de construction:           0.01 SBU
Estimation de l'espace disque requis:  6 Mo
```

# Installation de Man−pages

Installez Man−pages en lançant la commande suivante:

```
make install
```

# Contents of Man−pages

Dernière vérification effectuée auprès de la version 1.52.

## Fichiers support

Plusieurs pages de manuel qui ne viennent pas avec les packages.

## Descriptions

### manual pages

Les exemples de pages de manuel fournis décrivent toutes les fonctions C et C++, quelques fichiers /dev/ importants et autres.

# Dépendances d'installation de Man−pages

Dernière vérification effectuée auprès de la version 1.47.

Bash: sh Fileutils: install Make: make

# Installer Glibc–2.2.5

```
Estimation du temps de construction:           14.71 SBU
Estimation de l'espace disque requis:  369 Mo
```

## Installation de Glibc

Ce package nécessite que son correctif lui soit appliqué avant de pouvoir l'installer. Soyez sûr qu'il a été déballé avant de lancer les commandes d'installation.

Avant d'installer glibc, vous devez vous positionner dans le répertoire glibc–2.2.5 et y déballer glibc–linuxthreads, et non pas dans /usr/src comme vous le feriez normalement.

Ce package est connu pour avoir un mauvais fonctionnement si vous avez changé ses paramètres d'optimisation par défaut (incluant les options –march et –mcpu). Glibc se comporte mieux sans. Donc, si vous avez défini des variables d'environnement qui écrasent les optimisations par défaut, comme CFLAGS et CXXFLAGS, nous vous recommendons de les annuler ou de les modifier lors de la construction de Glibc. Vous avez été prévenu.

De même, ne donnez pas l'option ––enable–kernel au script de configuration. C'est connu pour générer des segmentation faults lorsque des packages comme fileutils, make et tar sont liés avec lui.

Généralement, compiler Glibc de n'importe quel autre façon que ce que suggère ce livre, impose un risque très important sur votre système.

Installez Glibc en lançant les commandes suivantes:

```
patch -Np1 -i ../glibc-2.2.5-2.patch &&
touch /etc/ld.so.conf &&
mkdir ../glibc-build &&
cd ../glibc-build &&
../glibc-2.2.5/configure --prefix=/usr --disable-profile \
    --enable-add-ons --libexecdir=/usr/bin &&
echo "cross-compiling = no" > configparms &&
make &&
make install &&
make localedata/install-locales &&
exec /static/bin/bash --login
```

Une alternative à l'exécution de **`make localedata/install-locales`** est de n'installer que les locales dont vous avez besoin ou que vous voulez. Cela se fait en exécutant la commande localedef. Vous trouverez des informations à ce sujet dans le fichier INSTALL de l'arborescence de glibc–2.2.5. Le programme **`localedef`** assume que le répertoire `/usr/lib/locale` existe, donc vous avez besoin de le créer auparavant.

Les pages man Linux Threads ne vont pas être installées maintenant car elles requièrent une installation fonctionnelle de Perl. Nous installerons Perl plus loin dans ce chapitre, donc nous reviendrons à l'installation des pages man Linux Threads après cela.

Durant l'étape de configuration, vous allez voir ces avertissements:

```
configure: warning:
```

```
*** These auxiliary programs are missing or too old: msgfmt
*** some features will be disabled.
*** Check the INSTALL file for required versions.
```

Le msgfmt manquant (du package gettext que nous installerons plus tard dans ce chapitre) ne gêne en rien. msgfmt est utilisé pour générer les fichiers de traductions binaires qui sont utilisés pour que votre système parle dans un langage différent. Parce que ces fichiers ont déjà été généré pour vous, nous n'avons pas besoin de msgfmt. Vous en aurez seulement besoin si vous faites des changements dans les fichiers sources (les fichiers `*.po` situés dans le répertoire `po`) qui vous imposeraient de recompiler les binaires.

# Explication des commandes

**`patch -Np1 -i ../glibc-2.2.5-2.patch :`** Ce correctif convertit toutes les occurences de `$(PERL)` en `/usr/bin/perl` dans le fichier `malloc/Makefile`. Ceci est fait parce que Glibc ne peut pas détecter automatiquement l'emplacement de perl, celui-ci n'ayant pas encore été installé. Ce correctif remplace aussi toutes les occurences du mot *root* par *0* dans le fichier `login/Makefile`. Ceci est fait parce que Glibc elle-même n'est pas encore installée et donc, la résolution des noms d'utilisateurs en identifiants ne fonctionne pas pour l'instant. Un **`chown root fichier`** échouerait, mais fonctionnerait en utilisant directement les numérosd'identifiant.

Ce correctif contient aussi quelques corrections de bugs et de trous de sécurité. En particulier, il contient les corrections "errlist", "dns resolver", "xdr_array", "calloc", "thread exit", "udivdi3", "math test", "restrict_arr" and "divbyzero", qui sont documentés sur http://www.zipworld.com.au/~gschafer/lfs-tweaks.html.

**`touch /etc/ld.so.conf :`** Une des étapes finales de l'installation de Glibc est de lancer ldconfig pour mettre à jour le cache du chargeurdynamique. Si ce fichier n'existe pas, l'installation s'interrompera avec uneerreur de lecture du fichier, donc nous créons un fichier vide (le fichier videcontiendra les valeurs par défaut pour Glibc en utilisant /lib et /usr/lib,ce qui est suffisant).

**`--disable-profile :`** Ceci désactive la construction des bibliothèques avec des informations pour l'évaluation de performances. Cette commande peut être omise si vous souhaitez travailler sur l'évaluation de performances.

**`--enable-add-ons :`** Ceci active le complément que nous installons avec Glibc, linuxthreads.

**`--libexecdir=/usr/bin :`** Cela force l'installation du programme pt_chown dans le répertoire /usr/bin.

**`echo "cross-compiling = no" > configparms :`** Nous faisons ceci parce que nous créons seulement notre propre système. La cross-compilation est utilisée, par exemple, pour construire un package pour un ApplePower PC sur un système Intel. La raison pour laquelle Glibc pense que nousfaisons de la cross-compilation est qu'il ne peut compiler un programme de testpour s'en apercevoir, donc il est par défaut en cross-compilation. Compiler le programme de test échoue car Glibc n'a pas encore été installée.

**`exec /static/bin/bash --login :`** Cette commande va démarrer un nouveau shell bash qui va remplacer l'actuel. Cela supprime le message "I have no name!" de l'invite de commande qui est dû à l'incapacité de bash à faire correspondre un ID utilisateur avec un nom utilisateur (qui est dû au fait que Glibc n'est pas installée).

# Contenu de Glibc

Dernière vérification effectuée auprès de la version 2.2.5.

## Programmes

catchsegv, gencat, getconf, getent, glibcbug, iconv, iconvconfig, ldconfig, ldd, lddlibc4, locale, localedef, mtrace, nscd, nscd_nischeck, pcprofiledump, pt_chown, rpcgen, rpcinfo, sln, sprof, tzselect, xtrace, zdump et zic

## Descriptions

### catchsegv

catchsegv peut être utilisé pour créer une trace de la pile lorsqu'un programme s'interrompt avec une erreur 'segmentation fault'.

### gencat

gencat génère des catalogues de messages.

### getconf

getconf affiche les valeurs de configuration du système pour les variables spécifiques aux systèmes de fichiers.

### getent

getent récupère des entrées d'une base de données administrative.

### glibcbug

glibcbug crée un rapport de bug sur glibc et l'envoie par courrier électronique à l'adresse email pour les bugs.

### iconv

iconv réalise des conversions de jeux de caractères.

### iconvconfig

iconvconfig crée un fichier de configuration chargé rapidement pour le module iconv.

### ldconfig

ldconfig crée un cache des bibliothèques dynamiques pour l'éditeur de liens dynamiques.

### ldd

ldd affiche les bibliothèques partagées requis par chaque programme ou bibliothèque partagée spécifié sur la ligne de commande.

### lddlibc4

lddlibc4 assiste ldd avec les fichiers objets.

### locale

locale est un programme Perl indiquant au compilateur d'autoriser (ou non) l'utilisation des variables locales POSIX pour les opérations intégrées.

### localedef

localedef compile les spécifications pour locale.

### mtrace

mtrace affiche les chemins 'multicast' d'une source à un récepteur (une requête des traces IP).

### nscd

nscd est un démon procurant un cache pour les requêtes DNS les plus courantes.

### nscd_nischeck

nscd_nischeck vérifie si un mode de sécurité est nécessaire pour les recherches NIS+.

### pcprofiledump

pcprofiledump affiche l'information générée par 'PC profiling'.

### pt_chown

pt_chown indique l'utilisateur, le groupe et les permissions d'accès au terminal du pseudo terminal esclave correspondant au pseudo terminal maître passé sur le descripteur de fichier 3. C'est le programme d'aide pour la fonction 'grantpt'. Il n'a pas été conçu pour être lancé directement en ligne de commande.

### rpcgen

rpcgen génère le code C pour implémenter le protocole RPC.

### rpcinfo

rpcinfo fait un appel RPC vers un serveur RPC.

### sln

sln crée le lien symbolique entre une source et sa destination. C'est lié statiquement, sans nécessiter de lien dynamique. Donc, sln est utilisé pour créer des liens symboliques avec des bibliothèques dynamiques si, pour quelque raison que ce soit, le système de liens dynamiques n'était pas fonctionnel.

### sprof

sprof lit et affiche les données d'objets partagés profilés.

### tzselect

tzselect questionne l'utilisateur sur sa position géographique courante et affiche la description de fuseau horaire résultante sur la sortie standard.

### xtrace

xtrace trace l'exécution de programmes en affichant la fonction actuellement exécutée.

### zdump

zdump est le dumper du fuseau horaire.

### zic

zic est le compilateur de fuseau horaire.

## Bibliothèques

ld.so, libBrokenLocale.[a,so], libBrokenLocale_p.a, libSegFault.so, libanl.[a,so], libanl_p.a, libbsd−compat.a, libc.[a,so], libc_nonshared.a, libc_p.a, libcrypt.[a,so], libcrypt_p.a, libdl.[a,so], libdl_p.a, libg.a, libieee.a, libm.[a,so], libm_p.a, libmcheck.a, libmemusage.so, libnsl.a, libnsl_p.a, libnss_compat.so, libnss_dns.so, libnss_files.so, libnss_hesiod.so, libnss_nis.so, libnss_nisplus.so, libpcprofile.so, libpthread.[a,so], libpthread_p.a, libresolv.[a,so], libresolv_p.a, librpcsvc.a, librpcsvc_p.a, librt.[a,so], librt_p.a, libthread_db.so, libutil.[a,so] et libutil_p.a

## Descriptions

### ld.so

ld.so est le programme d'aide pour les exécutables utilisant des bibliothèques partagées.

### libBrokenLocale, libBrokenLocale_p

Utilisé par des logiciels, tels que Mozilla, pour résoudreles locales manquantes.

### libSegFault

libSegFault est un gestionnaire pour le signal 'segmentation fault'. Il essaie de capturer ces signaux.

### libanl, libanl_p

libanl est une bibliothèque de recherche de 'asynchronous name'.

### libbsd−compat

libbsd−compat procure une portabilité nécessaire pour lancer certains programmes sous Linux.

### libc, libc_nonshared, libc_p

Ces fichiers constituent la principale bibliothèque C. Celle−ci est une collection de fonctions couramment utilisées dans les programmes. Cela évite au développeur d'écrire ses propres fonctions pour toutes sortes de tâches. Les plus communes telles que l'affichage d'une chaîne de caractères à l'écran sont déjà présentes et à disposition du programmeur.

La bibliothèque C (comme presque toutes les bibliothèques) est disponible en deux versions : dynamique et statique. En résumé, lorsqu'un programme utilise une bibliothèque C statique, le code de cette bibliothèque est copié dans l'exécutable. Lorsqu'un programme utilise une bibliothèque dynamique, cet exécutable ne contient pas le code de la bibliothèque, mais une routine qui charge la fonction de la bibliothèque au moment où le programme l'utilise. Cela réduit d'une façon significative la taille du programme. La documentation fournie avec la bibliothèque C décrit ce mécanisme plus en détails, il est trop compliqué de l'expliquer ici en une ou deux lignes.

### libcrypt, libcrypt_p

libcrypt est la bibliothèque de cryptographie.

### libdl, libdl_p

libdl est la bibliothèque d'interface pour le chargeur dynamique de liens.

### libg

libg est une bibliothèque de lancement pour g++.

### libieee

libieee est la bibliothèque des nombres flottants IEEE.

### libm, libm_p

libm est la bibliothèque de mathématiques.

### libmcheck

libmcheck contient du code exécuté au démarrage.

### libmemusage

libmemusage est utilisé par memusage pour aider à récupérer des informations sur l'utilisation de la mémoire par un programme.

### libnsl, libnsl_p

libnsl est la bibliothèque des services réseau.

### libnss_compat, libnss_dns, libnss_files, libnss_hesiod, libnss_nis, libnss_nisplus

L'idée principale est de mettre l'implémentation des différents services offerts pour accéder aux bases de données dans des modules séparés. Ceci a un certain nombre d'avantages:

- les contributeurs peuvent ajouter de nouveaux services sans les ajouter à la bibliothèque GNU C.
- les modules peuvent être mis à jour séparement.
- l'image de la bibliothèque C est plus petite.

### libpcprofile

Code utilisé par le noyau pour surveiller le temps CPU passé dans les fonctions, les lignes de codes sources et les instructions.

### libpthread, libpthread_p

La bibliothèques POSIX des threads.

### libresolv, libresolv_p

Les fonctions de cette bibliothèque permettent la création, l'envoi et l'interprétation des paquets des serveurs de noms Internet.

### librpcsvc, librpcsvc_p

Les fonctions de cette bibliothèque apportent différents services RPC.

### librt, librt_p

Les fonctions de cette bibliothèque apportent la plupart des interfaces spécifiées par l'extension temps réel POSIX.1b.

### libthread_db

Les fonctions de cette bibliothèque sont utiles pour construire des debuggers pour les programmes multi−tâches.

### libutil, libutil

Contient du code pour les fonctions standards utilisé pour les différents utilitaires Unix.

# Dépendances d'installation de Glibc

Dernière vérification effectuée auprès de la version 2.2.5.

Bash: sh Binutils: ar, as, ld, ranlib, readelf Diffutils: cmp
Fileutils: chmod, cp, install, ln, mknod, mv, mkdir, rm, touch Gcc: cc, cc1, collect2, cpp, gcc
Grep: egrep, grep Gzip: gzip Make: make Gawk: gawk Sed: sed
Sh−utils: date, expr, hostname, pwd, uname Texinfo: install−info, makeinfo
Textutils: cat, cut, sort, tr

# Installer GCC–3.2

```
Estimation du temps de construction:           13.26 SBU
Estimation de l'espace disque requis:  221 Mo
```

## Installation de GCC

Ce package nécessite que son correctif soit appliqué avant que vous puissiez l'installer. Assurez–vous qu'il a bien été déballé avant de lancer les commandes d'installation.

Ce package est connu pour avoir un mauvais fonctionnement si vous avez changé ses paramètres d'optimisation par défaut (incluant les options –march et –mcpu). GCC se comporte mieux sans. Donc, si vous avez défini des variables d'environnement qui écrasent les optimisations par défaut, comme CFLAGS et CXXFLAGS, nous vous recommendons de les annuler ou de les modifier lors de la construction de GCC. Vous avez été prévenu.

Installer GCC en exécutant les commandes suivantes. Ces commandes construiront les compilateurs C et C++. D'autres compilateurs sont disponibles dans le package gcc. Si vous voulez également construire tous les autres compilateurs disponibles, enlevez l'option ––enable–languages=c,c++ dans la commande de configuration. Consultez la documentation de GCC pour plus de détails sur lescompilateurs disponibles.

Note : la construction des autres compilateurs n'est pas testée par les personnes qui travaillent sur LFS.

```
patch -Np1 -i ../gcc-3.2.patch &&
mkdir ../gcc-build &&
cd ../gcc-build &&
../gcc-3.2/configure --prefix=/usr --enable-shared \
    --enable-languages=c,c++ --enable-threads=posix \
    --with-slibdir=/lib --enable-__cxa_atexit \
    --enable-clocale=gnu &&
make bootstrap &&
make install &&
ln -s ../usr/bin/cpp /lib &&
ln -s ../bin/cpp /usr/lib &&
ln -s gcc /usr/bin/cc
```

## Explication des commandes

**`--enable-threads=posix`:** Ceci active la gestion des exceptions en C++ pour du code multithreadé.

**`--enable-__cxa_atexit`:** Cette option résultera en des bibliothèques dynamiques C++ et des programmes C++, interopérables avec d'autres distributions linux.

**`--enable-clocale=gnu`:** Certaines personnes courent le risque de construire des bibliothèques C++ incompatibles avec ABI si ils n'installent pas tout la partie localedata de glibc. Utiliser ––enable–clocale=gnu permet de s'assurer que tout est bien fait dans tous les cas. Si vous ne voulez pas utiliser cette option, alors construisez au moins la locale *de_DE*. Quand GCC trouve cette locale spécifique, alors le mode locale correct (*gnu*) est implémenté.

## Contenu de GCC

Dernière vérification effectuée auprès de la version 3.1.

## Programmes

c++, c++filt, cc (lien vers gcc), cc1, cc1plus, collect2, cpp, cpp0, g++, gcc, gccbug, gcov, tradcpp0

## Descriptions

### cc, cc1, cc1plus, gcc

Ce sont les compilateurs C. Un compilateur transforme le code source au format texte dans un format que l'ordinateur comprends. Après qu'un code source ait été compilé en un fichier objet, un éditeur de lien va créer un fichier exécutable à partir d'un ou plusieurs de ces fichiers générés par le compilateur.

### c++, cc1plus, g++

Ce sont les compilateurs C++, l'équivalent de cc et gcc, etc...

### c++filt

Le langage C++ procure la surcharge de fonctions, c'est−à−dire qu'il est possible d'écrire plusieurs fonctions possédant le même nom, à condition que chaque fonction utilise des paramètres de différents types. Tous les noms des fonctions C++ sont codés avec un label assembleur bas niveau (ce procédé est aussi connu sous le nom de mangling). Le programme c++filt réalise l'opération inverse; il décode les noms bas niveau en des noms de niveau utilisateur, de manière à ce que l'éditeur de liens ne confonde pas les fonctions surchargées.

### collect2

collect2 assiste à la compilation de constructeurs.

### cpp, cpp0

cpp réalise un travail préliminaire sur un fichier source, comme inclure le contenu des fichiers d'entêtes dans le code source. Il suffit d'insérer une ligne comme #include <filename> dans votre fichier source. Le préprocesseur va insérer le contenu de ce fichier dans le fichier source.

### gccbug

gccbug est un script shell, utilisé pour simplifier la création de rapports de bugs.

### gcov

gcov analyse les programmes pour aider à créer des codes plus efficaces, plus rapides avec une optimisation.

### tradcpp0

Aucune description actuellement disponible.

## Bibliothèques

libgcc.a, libgcc_eh.a, libgcc_s.so, libiberty.a, libstdc++.[a,so], libsupc++.a

## Descriptions

### libgcc, libgcc_eh, libgcc_s

Fichiers de support pour gcc.

### libiberty

libiberty est une collection de sous−routines utilisées par différents programmes GNU, comme getopt, obstack, strerror, strtol et strtoul.

### libstdc++

libstdc++ est la bibliothèque C++. Elle est utilisée par les programmes C++ et contient des fonctions couramment utilisées par eux. De cette façon, le développeur n'a pas besoin d'écrire certaines fonctions (comme écrire une ligne de texte à l'écran) à partir de rien, à chaque fois qu'il crée un programme.

### libsupc++

libsup++ procure une aide au langage de programmation c++. Entre autres choses, libsup++ contient des routines pour la gestion des exceptions.

# Dépendances d'installation de GCC

Dernière vérification effectuée auprès de la version 2.95.3.

Bash: sh Binutils: ar, as, ld, nm, ranlib Diffutils: cmp
Fileutils: chmod, cp, ln, ls, mkdir, mv, rm, touch Find: find Gcc: cc, cc1, collect2, cpp0, gcc
Grep: egrep, grep Make: make Patch: patch Sed: sed
Sh−utils: basename, dirname, echo, expr, hostname, sleep, true, uname Tar: tar
Texinfo: install−info, makeinfo Textutils: cat, tail, tr

# Installer Zlib–1.1.4

```
Estimation du temps de construction:           0.07 SBU
Estimation de l'espace disque requis:  1 Mo
```

## Installation de Zlib

Installez Zlib en lançant les commandes suivantes:

```
./configure --prefix=/usr --shared &&
make LIBS="libz.so.1.1.4 libz.a" &&
make LIBS="libz.so.1.1.4 libz.a" install &&
cp zlib.3 /usr/share/man/man3
```

## Contenu de Zlib

Dernière vérification effectuée auprès de la version 1.1.4.

### Bibliothèques

libz[a,so]

### Descriptions

#### libz

Il s'agit de la bibliothèque zlib, utilisée par de nombreux programmes pour ses fonctions de compression/décompression.

## Dépendances d'installation de Zlib

Aucune dépendance encore vérifiée.

# Installer Findutils–4.1

```
Estimation du temps de construction:           0.10 SBU
Estimation de l'espace disque requis:  3 Mo
```

## Installer Findutils

Ce package nécessite que son correctif soit appliqué avant que vous puissiez l'installer. Assurez–vous qu'il soit déballé avant de lancer les commandes d'installation.

Installez Findutils en exécutant les commandes suivantes :

```
patch -Np1 -i ../findutils-4.1.patch &&
./configure --prefix=/usr &&
make libexecdir=/usr/bin &&
make libexecdir=/usr/bin install
```

## Note à propos de la conformité FHS

Par défaut, la localisation de la base de données de updatedb est dans /usr/var. Si vous préférez être conforme avec FHS, vous devrez utiliser une autre localisation. La commande suivante utilise le fichier de base de données `/var/lib/misc/locatedb` qui est compatible FHS.

```
patch -Np1 -i ../findutils-4.1.patch &&
./configure --prefix=/usr &&
make localstatedir=/var/lib/misc libexecdir=/usr/bin &&
make localstatedir=/var/lib/misc libexecdir=/usr/bin install
```

## Explication des commandes

**`patch -Np1 -i ../findutils-4.1.patch`**: Ce correctif est là pour réparer quelques erreurs de compilation en évitant des conflits de variables et en changeant de mauvaises syntaxes.

## Contenu du package Findutils

Dernière vérification effectuée auprès de la version 4.1.

### Programmes

bigram, code, find, frcode, locate, updatedb et xargs

### Descriptions

#### bigram

bigram est utilisé avec la commande code pour produire des bases de données de localisation ancienne formule. Pour en savoir plus sur ces trois derniers programmes, lire les pages de manuel updatedb.5.

### code

code est l'ancêtre de frcode. Il était utilisé dans les anciennes bases de données de localisation.

### find

Le programme find recherche des fichiers dans une arborescence selon certains critères. Si aucun critère n'est défini, il liste tous les fichiers du répertoire courant et de ses sous−répertoires.

### frcode

frcode est lancé par updatedb pour compresser la liste des fichiers de façon à réduire la taille de la base de données d'un facteur 4 à 5.

### locate

locate scanne une base de données contenant tous les fichiers et répertoires d'un système de fichiers. Ce programme liste les fichiers et répertoires de cette base de données selon certains critères de concordance. Si un utilisateur recherche un fichier, ce programme scannera la base de données pour lui donner l'emplacement exact où ces fichiers sont situés. Ceci n'a de sens que si la base de données est régulièrement mise à jour, autrement cela ne fournirait que des informations dépassées.

### updatedb

Le programme updatedb met à jour la base de données de localisation. Il parcourt entièrement les systèmes de fichiers (sans contre indications, il parcourt tous les systèmes de fichiers montés) et met à jour la base de données qui sera ensuite utilisée par le programme locate pour retrouver ces informations. Lancer cette commande quotidiennement est une bonne pratique pour retrouver des informations à jour au moment opportun.

### xargs

La commande xargs applique une commande à une liste de fichiers. S'il faut appliquer la même commande à de multiples fichiers, une liste peut être créée pour contenir la liste de tous ces fichiers (un par ligne) et xargs exécuté une commande sur chacun d'eux.

## Dépendances d'installation de Findutils

Dernière vérification effectuée auprès de la version 4.1.

Bash: sh Binutils: ar, as, ld, ranlib Diffutils: cmp Fileutils: chmod, cp, install, mv, rm
Grep: egrep, grep Gcc: cc1, collect2, cpp0, gcc Make: make Patch: patch Sed: sed
Sh−utils: basename, date, echo, hostname Textutils: cat, tr

# Installer Gawk–3.1.1

```
Estimation du temps de construction :           0.39 SBU
Estimation de l'espace disque requis :  15 Mo
```

## Installation de Gawk

Ce package nécessite que son correctif lui soit appliqué avant que vous puissiez l'installer. Assurez–vous qu'il a été déballé avant de lancer les commandes d'installation.

Avertissement : ne lancez PAS **`make uninstall`** pour ce package si vous utilisez le correctif pour modifier la définition du répertoire `libexec`. La règle *uninstall* dans le fichier `Makefile` lance une commande comme **`rm -rf <libexecdir>/*`**. Comme nous changeons le répertoire `libexec` en `/usr/bin`, cela exécuterait un **`rm -rf /usr/bin/*`**.

Installez Gawk en lançant les commandes suivantes :

```
patch -Np1 -i ../gawk-3.1.1-2.patch &&
./configure --prefix=/usr --libexecdir=/usr/bin &&
make &&
make install
```

## Explication de la commande

**`patch -Np1 -i ../gawk-3.1.1-2.patch:`** Ce correctif modifie le code déterminant l'emplacement du répertoire `libexec`. Ce correctif nous permettra de changer sa valeur en indiquant *--libexecdir* au script configure.

## Contenu de Gawk

Dernière vérification effectuée auprès de la version 3.1.1.

### Programmes

awk, gawk, gawk–3.1.1, grcat, igawk, pgawk, pgawk–3.1.1, pwcat

### Descriptions

#### awk

awk est un lien symbolique vers gawk.

#### gawk, gawk–3.1.1

gawk est l'implémentation GNU de awk, un langage basé sur le traitement et la reconnaissance de chaînes de caractères.

### grcat

grcat concatène la base de données des groupes, /etc/group.

### igawk

igawk est un script shell qui donne à gawk la possibilité d'inclure des fichiers.

### pgawk, pgawk–3.1.1

pgawk est la version de débuggage de gawk.

### pwcat

pwcat concatène la base de données des utilisateurs, /etc/passwd.

## Dépendances d'installation de Gawk

Dernière vérification effectuée auprès de la version 3.1.0.

(Aucune dépendance encore vérifiée)

# Installer Ncurses–5.2

```
Estimation du temps de construction:           1.88 SBU
Estimation de l'espace disque requis:  22 Mo
```

## Installation de Ncurses

Ce package nécessite que son correctif lui soit appliqué avant de pouvoir l'installer. Soyez sûr qu'il a été déballé avant de lancer les commandes d'installation.

Installez Ncurses en lançant les commandes suivantes:

```
patch -Np1 -i ../ncurses-5.2-2.patch &&
./configure --prefix=/usr --with-shared &&
make &&
make install &&
chmod 755 /usr/lib/*.5.2 &&
mv /usr/lib/libncurses.so.5* /lib &&
ln -s libncurses.a /usr/lib/libcurses.a &&
ln -sf ../../lib/libncurses.so.5 /usr/lib/libncurses.so &&
ln -sf ../../lib/libncurses.so.5 /usr/lib/libcurses.so
```

## Explication de la commande

**`patch -Np1 -i ../ncurses-5.2-patch:`** Ce correctif corrige un problème de compilation avec GCC–3.2 car Ncurses utilise des constructions qui ne sont plus valides pour le nouveau standard C++.

**`--with-shared:`** Ceci valide la construction des fichiers de bibliothèques partagées ncurses.

**`chmod 755 *.5.2:`** Les bibliothèques partagées devraient être exécutables. La routine d'installation de Ncurses ne positionne pas les permissions correctement, alors nous le faisons manuellement.

**`ln -sf libncurses.a libcurses.a:`** Certains programmes essayent de faire l'édition des liens avec –lcurses au lieu de –lncurses. Ce lien symbolique assure que de tels programmes seront liés sans erreurs.

## Contenu de Ncurses

Dernière vérification effectuée auprès de la version 5.2.

### Programmes

captoinfo (lien vers tic), clear, infocmp, infotocap (lien vers tic), reset (lien vers tset), tack, tic, toe, tput et tset.

### Descriptions

#### captoinfo

captoinfo convertit une description termcap en une description au format terminfo.

### clear

Le programme clear efface l'écran si cela est possible. Il utilise les variables d'environnement pour déterminer le type de terminal puis se renseigne dans la base de données terminfo pour trouver les commandes d'effacement de l'écran.

### infocmp

Le programme infocmp peut être utilisé pour comparer deux entrées binaires de terminfo, ré−écrire une description de terminfo avec l'avantage du champ terminfo use=, ou afficher une description terminfo depuis le fichier binaire (term) dans une variété de formats (fonctionnement inverse de tic).

### infotocap

infotocap convertit une description terminfo en une description au format termcap.

### reset

reset met en place les modes cooked et echo, désactive les modes cbreak et raw, active la prise en compte des retours chariot et réinitialise tous les caractères spéciaux non mis en place à leur valeurs par défaut avant d'initialiser le terminal de la même façon que tset.

### tack

tack vérifie les actions de terminfo.

### tic

tic est le compilateur des entrées de description de terminfo. Le programme traduit un fichier terminfo d'un format source en format binaire pour être utilisé par les routines de la bibliothèque ncurses. Les fichiers terminfo contiennent la description des possibilités d'un terminal.

### toe

Le programme toe liste tous les types de terminaux disponibles selon leur nom primaire avec pour chacun leur description.

### tput

Le programme tput utilise la base de données terminfo pour créer les valeurs associées aux capacités d'un terminal et les informations disponibles pour le shell, pour initialiser un terminal, ou retourner le nom complet d'un type du terminal requis.

### tset

Le programme tset initialise les terminaux de façon à ce qu'ils puissent être utilisés, mais cette fonctionnalité n'est plus tellement utilisée. Elle est fourni pour une compatibilité avec BSD4.4.

## Bibliothèques

libcurses.[a,so] (lien vers libncurses.[a,so]), libform.[a,so], libform_g.a, libmenu.[a,so], libmenu_g.a, libncurses++.a, libncurses.[a,so], libncurses_g.a, libpanel.[a,so] et libpanel_g.a

### libcurses, libncurses++, libncurses, libncurses_g

Les bibliothèques libncurses sont la base du système et sont utilisées pour afficher du texte à l'écran. Un exemple où ncurses est utilisé est dans le programme << make menuconfig>> .

### libform, libform_g

libform est utilisé pour implémenter des formes dans ncurses.

### libmenu, libmenu_g

libmenu est utilisé pour implémenter les menus dans ncurses.

### libpanel, libpanel_g

libpanel est utilisé pour implémenter des panneaux dans ncurses.

# Dépendances d'installation de Ncurses

Dernière vérification effectuée auprès de la version 5.2.

Bash: sh Binutils: ar, as, ld, ranlib Diffutils: cmp Fileutils: chmod, cp, install, ln, mkdir, mv, rm
Gcc: c++, cc1, cc1plus, collect2, cpp0, gcc Glibc: ldconfig Grep: egrep, fgrep, grep Make: make
Gawk: gawk Sed: sed Sh−utils: basename, date, echo, expr, hostname, uname
Textutils: cat, sort, tr, wc

# Installer Vim–6.1

```
Estimation du temps de construction:           0.81 SBU
Estimation de l'espace disque requis:  24 Mo
```

## Installation de Vim

Ce package nécessite que son correctif lui soit appliqué avant de pouvoir l'installer. Soyez sûr qu'il a été déballé avant de lancer les commandes d'installation.

Si vous ne souhaitez pas installer vim, des instructions de constructions pour d'autres éditeurs sont disponibles sur http://beyond.linuxfromscratch.org/view/cvs/postlfs/editors.html. Actuellement, il existe des instructions pour Emacs, nano et joe.

Installez Vim en lançant les commandes suivantes:

```
patch -Np1 -i ../vim-6.1.patch &&
./configure --prefix=/usr &&
make CPPFLAGS=-DSYS_VIMRC_FILE=\\\"/etc/vimrc\\\" &&
make install &&
ln -s vim /usr/bin/vi
```

Si vous pensez installer le système X Window sur votre système LFS, vous voudrez peut–être re–compiler Vim après avoir installé X. Vim est livré avec une jolie version GUI de l'éditeur qui requiert que X et quelques autres bibliothèques soient installées. Pour plus d'informations, consultez la documentation de Vim.

## Explication des commandes

**`patch -Np1 -i ../vim-6.1.patch`:** Ce correctif corrige un problème de compilation avec GCC–3.2.

**`make CPPFLAGS=-DSYS_VIMRC_FILE=\\\"/etc/vimrc\\\"`:** Positionner cette variable imposera à vim d'utiliser le fichier `/etc/vimrc` pour les paramètres globaux de vim. Normalement, ce fichier est situé dans `/usr/share/vim`, mais `/etc` est un endroit plus logique pour ce genre de fichiers.

## Contenu de Vim

Dernière vérification effectuée auprès de la version 6.1.

### Programmes

efm_filter.pl, efm_perl.pl, ex (lien vers vim), less.sh, mve.awk, pltags.pl, ref, rview (lien vers vim), rvim (lien vers vim), shtags.pl, tcltags, vi (lien vers vim), view (lien vers vim), vim, vim132, vim2html.pl, vimdiff (lien vers vim), vimm, vimspell.sh, vimtutor and xxd

### Descriptions

#### efm_filter.pl

efm_filter.pl est un filtre qui lit à partir de l'entrée standard, copie sur la sortie standard et crée un fichier d'erreurs qui peut être lu par vim.

### efm_perl.pl

efm_perl.pl reformatte les messages d'erreur de l'intepréteur Perl pour l'utiliser avec le mode quickfix de vim.

### ex

ex démarre vim dans le mode Ex.

### less.sh

less.sh est un script qui lance vim avec less.vim.

### mve.awk

mve.awk traite les erreurs de vim.

### pltags.pl

pltags.pl crée un fichier de tags pour du code Perl, en utilisation avec vim.

### ref

ref vérifie l'exactitude des arguments.

### rview

rview est une version restreinte de view. Aucune commande shell ne peut être lancé et Vim ne peut être suspendu.

### rvim

rvim est une version restreinte de vim. Aucune commande shell ne peut être lancé et Vim ne peut être suspendu.

### shtags.pl

shtags.pl génère un ficheir tag pour les scripts perl.

### tcltags

tcltags génère un fichier tag pour le code TCL.

### vi

vi démarre vim en mode de compatibilité avec vi.

### view

view démarre vim en mode lecture seule.

### vim

vim démarre vim de la façon normal, par défaut.

### vim132

vim132 lance vim pour un terminal avec 132 colonnes.

### vim2html.pl

vim2html.pl convertit la documentation vim en HTML.

### vimdiff

vimdiff édite deux ou trois versions d'un fichier avec Vim et affiche les différences.

### vimm

vimm active le 'DEC locator input model' sur un terminal distant.

### vimspell.sh

vimspell.sh est un script qui vérifie un fichier et génère les actions nécessaires pour l'utilisation du surlignage de syntaxe dans vim.

### vimtutor

vimtutor démarre le tutoriel de Vim.

### xxd

xxd crée un dump hexadécimal ou fait l'inverse.

## Dépendances d'installation de Vim

Dernière vérification effectuée auprès de la version 6.0.

Bash: sh Binutils: as, ld, strip Diffutils: cmp, diff Fileutils: chmod, cp, ln, mkdir, mv, rm, touch
Find: find Gcc: cc1, collect2, cpp0, gcc Grep: egrep, grep Make: make Net−tools: hostname Sed: sed
Sh−utils: echo, expr, uname, whoami Textutils: cat, tr, wc

# Installation de Bison−1.35

```
Estimation du temps de construction :           0.27 SBU
Estimation de l'espace disque requis :  6 Mo
```

## Installation de Bison

Installez Bison en lançant les commandes suivantes :

```
./configure --prefix=/usr &&
make &&
make install
```

Certains programmes n'utilisent pas bison et essaient de localiser le programme yacc (bison est une −meilleure− alternative à yacc). Donc pour satisfaire ces quelques programmes, nous allons créer un script bash nommé yacc faisant appel à bison qui va émuler à son tour les conventions de nommage des fichiers de sortie de yacc.

Créez un nouveau fichier `/usr/bin/yacc` en exécutant ce qui suit :

```
cat > /usr/bin/yacc << "EOF"
#!/bin/sh
# Begin /usr/bin/yacc

exec /usr/bin/bison −y "$@"

# End /usr/bin/yacc
EOF
chmod 755 /usr/bin/yacc
```

## Contenu de bison

Dernière vérification effectuée auprès de la version 1.35.

### Programmes

bison et yacc

### Descriptions

#### bison

bison est un générateur d'analyse syntaxique, en remplacement de YACC. YACC signifie Encore un Autre Compilateur de Compilateurs (Yet Another Compiler Compiler). Qu'est−ce que bison alors ? Ce programme génère un autre programme en analysant la structure d'un fichier texte. Au lieu d'écrire le programme, l'utilisateur spécifie comment les choses doivent être reliées et, avec ces règles, un programme est construit pour analyser le fichier texte.

Il y a de nombreux exemples où une structuration est nécessaire, l'un d'eux est une calculatrice.

Soit la chaine :

1 + 2 * 3

Un humain peut facilement arriver au résultat de 7. Pourquoi ? A cause de la structure. Notre cerveau sait comment interpréter la chaîne. L'ordinateur ne le sait pas, et bison est un outil pour l'aider à comprendre cela, en présentant la chaîne sous la forme suivante au compilateur :

```
    +          / \          *   1          / \
2   3
```

En commençant en bas de l'arbre, et en arrivant aux nombres 2 et 3, qui sont liés par un symbole de multiplication, l'ordinateur multiplie 2 par 3. Le résultat de cette multiplication est mémorisé et la chose suivante que l'ordinateur voit est le résultat de 2*3 et le nombre 1 qui sont reliés par le symbole d'addition. L'ajout de 1 au résultat précédent donne 7. En calcul, les opérations les plus complexes peuvent être exprimées sous ce format d'arbre, et l'ordinateur commence simplement en bas, et travaille en cheminant vers le haut, pour arriver à la réponse correcte. Bien entendu, bison n'est pas utilisé uniquement pour les calculatrices.

### yacc

Nous créons un script nommé yacc qui appelle bison avec l'option −y. Ceci est fait pour des raisons de compatibilité avec des programmes qui utilisent yacc au lieu de bison.

## Dépendances d'installation de Bison

Dernière vérification effectuée auprès de la version 1.31.

Bash: sh Binutils: ar, as, ld, ranlib Diffutils: cmp
Fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm, rmdir Gcc: cc, cc1, collect2, cpp0, gcc
Grep: egrep, fgrep, grep Make: make Sed: sed
Sh−utils: basename, dirname, echo, expr, hostname, sleep, uname Texinfo: install−info
Textutils: cat, head, tr, uniq

# Installer Less–374

```
Estimation du temps de construction:           0.13 SBU
Estimation de l'espace disque requis:  2 Mo
```

# Installation de Less

Installez Less en exécutant les commandes suivantes:

```
./configure --prefix=/usr --bindir=/bin --sysconfdir=/etc &&
make &&
make install
```

# Contenu de Less

Dernière vérification effectuée auprès de la version 374.

## Programmes

less, lessecho et lesskey

## Description

### less

Le programme less est un paginateur de fichier (ou un afficheur de texte). Il affiche le contenu d'un fichier et permet le défilement du texte. Less est une amélioration du paginateur habituel appelé << more>> . Less permet également le défilement arrière dans les fichiers et n'a pas besoin de lire tout le fichier quand il démarre, ce qui le rend plus rapide lors de la lecture de gros fichiers.

### lessecho

lessecho est utile pour étendre les méta caractères, comme * et ?, dans les noms de fichiers sur les systèmes Unix.

### lesskey

lesskey est utilisé pour spécifier les associations de touches à less.

# Dépendances d'installation de Less

Dernière vérification effectuée auprès de la version 358.

Bash: sh Binutils: as, ld Diffutils: cmp Fileutils: chmod, install, mv, rm, touch Grep: egrep, grep
Gcc: cc1, collect2, cpp0, gcc Make: make Sed: sed Sh–utils: expr, hostname, uname Textutils: cat, tr

# Installer Groff–1.18

```
Estimation du temps de construction:           1.08 SBU
Estimation de l'espace disque requis:  18 Mo
```

## Installation de Groff

Installez Groff en exécutant les commandes suivantes:

```
./configure --prefix=/usr &&
make PROCESSEDEXAMPLEFILES="" &&
make PROCESSEDEXAMPLEFILES="" install &&
ln -s soelim /usr/bin/zsoelim &&
ln -s eqn /usr/bin/geqn &&
ln -s tbl /usr/bin/gtbl
```

## Explications des commandes

**`make PROCESSEDEXAMPLEFILES="":`** Groff a quelques dépendances supplémentaires que nous n'installons pas avec LFS. Cette option désactive la nécessité de ces outils.

**`ln -s ...`**: Ces liens symboliques sont nécessaires pour que un programme tel que **`xman`** et d'autres traitant les documents groff/man puissent fonctionner correctement.

## Contenu de Groff

Dernière vérification effectuée auprès de la version 1.17.2.

### Programmes

addftinfo, afmtodit, eqn, geqn (lien vers eqn), grn, grodvi, groff, grog, grolbp, grolj4, grops, grotty, gtbl (lien vers tbl), hpftodit, indxbib, lkbib, lookbib, mmroff, neqn, nroff, pfbtops, pic, post–grohtml, pre–grohtml, refer, soelim, tbl, tfmtodit, troff et zsoelim (lien vers soelim)

### Descriptions

#### addftinfo

addftinfo lit un fichier de polices troff et ajoute des informations de description des polices utilisées par le système groff.

#### afmtodit

afmtodit crée un fichier de polices utilisables par groff et grops.

#### eqn

eqn traduit les descriptions d'équations contenues dans des fichiers sources en commandes compréhensibles par troff.

### geqn

geqn est l'implémentation GNU de eqn.

### grn

grn est un préprocesseur groff pour les fichiers gremlin.

### grodvi

grodvi est un driver de groff qui produit un format TeX dvi.

### groff

groff est l'interface du système de formatage de documents groff. Normalement, il exécute le programme troff et le post−processeur approprié en fonction du périphérique sélectionné.

### grog

grog lit les fichiers et détermine quelles options de groff −e, −man, −me, −mm, −ms, −p, −s et −t sont nécessaires pour l'impression des fichiers, et affiche la commande groff avec ces options sur la sortie standard.

### grolbp

grolbp est un driver pour groff qui permet l'impression sur des imprimantes Canon CAPSL (imprimantes laser série LBP−4 et LBP−8).

### grolj4

grolj4 est un driver pour groff qui permet l'impression au format PCL5 pour une imprimante HP Laserjet 4.

### grops

grops convertit la sortie troff GNU en Postscript.

### grotty

grotty convertit la sortie troff GNU en un format compatible pour les périphériques de type 'typewriter'.

### gtbl

gtbl est l'implémentation GNU de tbl.

### hpftodit

hpftodit crée un fichier de police utilisable avec groff −Tlj4 depuis un fichier de type 'HP tagged police metric'.

### indxbib

indxbib crée un index inversé pour la base de données bibliographique d'un fichier donné de façon à pouvoir utiliser refer, lookbib, et lkbib.

### lkbib

lkbib cherche les références dans les bases de données bibliographiques qui contiennent certaines clefs et affiche toutes les références trouvées sur la sortie standard.

### lookbib

lookbib affiche une invite sur le flux d'erreur standard (à moins que l'entrée standard ne soit pas un terminal), lit depuis l'entrée standard une suite de mots clefs, recherche dans les bases de données bibliographiques du fichier spécifié les références contenant ces mots clefs, les affiche sur la sortie standard, et répète ce processus jusqu'à ce qu'il n'y ait plus d'entrée.

### mmroff

mmroff est un préprocesseur basic pour groff.

### neqn

Le script neqn formate les équations pour une sortie ASCII.

### nroff

Le script nroff émule la commande nroff en utilisant groff.

### pfbtops

pfbtops convertit une police Postscript au format .pfb en ASCII.

### pic

pic compile des descriptions d'images contenues dans des fichiers sources en commandes compréhensibles par TeX ou troff.

### pre−grohtml et post−grohtml

pre− et post−grohtml traduisent la sortie de GNU troff au format html.

### refer

refer copie le contenu d'un fichier sur la sortie standard, sauf les parties situées entre .[ et .] interprétées comme des citations, et celles situées entre .R1 et .R2 qui sont interprétées comme des commandes de traitement de ces citations.

### soelim

soelim examine des fichiers et remplace les lignes du style *.so fichier* par le contenu de *fichier*.

### tbl

tbl traduit les descriptions de tables contenues dans des fichiers sources en commandes compréhensibles par troff.

### tfmtodit

tfmtodit crée un fichier police utilisable par **`groff -Tdvi`**.

### troff

troff est quasi identique à la commande Unix troff. Normalement il devrait être appelé par la commande groff, qui exécutera les préprocesseurs et post− processeurs dans l'ordre approprié et avec les options appropriées.

### zsoelim

zsoelim est l'implémentation GNU de soelim.

## Dépendances d'installation de Groff

Dernière vérification effectuée auprès de la version 1.17.2.

Bash: sh Binutils: ar, as, ld, ranlib Bison: bison Diffutils: cmp
Fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm, touch
Gcc: cc1, cc1plus, collect2, cpp0, g++, gcc Grep: egrep, grep Make: make Gawk: awk Sed: sed
Sh−utils: basename, date, echo, expr, hostname, uname Textutils: cat, tr

# Installer Textutils–2.1

```
Estimation du temps de construction:           0.83 SBU
Estimation de l'espace disque requis:  17 Mo
```

# Installation de Textutils

Installez Textutils en lançant les commandes suivantes:

```
./configure --prefix=/usr &&
make &&
make install &&
mv /usr/bin/{cat,head} /bin
```

# Contenu de Textutils

Dernière vérification effectuée auprès de la version 2.0.

## Programmes

cat, cksum, comm, csplit, cut, expand, fmt, fold, head, join, md5sum, nl, od, paste, pr, ptx, sort, split, sum, tac, tail, tr, tsort, unexpand, uniq et wc

## Descriptions

### cat

cat concatène un ou plusieurs fichiers ou entrée standard vers la sortie standard.

### cksum

cksum affiche la somme de contrôle cyclique et le compte d'octets des fichiers spécifiés.

### comm

comm compare deux fichiers triés ligne par ligne.

### csplit

csplit découpe un fichier en plusieurs sections nommées xx01, xx02, ..., et affiche le compte d'octets de chaque section sur la sortie standard.

### cut

cut sélectionne certaines colonnes d'un ou plusieurs fichiers et les affiche sur la sortie standard.

### expand

expand convertit les tabulations d'un fichier en espaces, il affiche le résultat sur la sortie standard.

### fmt

fmt reformate chaque paragraphe d'un ou plusieurs fichiers et affiche le résultat sur la sortie standard.

### fold

fold enregistre les lignes entrées dans des fichiers spécifiés (sur l'entrée standard par défaut), et écrit sur la sortie standard.

### head

head affiche les xx premières lignes (10 par défaut) des fichiers spécifiés sur la sortie standard.

### join

join fusionne les lignes de deux fichiers ayant un champ commun.

### md5sum

md5sum affiche ou vérifie la somme de contrôle MD5.

### nl

nl affiche les fichiers spécifiés sur la sortie standard en ajoutant le numéro de ligne.

### od

od affiche le contenu d'un fichier en octal par défaut sur la sortie standard.

### paste

paste affiche des lignes regroupant les lignes correspondantes de chaque fichier, séparées par des tabulations, sur la sortie standard.

### pr

pr met en page des fichiers pour l'impression.

### ptx

ptx produit un index permuté du contenu d'un fichier.

### sort

sort écrit la concaténation triée de fichiers sur la sortie standard.

### split

split découpe un fichier en sections de taille préfixée nommées PREFIXaa, PREFIXab, ...

### sum

sum affiche la somme de contrôle et le compte de blocs pour chaque fichier spécifié.

### tac

tac écrit les fichiers spécifiés sur la sortie standard, dans l'ordre inverse.

### tail

tail affiche les xx dernières lignes (10 par défaut) des fichiers spécifiés sur la sortie standard.

### tr

tr transpose ou élimine des caractères de l'entrée standard et affiche le résultat sur la sortie standard.

### tsort

tsort affiche une liste ordonnée cohérente à partir de fichiers partiellement ordonnés.

### unexpand

unexpand convertit les espaces en tabulations pour les fichiers spécifiés, et affiche le résultat sur la sortie standard.

### uniq

uniq efface les lignes dupliquées d'un fichier trié.

### wc

wc affiche le nombre de lignes, de mots et d'octets pour tous les fichiers spécifiés, et un total du nombre de lignes si plusieurs fichiers ont été spécifiés.

## Dépendances d'installation de Textutils

Dernière vérification effectuée auprès de la version 2.0.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh Binutils: ar, as, ld, ranlib
Diffutils: cmp Fileutils: chmod, install, ls, mv, rm Gettext: msgfmt, xgettext
Gcc: cc, cc1, collect2, cpp0, gcc Glibc: getconf Grep: egrep, fgrep, grep M4: m4 Make: make
Gawk: gawk Net−tools: hostname Perl: perl Sed: sed Sh−utils: basename, echo, expr, sleep, uname
Tar: tar Texinfo: install−info, makeinfo Textutils: cat, tr

# Installer Sed–3.02

```
Estimation du temps de construction:           0.09 SBU
Estimation de l'espace disque requis:  2 Mo
```

# Installation de Sed

Installez Sed en lançant les commandes suivantes:

```
./configure --prefix=/usr --bindir=/bin &&
make &&
make install
```

# Contenu de Sed

Dernière vérification effectuée auprès de la version 3.02.

## Programmes

sed

## Descriptions

### sed

sed est un éditeur de flux. Un éditeur de flux est utilisé pour exécuter des transformations de texte basiques sur un flux d'entrée (un fichier ou l'entrée reçue d'un tuyau).

# Dépendances d'installation de Sed

Dernière vérification effectuée auprès de la version 3.02.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh Binutils: ar, as, ld, ranlib
Diffutils: cmp Fileutils: chmod, install, ls, mv, rm Gcc: cc1, collect2, cpp0, gcc Glibc: getconf
Grep: egrep, fgrep, grep M4: m4 Make: make Gawk: gawk Sed: sed
Sh–utils: echo, expr, hostname, sleep Texinfo: install–info, makeinfo Textutils: cat, tr

# Installer Flex–2.5.4a

```
Estimation du temps de construction:           0.05 SBU
Estimation de l'espace disque requis:  3 Mo
```

## Installation de Flex

Installez Flex en exécutant les commandes suivantes :

```
./configure --prefix=/usr &&
make &&
make install
```

Quelques programmes ne connaissent pas flex et essaient de trouver le programme lex(flex est une –meilleure– alternative à lex). Donc pour satisfaire ces quelques programmes, nous allons créer un script bash nommé lex, qui appelle flex et qui imite lex.

Créez un nouveau fichier `/usr/bin/lex` en exécutant ceci :

```
cat > /usr/bin/lex << "EOF"
#!/bin/sh
# Debut /usr/bin/lex

exec /usr/bin/flex -l "$@"

# Fin /usr/bin/lex
EOF
chmod 755 /usr/bin/lex
```

## Contenu de Flex

Dernière vérification effectuée auprès de la version 2.5.4a.

### Programmes

flex, flex++ (lien vers flex) et lex

### Descriptions

#### flex

flex est un utilitaire destiné à générer des programmes pour la reconnaissance de motifs dans des textes. La reconnaissance de motifs est très utile dans de nombreuses applications. Un utilisateur définit des règles à rechercher et flex créera un programme destiné à reconnaître ces motifs. Utiliser flex est un avantage car il est beaucoup plus facile de définir des règles de recherche que d'écrire un programme pour repérer le texte voulu.

#### flex++

flex++ invoque une version de flex utilisé exclusivement par les scanners C++.

#### lex

Nous créons un script bash, nommé lex, appellant flex avec l'option –l. Ceci est fait pour des raisons de compatibilité avec les programmes utilisant lex au lieu de flex.

### Bibliothèques

libfl.a

### Descriptions

#### libfl

libfl est la bibliothèque flex.

## Dépendances d'installation de Flex

Dernière vérification effectuée auprès de la version 2.5.4a.

Bash: sh Binutils: ar, as, ld, ranlib Bison: bison Diffutils: cmp
Fileutils: chmod, cp, install, ln, mv, rm, touch Gcc: cc1, collect2, cpp0, gcc Grep: egrep, grep
Make: make Sed: sed Sh–utils: echo, hostname Textutils: cat, tr

# Installation de Binutils–2.13

```
Estimation du temps de construction :           2.48 SBU
Estimation de l'espace disque requis :  94 Mo
```

## Installation de Binutils

Ce package est connu pour se comporter bizarrement si vous avez changé les options d'optimisation par défaut (y compris les options –march et –mcpu). Binutils se comporte mieux sans. Donc, si vous avez défini des variables d'environnement qui écrasent les optimisations par défaut, comme CFLAGS et CXXFLAGS, nous vous recommendons de les annuler ou de les modifier lors de la construction de Binutils. Vous avez été prévenu.

Installez Binutils en lançant les commandes suivantes:

```
mkdir ../binutils-build &&
cd ../binutils-build &&
../binutils-2.13/configure --prefix=/usr --enable-shared &&
make tooldir=/usr &&
make tooldir=/usr install &&
make tooldir=/usr install-info &&
cp ../binutils-2.13/include/libiberty.h /usr/include
```

## Explication des commandes

**`tooldir=/usr`:** Normalement, la variable tooldir (répertoire où les exécutables de binutils sont stockés) est définie à $(exec_prefix)/$(target_alias) ce qui correspond, par exemple, à /usr/i686–pc–linux–gnu. Comme nous ne construisons que notre système, nous n'avons pas besoin de ce répertoire cible dans /usr. Cette configuration est intéressante si le système était utilisé pour de la cross–compilation (par exemple, pour compiler un package sur une machine Intel en générant du code qui sera exécuté sur des machines Apple PowerPC).

**`make tooldir=/usr install-info`:** Ceci installera les pages d'informations de binutil.

**`cp ../binutils-2.13/include/libiberty.h /usr/include`:** Le fichier d'entête libiberty.h est nécessaire pour compiler certains logiciels.

## Contenu de Binutils

Dernière vérification effectuée auprès de la version 2.12.1.

### Programmes

addr2line, ar, as, gasp, gprof, ld, nm, objcopy, objdump, ranlib, readelf, size, strings et strip

### Descriptions

#### addr2line

addr2line traduit les adresses d'un programme en nom de fichiers et numéros de ligne. A partir d'une adresse et d'un exécutable, il utilise les informations de déboguage présentes dans l'exécutable pour trouver quel nom

de fichier et quel numéro de ligne sont associés à une adresse donnée.

### ar

Le programme ar crée, modifie et extrait des données depuis des archives. Une archive est un fichier contenant un ensemble d'autres fichiers dans une structure qui permet de retrouver les fichiers individuels d'origine (appelés membres de l'archive).

### as

as a d'abord été prévu pour assembler la sortie du compilateur GNU C, gcc, pour qu'elle puisse être utilisée par l'éditeur de liens ld.

### gasp

gasp est le préprocesseur de macros assembleur.

### gprof

gprof affiche les données de profil sous forme de graphe d'appel.

### ld

ld combine plusieurs fichiers objets et archives, modifie l'emplacement de leurs données et lie les références des symboles. Souvent, la dernière étape dans la construction d'un nouveau programme compilé à exécuter est un appel à ld.

### nm

nm liste les symboles des fichiers objet.

### objcopy

L'utilitaire objcopy copie le contenu d'un fichier objet dans un autre. objcopy utilise la bibliothèque GNU BFD pour lire et écrire les fichiers objet. Il peut écrire le fichier objet destination dans un format différent de celui du fichier objet source.

### objdump

objdump affiche des informations sur un ou plusieurs fichiers objet. Les options déterminent quelles informations spécifiques sont à afficher. Ces informations sont essentiellement utiles pour les programmeurs qui travaillent sur des outils de compilation, à l'inverse des programmeurs qui veulent juste compiler leur programme pour qu'il fonctionne.

### ranlib

ranlib génère un index du contenu d'une archive, et le stocke dans l'archive. L'index liste chaque symbole défini par un membre d'une archive, qui est un fichier objet re−localisable.

#### readelf

readelf affiche des informations sur les binaires de type elf.

#### size

size liste les tailles des sections, et la taille totale, pour chaque fichier objet de sa liste d'arguments. Par défaut, une ligne est générée en sortie pour chaque fichier objet ou chaque module dans une archive.

#### strings

Pour chaque fichier donné, strings affiche les séquences de caractères imprimables qui ont au moins une longueur de quatre caractères (ou le nombre spécifié en option du programme) et qui sont suivies d'un caractère non imprimable. Par défaut, il affiche seulement les chaînes des parties initialisées et chargées des fichiers objet. Pour les autres types de fichier, il affiche les chaînes de tout le fichier.

strings est principalement utile pour déterminer le contenu des fichiers binaires.

#### strip

strip débarrasse les fichiers objet de tous leurs symboles, ou de certains en particulier. La liste des fichiers objet peut inclure des archives. Au moins un fichier objet doit être fourni. strip modifie les fichiers fournis en argument, plutôt que d'écrire les copies modifiées sous d'autres noms.

### Bibliothèques

libbfd.[a,so] et libopcodes.[a,so]

### Descriptions

#### libbfd

libbfd est une bibliothèqes de description des fichiers binaires.

#### libopcodes

libopcodes est une bibliothèque native pour traiter les opcodes et est utilisé dans le but de construire des utilitaires comme ojbdump. Les opcodes correspondent à une version texte facilement lisible d'instructions processeurs.

## Dépendances d'installation de Binutils

Dernière vérification effectuée auprès de la version 2.11.2.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh
Binutils: ar, as, ld, nm, ranlib, strip Diffutils: cmp
Fileutils: chmod, cp, ln, ls, mkdir, mv, rm, rmdir, touch Flex: flex
Gcc: cc, cc1, collect2, cpp0, gcc Glibc: ldconfig Grep: egrep, fgrep, grep M4: m4 Make: make
Gawk: gawk Sed: sed Sh−utils: basename, echo, expr, hostname, sleep, true, uname
Texinfo: install−info, makeinfo Textutils: cat, sort, tr, uniq

# Installer Fileutils–4.1

```
Estimation du temps de construction:           0.68 SBU
Estimation de l'espace disque requis:  17 Mo
```

# Installation de Fileutils

Installez Fileutils en lançant les commandes suivantes :

```
./configure --prefix=/usr --bindir=/bin &&
make &&
make install &&
ln -s ../../bin/install /usr/bin
```

# Contenu de Fileutils

Dernière vérification effectuée auprès de la version 4.1.

## Programmes

chgrp, chmod, chown, cp, dd, df, dir, dircolors, du, install, ln, ls, mkdir, mkfifo, mknod, mv, rm, rmdir, shred, sync, touch et vdir

## Descriptions

### chgrp

chgrp remplace le groupe des propriétaires des fichiers spécifiés par celui dont le nom ou le numéro est passé en paramètre.

### chmod

chmod change les permissions des fichiers spécifiés en fonction du mode, ce mode pouvant être une représentation symbolique des changements à effectuer, ou une représentation octale d'un nombre représentant le champ de bits des nouvelles permissions.

### chown

chown change le propriétaire et/ou le groupe de propriétaires des fichiers spécifiés.

### cp

cp copie des fichiers d'un répertoire à un autre.

### dd

dd copie un fichier (de l'entrée standard vers la sortie standard, par défaut) en fonction d'une taille de bloc et des options de conversion du fichier données par l'utilisateur.

### df

df affiche l'espace disque disponible du système de fichiers contenant les fichiers passés en argument. Sans argument, df indiquera les quantités correspondant à tous les systèmes de fichiers montés.

### dir, ls et vdir

dir et vdir sont des versions de ls avec des formats d'affichage différents. Ces programmes listent tous les fichiers ou répertoires donnés en argument. Le contenu des répertoires est trié alphabétiquement. Pour ls, les fichiers sont affichés en colonnes par défaut et triés verticalement si la sortie standard est un terminal; ils sont affichés ligne par ligne dans le cas contraire. Pour dir, les fichiers sont par défaut affichés en colonnes et triés verticalement. Pour vdir, l'affichage par défaut est le format long.

### dircolors

dircolors affiche une liste de commandes permettant de définir la variable d'environnement LS_COLOR. La variable LS_COLOR est utilisée pour changer les couleurs d'affichage par défaut de la commande ls et d'autres utilitaires.

### du

du affiche l'espace disque utilisé par chaque fichier et répertoire listés en argument sur la ligne de commande et pour chacun de leurs sous−répertoires.

### install

install copie les fichiers puis positionne leurs permissions et, si possible, leur propriétaire et leur groupe.

### ln

ln crée des liens matériels ou symboliques entre fichiers.

### mkdir

mkdir crée un répertoire correspondant à chacun des noms mentionnés en argument.

### mkfifo

mkfifo crée un fichier FIFO avec les noms passés en argument.

### mknod

mknod crée un fichier FIFO, fichier spécial en mode caractère ou fichier spécial en mode bloc, avec le nom passé en argument.

### mv

mv déplace des fichiers d'un répertoire à un autre ou les renomme, selon les arguments qui lui sont passés.

### rm

rm efface des fichiers ou des répertoires.

### rmdir

rmdir efface des répertoires, s'ils sont vides.

### shred

shred efface un fichier de façon sûre, en écrasant son contenu afin qu'il ne puisse pas être récupéré.

### sync

sync force l'écriture des blocs modifiés et met à jour le super bloc.

### touch

touch remplace le temps d'accès et de modification des fichiers passés en argument par l'heure et la date courantes. S'ils n'existent pas, des fichiers vides sont créés.

## Dépendances d'installation de Fileutils

Dernière vérification effectuée auprès de la version 4.1.

Bash: sh Binutils: ar, as, ld, ranlib Diffutils: cmp
Fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm, rmdir Gettext: msgfmt, xgettext
Gcc: cc, cc1, collect2, cpp0, gcc Grep: egrep, fgrep, grep Make: make Perl: perl Sed: sed
Sh−utils: basename, echo, expr, hostname, sleep, uname Texinfo: install−info Textutils: cat, tr

# Installer Sh–utils–2.0

```
Estimation du temps de construction:           0.42 SBU
Estimation de l'espace disque requis:  12 Mo
```

## Installation de Sh–utils

Ce package requiert que son correctif pour hostname lui soit appliqué avant qu'il ne soit installé. Assurez–vous qu'il soit déballé avant de lancer les commandes d'installation.

Installez Sh–utils en lançant les commandes suivantes:

```
patch -Np1 -i ../sh-utils-2.0-hostname.patch &&
./configure --prefix=/usr &&
make &&
make install &&
mv /usr/bin/{basename,date,echo,false,pwd} /bin &&
mv /usr/bin/{sleep,stty,su,test,true,uname} /bin &&
mv /usr/bin/chroot /usr/sbin
```

## Notes sur la conformité FHS

Il y a une commande installée dans ce package qui est nommée test. Elle est souvent utilisée dans les scripts shell pour évaluer des conditions, mais elle est encore plus souvent rencontrée sous la forme **[ condition ]**. Ces crochets sont construits dans l'interpréteur bash, mais le FHS définit qu'il devrait y avoir un binaire `[`. Créer le en lançant :

```
ln -s test /bin/[
```

## Explications des commandes

**`patch -Np1 -i ../sh-utils-2.0-hostname.patch`:** Ce correctif empêche la construction du programme hostname que nous installerons plus tard avec le package net–tools. Le programme hostname du package net–tools est une bien meilleure version (et dans quelques cas requis car il supporte des options nécessaires à certaines programmes comme XFree86).

## Contenu de Sh–utils

Dernière vérification effectuée auprès de la version 2.0.

### Programmes

basename, chroot, date, dirname, echo, env, expr, factor, false, groups, hostid, id, logname, nice, nohup, pathchk, pinky, printenv, printf, pwd, seq, sleep, stty, su, tee, test, true, tty, uname, uptime, users, who, whoami et yes

### Descriptions

### basename

basename enlève le répertoire et les suffixes d'un nom de fichier.

### chroot

chroot exécute une commande ou un shell interactif depuis un répertoire racine particulier.

### date

date affiche le temps courant dans le format spécifié, ou met à jour la date système.

### dirname

dirname enlève les suffixes ne faisant pas partie du répertoire d'un nom de fichier.

### echo

echo affiche une ligne de texte.

### env

env exécute un programme dans un environnement modifié.

### expr

expr évalue des expressions.

### factor

factor affiche les facteurs premiers des nombres entiers spécifiés.

### false

false sort toujours avec un code de statut indiquant une erreur.

### groups

groups affiche les groupes auquels appartient l'utilisateur.

### hostid

hostid affiche l'identifiant numérique (en hexadécimal) pour l'hôte courant.

### id

id affiche les identifiants effectifs d'utilisateur et de groupe de l'utilisateur courant ou d'un utilisateur donné.

### logname

logname affiche le nom de login de l'utilisateur.

### nice

nice exécute un programme avec une priorité de scheduling modifiée.

### nohup

nohup lance une commande qui ne tiendra pas compte du signal hangups, avec une sortie vers un fichier de trace.

### pathchk

pathchk vérifie si les noms de fichiers sont valides et portables.

### pinky

pinky est une commande finger allégée qui permet de retrouver des informations sur certains utilisateurs.

### printenv

printenv affiche tout ou partie des variables d'environnement.

### printf

printf formate et affiche des données (identique à la fonction printf en C).

### pwd

pwd affiche le nom du répertoire de travail courant.

### seq

seq affiche des nombres dans un certain ordre avec un incrément à spécifier.

### sleep

sleep stoppe une tâche pendant un certain laps de temps.

### stty

stty change et affiche les caractéristiques d'une ligne de terminal.

### su

su exécute un shell en substituant l'identifiant d'utilisateur et de groupe.

### tee

tee lit l'entrée standard et écrit sur la sortie standard et un fichier.

### test

test vérifie les types de fichiers et compare les valeurs.

### true

true sort toujours avec un code de statut indiquant le succès.

### tty

tty affiche le nom de fichier du terminal connecté à l'entrée standard.

### uname

uname affiche des informations système.

### uptime

uptime donne le temps depuis lequel le système fonctionne.

### users

users affiche les noms des utilisateurs actuellement connectés sur la machine hôte.

### who

who affiche qui est connecté.

### whoami

whoami affiche le nom de l'utilisateur associé à l'identifiant utilisateur effectif courant.

### yes

yes affiche 'y' ou une chaîne de caractères donnée répétitivement jusqu'à ce que la tâche soit tuée.

## Dépendances d'installation de Sh−utils

Dernière vérification effectuée auprès de la version 2.0.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh Binutils: ar, as, ld, ranlib
Diffutils: cmp Fileutils: chmod, chown, install, ls, mv, rm Gettext: msgfmt, xgettext
Gcc: cc, cc1, collect2, cpp0, gcc Glibc: getconf Grep: egrep, fgrep, grep M4: m4 Make: make
Gawk: gawk Perl: perl Sed: sed Sh−utils: basename, echo, expr, hostname, sleep, uname Tar: tar
Texinfo: install−info, makeinfo Textutils: cat, tr

# Installer Gettext–0.11.5

```
Estimation du temps de construction:           0.99 SBU
Estimation de l'espace disque requis:  39 Mo
```

# Installation de Gettext

Installez Gettext en exécutant les commandes suivantes:

```
./configure --prefix=/usr &&
make &&
make install
```

# Contenu de Gettext

Dernière vérification effectuée auprès de la version 0.11.2.

## Programmes

config.charset, config.rpath, gettext, gettextize, hostname, msgattrib, msgcat, , msgcmp, msgcomm, msgconv, msgen, msgexec, msgfilter, msgfmt, msggrep, msginit, msgmerge, msgunfmt, msguniq, ngettext, project–id, team–address, trigger, urlget, user–email et xgettext

## Descriptions

### config.charset

Le script config.charset affiche une table, dépendant du système, des alias de caractères.

### config.rpath

Le script config.rpath affiche un ensemble de variables, dépendant du système, décrivant comment initialiser le chemin de recherche des bibliothèques dynamiques dans un exécutable.

### gettext

Le package gettext est utilisé pour l'internationalisation (aussi nommée i18n) et pour la localisation (connue aussi sous le nom de l10n). Les programmes peuvent ainsi être compilés avec le support de langue native (Native Language Support) pour permettre l'affichage dans la langue native de l'utilisateur plutôt qu'en anglais qui est la langue par défaut.

### gettextize

Le programme gettextize copie tous les fichiers au standard gettext dans un répertoire. Ceci est utile pour créer un package de traductions gettext.

### hostname

Le programme hostname affiche le nom réseau de l'hôte, dans plusieurs formats.

### msgattrib

Le programme msgattrib filtre les messages d'un catalogue de traduction d'après leurs attributs, et manipule les attributs.

### msgcat

Le programme msgcat trouve les messages qui sont communs à plusieurs traductions.

### msgcmp

Le programme msgcmp compare deux fichiers de traduction.

### msgcomm

Le programme msgcomm recherche les messages qui apparaissent dans plusieurs fichiers .po. Ceci est utile pour comparer l'état de la traduction.

### msgconv

Le programme msgconv convertit un catalogue de traduction dans un encodage de caractères différent.

### msgen

Le programme msgen crée un catalogue de traductions anglaises.

### msgexec

Le programme msgexec applique une commande à toutes les traductions d'un catalogue de traductions.

### msgfilter

Le programme msgfilter applique un filtre à toutes les traductions d'un catalogue de traductions.

### msgfmt

Le programme msgftm compile les traductions en code machine. Ceci est utile pour créer le fichier final de traduction du programme/package.

### msggrep

Le programme msggrep extrait tous les messages d'un catalogue de traduction qui suive un certain modèle ou appartienne à des fichiers sources donnés.

### msginit

Le programme msginit crée un nouveau fichier PO, initialisant les informations meta avec les valeurs de l'environnement de l'utilisateur.

### msgmerge

Le programme msgmerge combine deux traductions en un fichier. Cela permet de mettre à jour les fichiers de traduction à partir de la source.

### msgunfmt

Le programme msgunfmt décompile les fichiers de traduction en leur version texte. Ceci est utilisé uniquement dans le cas où les version compilées sont disponibles.

### msguniq

Le programme msguniq unifie des traductions dupliquées dans un catalogue de traduction.

### ngettext

Le programme ngettext affiche la traduction dans le langage natif d'un message textuel dont la forme grammaticale dépend d'un nombre.

### project−id

Le script project−id affiche l'identification d'un package ou la version d'un package.

### team−address

Le script team−address envoie l'adresse de l'équipe sur stdout et affiche quelques informations supplémentaires.

### trigger

Le script trigger teste si le package courant est un package GNOME ou KDE.

### urlget

Le programme urlget récupère le contenu d'une URL.

### user−email

Le script user−email affiche l'adresse email de l'utilisateur, avec la confirmation de l'utilisateur.

### xgettext

Le programme xgettext extrait les lignes du message à partir des sources C. Cela permet la création du fichier modèle de traduction.

## Bibliothèques

libgettextlib[a,so], libgettextsrc[a,so]

## Descriptions

### libgettextlib

Aucune description actuellement disponible.

### libgettextsrc

Aucune description actuellement disponible.

# Dépendances d'installation de Gettext

Dernière vérification effectuée auprès de la version 0.10.40.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh
Binutils: ar, as, ld, nm, ranlib, strip Bison: bison Diffutils: cmp
Fileutils: chmod, install, ln, ls, mkdir, mv, rm, rmdir Gcc: cc, cc1, collect2, cpp0, gcc
Grep: egrep, fgrep, grep M4: m4 Make: make Gawk: gawk Sed: sed
Sh−utils: basename, echo, expr, hostname, sleep, uname Texinfo: install−info, makeinfo
Textutils: cat, sort, tr, uniq

# Installer Net–tools–1.60

```
Estimation du temps de construction:           0.16 SBU
Estimation de l'espace disque requis:  5 Mo
```

## Installation de Net–tools

Installez Net–tools en lançant les commandes suivantes:

```
make &&
make update
```

Si vous voulez accepter toutes les réponses par défaut, vous pouvez lancer à la place les commandes suivantes:

```
yes "" | make &&
make update
```

Si vous ne savez pas quoi répondre à toutes les questions posées lors de la phase du **`make`**, alors acceptez les réponses par défaut, ce qui sera bien dans la majorité des cas. Il va vous être posé toute une série de question sur le type de protocoles réseaux que vous souhaitez activer dans votre noyau.

Les réponses par défaut vont activer tous les outils du package qui fonctionnent avec la plupart des protocoles communs comme TCP, PPP et quelques autre encore. Vous avez toujours besoin d'activer ces protocoles dans le noyau. Ce que vous faites ici est tout simplement de dire aux programmes d'être capable d'utiliser ces protocoles, mais c'est au noyau de les rendre disponibles pour le système.

## Explication des commandes

**`make update:`** Réalise la même chose qu'un **`make install`** avec la différence que make update n'effectue pas de sauvegardes des fichiers qu'il remplace. Une des choses que net–tools remplace est la version sh–utils de `/bin/hostname` (la version net–tools est de loin meilleure que celle de sh–utils).

De même, si vous décidez dans le futur de réinstaller ce package, un **`make update`** ne sauvegardera pas les fichiers d'une précédente installation de net–tools.

## Contenu de Net–tools

Dernière vérification effectuée auprès de la version 1.60.

### Programmes

arp, dnsdomainname (lien vers hostname), domainname (lien vers hostname), hostname, ifconfig, nameif, netstat, nisdomainname (lien vers hostname), plipconfig, rarp, route, slattach et ypdomainname (lien vers hostname)

### Descriptions

### arp

arp manipule le cache ARP du noyau, habituellement pour ajouter ou supprimer une entrée, ou affiche le contenu du cache ARP.

### dnsdomainname

dnsdomainname affiche le nom de domaine DNS du système.

### domainname

domainname affiche ou enregistre le nom de domaine NIS/YP du système.

### hostname

hostname est utilisé pour enregistrer ou afficher le nom d'hôte du système.

### ifconfig

ifconfig est la commande principale pour configurer les interfaces réseau.

### nameif

nameif nomme les interfaces réseau, basées sur les adresses MAC.

### netstat

netstat est un utilitaire ayant de nombreux usages; il affiche les connections réseau, les tables de routage, les statistiques des interfaces, les connections masquées et les membres multicast.

### nisdomainname

nisdomainname affiche ou enregistre le nom de domaine NIS/YP du système.

### plipconfig

plipconfig permet de règler finement les paramètres d'un périphérique PLIP. En étant optimiste, il le rend plus rapide.

### rarp

Apparenté au programme arp, rarp manipule la table RARP du système.

### route

route est l'utilitaire utilisé pour manipuler les tables de routage IP.

### slattach

slattach attache une interface réseau à une ligne série, c'est−à−dire qu'il positionne une ligne normale pour un terminal dans un des nombreux modes "réseau".

### ypdomainname

updomainname affiche ou enregistre le nom de domaine NIS/YP du système.

## Dépendances d'installation de Net−tools

Dernière vérification effectuée auprès de la version 1.60.

Bash: bash, sh Binutils: ar, as, ld Fileutils: install, ln, ls, mv, rm Gcc: cc, cc1, collect2, cpp0
Make: make Sh−utils: echo

# Installer Perl–5.8.0

```
Estimation du temps de construction:           3.81 SBU
Estimation de l'espace disque requis:  52 Mo
```

## Installation de Perl

Installez Perl en lançant les commandes suivantes:

```
./configure.gnu --prefix=/usr &&
make &&
make install
```

Si vous souhaitez avoir plus de contrôle sur la façon dont perl configure sa propre construction, vous pouvez lancer le script interactif **`Configure`** et modifier la façon dont perl est construit. Si vous pensez pouvoir vivre avec la configuration que perl détecte automatiquement, alors utilisez simplement les commandes ci–dessus.

## Contenu de Perl

Dernière vérification effectuée auprès de la version 5.6.1.

### Programmes

a2p, c2ph, dprofpp, find2perl, h2ph, h2xs, perl, perl5.6.1, perlbug, perlcc, perldoc, pl2pm, pod2html, pod2latex, pod2man, pod2text, pod2usage, podchecker, podselect, pstruct, s2p et splain

### Descriptions

#### a2p

a2p est un traducteur perl vers awk.

#### c2ph

c2ph affiche les strutures C telles que générées avec "cc –g –S".

#### dprofpp

dprofpp affiche les données profile de perl.

#### find2perl

find2perl traduit les lignes de commande find en code Perl.

#### h2ph

h2ph convertit les fichiers d'entêtes .h C en fichiers d'entêtes .ph Perl.

### h2xs

h2xs convertit les fichiers d'entêtes .h C en extensions Perl.

### perl, perl5.6.1

perl est l'acronyme de Practical Extraction and Report Language (Langage pratique d'extraction et de rapport). Il combine certaines des meilleurs qualités de C, sed, awk et sh dans un puissant langage.

### perlbug

perlbug aide à générer des rapports de bugs sur perl ou sur un des modules qui sont inclus, et les envoie par mail.

### perlcc

perlcc génère des exécutables à partir de programmes Perl.

### perldoc

perldoc recherche une documentation au format .pod, intégré dans l'arbre d'installation de Perl ou dans un script perl, et l'affiche via "pod2man | nroff −man | $PAGER".

### pl2pm

pl2pm est un outil aidant à la conversion des bibliothèques style Perl4 en des modules de bibliothèques style Perl5.

### pod2html

pod2html convertit en HTML des fichiers au format pod.

### pod2latex

pod2latex convertit en LaTeX des fichiers au format pod.

### pod2man

pod2man convertit en *roff des fichiers au format pod.

### pod2text

pod2text convertit en ASCII des fichiers au format pod.

### pod2usage

pod2usage affiche les messages d'usage à partir des documentations intégrées.

### podchecker

podchecker vérifie la syntaxe des fichiers de documentation pod.

### podselect

podselect envoie sur la sortie standard les sections sélectionnées d'une documentation pod.

### pstruct

pstruct affiche les structures C telles que générées à partir de la commande "cc −g −S".

### s2p

s2p est un traducteur sed vers perl.

### splain

splain est un programme qui force de manière importante les messages de diagnostique avec Perl.

## Bibliothèques

attrs.so, B.so, ByteLoader.so, DProf.so, Dumper.so, DynaLoader.a, Fcntl.so, Glob.so, Hostname.so, IO.so, libperl.a, Opcode.so, Peek.so, POSIX.so, re.so, SDBM_File.so, Socket.so, Syslog.so et SysV.so

## Descriptions

### attrs

Aucune description disponible.

### B

Aucune description disponible.

### ByteLoader

Aucune description disponible.

### DProf

Aucune description disponible.

### Dumper

Aucune description disponible.

### DynaLoader

Aucune description disponible.

### Fcntl

Aucune description disponible.

### Glob

Aucune description disponible.

### Hostname

Aucune description disponible.

### IO

Aucune description disponible.

### libperl

Aucune description disponible.

### Opcode

Aucune description disponible.

### Peek

Aucune description disponible.

### POSIX

Aucune description disponible.

### re

Aucune description disponible.

### SDBM_File

Aucune description disponible.

### Socket

Aucune description disponible.

### Syslog

Aucune description disponible.

### SysV

Aucune description disponible.

## Dépendances d'installation de Perl

Dernière vérification effectuée auprès de la version 5.6.1.

Bash: sh Binutils: ar, as, ld, nm Diffutils: cmp Fileutils: chmod, cp, ln, ls, mkdir, mv, rm, touch
Gcc: cc, cc1, collect2, cpp0, gcc Grep: egrep, grep Make: make Gawk: awk Sed: sed
Sh−utils: basename, date, echo, expr, hostname, pwd, uname, whoami
Textutils: cat, comm, sort, split, tr, uniq, wc

# Installer Linux threads–2.2.5 man pages

```
Estimation du temps de construction:           0.01 SBU
Estimation de l'espace disque requis:  1,5 Mo
```

## Installation des pages man Linux threads

Déballez le package glibc–linuxthreads et vous remarquerez que vous obtenez deux nouveaux répertoires. Entrez dans le répertoire `linuxthreads`, et non pas le répertoire `linuxthreads_db`.

Installez les pages man Linux threads en lançant les commandes suivantes:

```
make -C man &&
make -C man install
```

## Contenu des pages man Linux threads

Dernière vérification effectuée auprès de la version 2.2.5.

### Fichiers de support

Différentes pages man sur l'API threads de Linux.

### Descriptions

#### Pages man

Contient les pages man concernant l'API Glibc des threads Linux.

## Dépendances d'installation des pages man de Linux Threads

Pas encore vérifié.

# Installer M4–1.4

```
Estimation du temps de construction:           0.08 SBU
Estimation de l'espace disque requis:  3 Mo
```

## Installation de M4

Installez M4 en lançant les commandes suivantes:

```
./configure --prefix=/usr &&
make &&
make install
```

## Contenu de M4

Dernière vérification effectuée auprès de la version 1.4.

### Programmes

m4

### Descriptions

#### m4

m4 est un processeur de macro. Il copie l'entrée sur la sortie en développant les macros au fur et à mesure. Les macros sont soit prédéfinies, soit définies par l'utilisateur, et peuvent prendre un nombre quelconque d'arguments. En plus de développer les macros, m4 a des fonctions prédéfinies pour inclure des fichiers nommés, lancer des commandes Unix, faire de l'arithmétique d'entiers, manipuler du texte de diverses manières, faire de la récursivité, etc. m4 peut être utilisé soit comme frontal d'un compilateur, ou comme processeur de macros à part.

## Dépendances d'installation de M4

Dernière vérification effectuée auprès de la version 1.4.

Bash: sh Binutils: ar, as, ld, ranlib Diffutils: cmp Fileutils: chmod, cp, install, mv, rm Make: make
Gcc: cc1, collect2, cpp0, gcc Grep: egrep, grep Sed: sed Sh–utils: date, echo, hostname
Textutils: cat, tr

# Installer Texinfo–4.2

```
Estimation du temps de construction:           0.43 SBU
Estimation de l'espace disque requis:  12 Mo
```

## Installation de Texinfo

Installez Texinfo en exécutant les commandes suivantes:

```
./configure --prefix=/usr &&
make &&
make install &&
make TEXMF=/usr/share/texmf install-tex
```

## Explication des commandes

**`make TEXMF=/usr/share/texmf install-tex`:** Ceci installe les composants texinfo qui font partie d'une installation TeX. Même si TeX n'est pas installé sur LFS, ils sont installés ici pour compléter l'installation de texinfo.

## Contenu de Texinfo

Dernière vérification effectuée auprès de la version 4.2.

### Programmes

info, infokey, install–info, makeinfo, texi2dvi et texindex

### Descriptions

#### info

Le programme info lit les documents de type Info, habituellement stockés dans le répertoire /usr/doc/info. Les documents info sont identiques aux pages de manuel, mais ils tendent à être plus détaillés et n'expliquent pas seulement les différentes options d'un programme.

#### infokey

infokey compile un fichier source contenant des personnalisations Info en un format binaire.

#### install–info

Le programme install–info met à jour les documents info. Lorsque le programme info est invoqué, il présente une liste des sujets disponibles (c'est–à–dire disponibles dans les documents info). Le programme install–info est utilisé pour tenir à jour cette liste. Si des fichiers info sont effacés manuellement, vous devez aussi supprimer les entrées correspondantes dans le fichier d'index. Ce programme le fait et fait aussi l'inverse lorsque des documents info sont ajoutés.

### makeinfo

Le programme makeinfo traduit des documents sources Texinfo en d'autres formats. Les formats disponibles sont : fichiers info, texte et HTML.

### texi2dvi

Le programme texi2dvi affiche les documents Texinfo.

### texindex

Le programme texindex est utilisé pour trier les fichiers d'index Texinfo.

## Dépendances d'installation de Texinfo

Dernière vérification effectuée auprès de la version 4.0.

Bash: sh Binutils: ar, as, ld, ranlib Diffutils: cmp Fileutils: chmod, install, ln, ls, mkdir, mv, rm
Gcc: cc1, collect2, cpp0, gcc Grep: egrep, fgrep, grep Make: make Sed: sed
Sh−utils: basename, echo, expr, hostname, sleep Texinfo: makeinfo Textutils: cat, tr

# Installation de Autoconf–2.53

```
Estimation du temps de construction :           0.05 SBU
Estimation de l'espace disque requis :  6 Mo
```

# Installation de Autoconf

Installez Autoconf en lançant les commandes suivantes:

```
./configure --prefix=/usr &&
make &&
make install
```

# Contenu de Autoconf

Dernière vérification effectuée auprès de la version 2.53.

## Programmes

autoconf, autoheader, autom4te, autoreconf, autoscan, autoupdate et ifnames

## Descriptions

### autoconf

autoconf est un outil destiné à produire des scripts shell qui configurent automatiquement les packages de code source logiciel pour les adapter à de nombreuses sortes de systèmes de type Unix. Les scripts de configuration produits par autoconf sont indépendants d'autoconf lorsqu'ils sont exécutés, aussi leurs utilisateurs n'ont pas besoin d'avoir Autoconf installé.

### autoheader

Le programme autoheader permet de créer un fichier modèle des déclarations C #define, pour l'usage de configure.

### autom4te

autom4te lance M4 de GNU sur des fichiers.

### autoreconf

Si il y a beaucoup de scripts configure générés par autoconf, le programme autoreconf peut vous faire gagner du temps. Il exécute autoconf et autoheader (si nécessaire) de façon répétée pour reconstruire les scripts configure d'autoconf et les modèles d'entête de configuration dans l'arborescence rattachée au répertoire courant.

### autoscan

Le programme autoscan peut aider à créer un fichier configure.in pour un package logiciel. autoscan examine les fichiers sources dans l'arborescence de dossiers. Si aucun répertoire n'est fourni sur la ligne de commande, il utilise le répertoire courant. Il cherche dans les fichiers sources les problèmes communs de portabilité et

crée un fichier configure.scan qui est un préliminaire à configure.in pour ce package.

### autoupdate

Le programme autoupdate met à jour le fichier configure.in qui appelle les macros autoconf par leurs anciens noms, afin qu'il utilise les noms de macros actuels.

### ifnames

ifnames peut être utile lors de l'écriture d'un configure.in pour un package. Il écrit les identifiants déjà utilisés par le package dans des conditions pour le pré−processeur C. Si un package a déjà été configuré pour avoir une certaine portabilité, ce programme peut aider à identifier ce que son script configure doit vérifier. Ceci peut aider à combler certains manques dans un fichier configure.in généré par autoscan.

## Dépendances d'installation d'Autoconf

Dernière vérification effectuée auprès de la version 2.52.

Bash: sh Diffutils: cmp Fileutils: chmod, install, ln, ls, mkdir, mv, rm Grep: fgrep, grep M4: m4
Make: make Gawk: gawk Sed: sed Sh−utils: echo, expr, hostname, sleep, uname Texinfo: install−info
Textutils: cat, tr

# Installation de Automake–1.6.3

```
Estimation du temps de construction :           0.03 SBU
Estimation de l'espace disque requis :  6 Mo
```

# Installation de Automake

Installez Automake en lançant les commandes suivantes:

```
./configure --prefix=/usr &&
make install
```

# Contenu de Automake

Dernière vérification effectuée auprès de la version 1.6.2.

## Programmes

acinstall, aclocal, aclocal–1.6, automake, automake–1.6, compile, config.guess, config.sub, depcomp, elisp–comp, install–sh, mdate–sh, missing, mkinstalldirs, py–compile, ylwrap

## Descriptions

### acinstall

acinstall est un script qui installe les fichiers M4 style aclocal.

### aclocal, aclocal–1.6

automake inclut un certain nombre de macros Autoconf qui peuvent être utilisées dans les packages, certaines d'entre elles sont nécessaires par automake dans certaines situations. Ces macros doivent être définies dans le fichier aclocal.m4, sinon elles ne seront pas vues par autoconf.

Le programme aclocal générera automatiquement des fichiers aclocal.m4 basés sur le contenu de configure.in. Ceci fournit un moyen commode d'obtenir des macros fournies par automake, sans avoir à rechercher ici ou là. De plus, le mécanisme aclocal est extensible à l'usage par d'autres packages.

### automake, automake–1.6

Pour créer tous les fichiers makefile.in d'un package, exécutez le programme automake dans le répertoire racine de ce package, sans argument. automake trouvera automatiquement chaque Makefile.am approprié (en parcourant configure.in) et générera les Makefile.in correspondants.

### compile

compile est un script qui agit comme un lien avec les compilateurs.

### config.guess

config.guess est un script qui essaie de deviner le nom canonique du système.

### config.sub

config.sub est un script contenant une sous−routine de validation de configuration.

### depcomp

depcomp est un script qui compile un programme en générant les dépendances comme effets de bord.

### elisp−comp

elisp−comp est un script qui compile les fichiers .el.

### install−sh

install−sh est un script qui installe un programme, script ouf fichier de données.

### mdate−sh

mdate−sh est un script qui affiche l'heure de modification d'un fichier ou d'un répertoire.

### missing

missing est un script qui agit comme un tube commun pour les quelques programmes GNU manquants lors d'une installation.

### mkinstalldirs

mkinstalldirs est un script qui crée une hiérarchie de répertoires.

### py−compile

py−compile est un script qui compile un programme Python.

### ylwrap

ylwrap est un script qui agit comme un lien en invoquant lex/yacc.

## Dépendances d'installation d'Automake

Dernière vérification effectuée auprès de la version 1.5.

Bash: sh Diffutils: cmp Fileutils: chmod, install, ls, mkdir, mv, rm, rmdir Grep: fgrep, grep
Make: make Perl: perl Sed: sed Sh−utils: echo, expr, hostname, sleep Texinfo: install−info
Textutils: cat, tr

# Installation de Bash–2.05a

```
Estimation du temps de construction :           0.82 SBU
Estimation de l'espace disque requis :  14 Mo
```

# Installation de Bash

Installez Bash en lançant les commandes suivantes:

```
./configure --prefix=/usr --bindir=/bin &&
make &&
make install &&
exec /bin/bash --login
```

# Contenu de Bash

Dernière vérification effectuée auprès de la version 2.05a.

## Programmes

bash, sh (lien vers bash) et bashbug

## Descriptions

### bash

bash est le Bourne–Again SHell, un interpréteur de commandes très largement répandu sur les systèmes Unix. bash est un programme qui lit depuis l'entrée standard, le clavier. L'utilisateur tape quelque chose et le programme évaluera ce qu'il a tapé et fera quelque chose avec, comme lancer un programme.

### bashbug

bashbug est un shell de script afin d'aider l'utilisateur à composer et à expédier par courrier électronique sous un format standard les rapports de bogues concernant bash.

### sh

sh est un lien symbolique vers le programme bash. Quand bash est invoqué en tant que sh, il essaye de simuler le comportement du démarrage des versions historiques de sh aussi similairement que possible, tout en restant conforme au standard POSIX aussi.

# Dépendances d'installation de Bash

Dernière vérification effectuée auprès de la version 2.05a.

Bash: bash, sh Binutils: ar, as, ld, ranlib, size Diffutils: cmp
Fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm Gcc: cc, cc1, collect2, cpp0, gcc
Grep: egrep, grep Make: make Gawk: awk Sed: sed
Sh–utils: basename, echo, expr, hostname, sleep, uname Texinfo: install–info Textutils: cat, tr, uniq

# Installer File–3.39

```
Estimation du temps de construction:           0.21 SBU
Estimation de l'espace disque requis:  2 Mo
```

## Installation de File

Installez File en lançant les commandes suivantes :

```
./configure --prefix=/usr --datadir=/usr/share/misc &&
make &&
make install
```

## Contenu de File

Dernière vérification effectuée auprès de la version 3.39.

### Programmes

file

### Description

#### file

file teste le fichier spécifié et essaie de le classifier. Il y a trois séries de tests, exécutées dans cet ordre : tests du système de fichiers, tests du nombre magique et tests de langage. Le premier test à s'exécuter avec succès affiche le type du fichier traité.

## Dépendances d'installation de File

Dernière vérification effectuée auprès de la version 3.37.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh Binutils: as, ld Diffutils: cmp Fileutils: chmod, install, ln, ls, mv, rm, touch Gcc: cc1, collect2, cpp0, gcc Grep: egrep, grep M4: m4 Make: make Gawk: gawk Sed: sed Sh–utils: echo, expr, hostname, sleep Texinfo: makeinfo Textutils: cat, tr

# Installer Libtool–1.4.2

```
Estimation du temps de construction:           0.15 SBU
Estimation de l'espace disque requis:  7 Mo
```

# Installation de Libtool

Installez Libtool en exécutant les commandes suivantes:

```
./configure --prefix=/usr &&
make &&
make install
```

# Contenu de libtool

Dernière vérification effectuée auprès de la version 1.4.2.

## Programmes

libtool et libtoolize

## Descriptions

### libtool

libtool fournit des services généraux de support de construction de bibliothèques.

### libtoolize

libtoolize fournit un moyen standard d'ajouter un support libtool à un package.

## Bibliothèques

libltdl.a, libltdl.so (lien vers libltdl.so.3.1.0), libltdl.so.3 (lien vers libltdl.so.3.1.0) et libltdl.so.3.1.0

## Descriptions

### libltdl, libltdl.so.3, libltdl.so.3.1.0

libltdl est une petite bibliothèque qui a pour but de masquer aux programmeurs les difficultés variées liées à l'ouverture dynamique des bibliothèques.

# Dépendances d'installation de Libtool

Dernière vérification effectuée auprès de la version 1.4.2.

Bash: sh Binutils: ar, as, ld, nm, ranlib, strip Diffutils: cmp
Fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm, rmdir Gcc: cc, cc1, collect2, cpp0
Glibc: ldconfig Grep: egrep, fgrep, grep Make: make Sed: sed
Sh–utils: echo, expr, hostname, sleep, uname Texinfo: install–info Textutils: cat, sort, tr, uniq

# Installation de Bin86–0.16.3

```
Estimation du temps de construction :           0.07 SBU
Estimation de l'espace disque requis :  2 Mo
```

## Installation de Bin86

Ce package est nécessaire seulement si vous décidez d'utiliser Lilo sur votre système LFS. Si vous escomptez utiliser un autre outil tel que Grub, vous n'aurez pas besoin de bin86. Vérifiez la documentation de votre chargeur préféré pour voir si vous avez besoin du package bin86 (habituellement seulement ld86 et/ou as86 de ce package sont requis).

Gardez à l'esprit qu'il n'y a pas seulement les chargeurs qui utilisent le package bin86. Il y a toujours une chance qu'un autre package ait besoin de programmes de ce package, donc garder tout ça à l'esprit si vous décidez de le sauter.

Installez Bin86 en exécutant les commandes suivantes:

```
make &&
make PREFIX=/usr install
```

## Contenu de Bin86

Dernière vérification effectuée auprès de la version 0.16.3.

### Programmes

as86, as86_encap, ld86, nm86 (lien vers objdump86), objdump86 et size86 (lien vers objdump86)

### Descriptions

#### as86

as86 est un assembleur pour les processeurs 8086...80386.

#### as86_encap

as86_encap est un script shell pour appeller as86 et convertir le binaire créé en un fichier C prog.v à inclure ou à lier avec des programmes comme les installeurs de blocs de démarrage.

#### ld86

ld86 ne comprends que les fichiers objets produits par l'assembleur as86. Il peut les lier dans un exécutable I&D soit impur, soit séparé.

#### nm86

La table des symboles du fichier binaire.

#### objdump86

Il affiche des informations détaillées sur un fichier binaire.

#### size86

Il résume les tailles des données dans un fichier binaire.

## Dépendances d'installation de Bin86

Dernière vérification effectuée auprès de la version 0.16.0.

Bash: sh Binutils: as, ld, strip Fileutils: chmod, install, ln, mv Gcc: cc, cc1, collect2, cpp0
Make: make Sed: sed

# Installation de Bzip2–1.0.2

```
Estimation du temps de construction :           0.09 SBU
Estimation de l'espace disque requis :  3 Mo
```

# Installation de Bzip2

Installez Bzip2 en lançant les commandes suivantes :

```
make -f Makefile-libbz2_so &&
make &&
make install &&
cp bzip2-shared /bin/bzip2 &&
ln -s libbz2.so.1.0 libbz2.so &&
cp -a libbz2.so* /lib &&
rm /lib/libbz2.so &&
ln -s ../../lib/libbz2.so.1.0 /usr/lib/libbz2.so &&
rm /usr/bin/{bunzip2,bzcat,bzip2} &&
mv /usr/bin/{bzip2recover,bzless,bzmore} /bin &&
ln -s bzip2 /bin/bunzip2 &&
ln -s bzip2 /bin/bzcat
```

Bien que cela ne soit pas strictement parlant une partie d'un système LFS de base, il est intéressant de mentionner qu'un correctif pour Tar peut être téléchargé pour permettre au programme Tar de compresser et décompresser en utilisant facilement bzip2/bunzip2. Avec un programme Tar classique, vous devez utiliser des commandes comme **`bzcat fichier.tar.bz | tar -xv`** ou **`tar --use-compress-prog=bunzip2 -xvf fichier.tar.bz2`** pourutiliser bzip2 et bunzip2 avec tar. Ce correctif apporte l'option **`-j`** de manière à pouvoir décompresser une archive Bzip2avec la commande **`tar -xvfj file.tar.bz2`**. Appliquer cecorrectif sera mentionné plus tard lors de la ré–installation du package Tar.

# Explication de la commande

**`make -f Makefile-libbz2_so:`** cela provoque la construction de bzip2 en utilisant un fichier Makefile différent, dans ce cas le fichier Makefile–libbz2_so qui crée une bibliothèque dynamique libbz2.so et la lie avec les utilitaires de bzip2.

# Contenu de Bzip2

Dernière vérification effectuée auprès de la version 1.0.2.

## Programmes

bunzip2 (lien vers bzip2), bzcat (lien vers bzip2), bzcmp, bzdiff, bzegrep, bzfgrep, bzgrep, bzip2, bzip2recover, bzless et bzmore

## Descriptions

### bunzip2

bunzip2 décompresse les fichiers compressés avec bzip2.

#### bzcat

bzcat (ou bzip2 −dc) décompresse les fichiers spécifiés sur la sortie standard.

#### bzcmp, bzdiff

bzcmp et bzdiff sont utilisés pour appeler les programmes cmp et diff avec des fichiers compressés avec bzip2.

#### bzegrep, bzfgrep, bzgrep

bzegrep, bzfgrep et bzgrep appellent respectivement egrep, fgrep, or grep sur des fichiers compressés avec bzip2.

#### bzip2

bzip2 compresse les fichiers grâce à l'algorithme Burrows−Wheeler et au codage Huffman. La compression est généralement bien meilleure à celle obtenue avec des compresseurs plus conventionnels basés sur LZ77/LZ78, et approche les performances de la famille PPM des compresseurs statistiques.

#### bzip2recover

bzip2recover recupère les données de fichiers bzip2 endommagés.

#### bzless

bzless est un filtre permettant d'examiner des fichiers textes compressés, un écran à la fois, sur un terminal, de la même manière que less.

#### bzmore

bzmore est un filtre permettant d'examiner des fichiers textes compressés, un écran à la fois, sur un terminal, de la même manière que more.

### Bibliothèques

libbz2.a, libbz2.so (lien vers libbz2.so.1.0), libbz2.so.1.0 (lien vers libbz2.so.1.0.2) et libbz2.so.1.0.2

#### libbz2

libbz2 est la bibliothèque implémentant l'algorithme de compression Burrows−Wheeler.

## Dépendances d'installation de Bzip2

Dernière vérification effectuée auprès de la version 1.0.1.

Bash: sh Binutils: ar, as, ld, ranlib Fileutils: cp, ln, rm Gcc: cc1, collect2, cpp0, gcc Make: make

# Installer Ed–0.2

```
Estimation du temps de construction:           0.06 SBU
Estimation de l'espace disque requis:  3 Mo
```

## Installation de Ed

Ce package nécessite que son correctif lui soit appliqué avant que vous ne puissiez l'installer. Assurez–vous qu'il a été déballé avant de lancer les commandes d'installation.

Ed n'est pas quelque chose que vous utiliserez directement. Il est installé car il peut être utilisé par le programme patch si vous rencontrez un fichier correctif basé sur ed. Ceci arrive rarement car les correctifs basés sur diff sont préférés de nos jours.

Installez Ed en lançant les commandes suivantes :

```
patch -Np1 -i ../ed-0.2.patch &&
./configure --prefix=/usr &&
make &&
make install &&
mv /usr/bin/{ed,red} /bin
```

## Explication des commandes

**`patch -Np1 -i ../ed-0.2.patch :`** Ce correctif corrige une vulnérabilité dans ed qui concerne les liens symboliques. Les exécutables de ed créent dans /tmp des fichiers dont les noms sont prévisibles. En utilisant diverses attaques sur les liens symboliques, il est possible de faire écrire à ed dans des fichiers qu'il ne devrait pas, de changer des permissions de fichiers, etc...

## Contenu de Ed

Dernière vérification effectuée auprès de la version 0.2.

### Programmes

ed et red (lien vers ed)

### Description

#### ed

ed est un éditeur de texte par ligne. Il est utilisé pour créer, afficher, modifier et réaliser d'autres manipulations sur les fichiers texte.

#### red

red est un ed resteint : il ne peut que éditer les fichiers du répertoire courant et ne peut pas exécuter des commandes du shell.

# Dépendances d'installation de Ed

Dernière vérification effectuée auprès de la version 0.2.

Bash: sh Binutils: ar, as, ld, ranlib Diffutils: cmp Fileutils: chmod, cp, install, ln, mv, rm, touch Gcc: cc1, collect2, cpp0, gcc Grep: egrep, grep Make: make Sed: sed Sh−utils: hostname Textutils: cat, tr

# Installer Kbd–1.06

```
Estimation du temps de construction:           0.12 SBU
Estimation de l'espace disque requis:  8 Mo
```

## Installation de Kbd

Avant que Kbd ne soit installé, il faut que le fichier correctif de Kbd soit déballé. Assurez–vous qu'il soit déballé avant de lancer les commandes d'installation.

Installez Kbd en lançant les commandes suivantes:

```
patch -Np1 -i ../kbd-1.06-3.patch &&
./configure &&
make &&
make install
```

## Explications des commandes

**`patch -Np1 -i ../kbd-1.06-3.patch`**: Ce correctif résout deux problèmes. Le premier concerne le comportement de **`loadkeys -d`**, qui ne fonctionne plus sur la version courante de kbd. Il est nécessaire de corriger ceci, car les scripts de démarrage repose sur une utilisation propre de **`loadkeys -d`**. La seconde partie du correctif change un Makefile pour que certains utilitaires (setlogcons, setvesablank et getunimap), qui ne sont pas installés par défaut, soient quand même installés.

## Contenu de Kbd

Dernière vérification effectuée auprès de la version 1.06.

### Programmes

chvt, deallocvt, dumpkeys, fgconsole, getkeycodes, getunimap, kbd_mode, kbdrate, loadkeys, loadunimap, mapscrn, openvt, psfaddtable (lien vers psfxtable), psfgettable (lien vers psfxtable), psfstriptable (lien vers psfxtable), psfxtable, resizecons, setfont, setkeycodes, setleds, setlogcons, setmetamode, setvesablank, showfont, showkey, unicode_start, et unicode_stop

### Descriptions

#### chvt

chvt change le terminal virtuel de premier plan.

#### deallocvt

deallocvt désalloue les terminaux virtuels inutilisés.

#### dumpkeys

dumpkeys affiche les tables de conversion du clavier.

### fgconsole

fgconsole affiche le numéro du terminal virtuel actif.

### getkeycodes

getkeycodes affiche la table de correspondance de scancode en keycode du noyau.

### getunimap

getunimap affiche le unimap actuellement utilisé.

### kbd_mode

kbd_mode indique ou positionne le mode du clavier.

### kbdrate

kbdrate positionne les taux de répétition et de temporisation du clavier.

### loadkeys

loadkeys charge les tables de conversion du clavier.

### loadunimap

loadunimap charge la table de correspondance d'unicode en police du noyau.

### mapscrn

mapscrn charge une table de correspondance des caractères de sortie définie par l'utilisateur dans le pilote de la console. Notez qu'il est obsolète et que ses fonctionnalités sont reprises dans setfont.

### openvt

openvt lance un programme sur un nouveau terminal virtuel (VT).

### psfaddtable, psfgettable, psfstriptable, psfxtable

C'est un ensemble d'outils pour gérer les tables de caractères Unicode pour les polices de console.

### resizecons

resizeicons modifie la représentation de la taille de la console pour le noyau.

### setfont

Ceci vous permet de modifier les polices EGA/VGA dans la console.

### setkeycodes

setkeycodes charge les entrées de la table de correspondance scancode en keycode du noyau.

### setleds

setleds positionne les LED du clavier. Beaucoup de personnes trouvent utile d'avoir numlock activé par défaut, et, en utilisant ce programme, vous pourrez y arriver.

### setlogcons

setlogcons sends kernel messages to the console.

### setmetamode

setmetamode définit la gestion de la touche méta du clavier.

### setvesablank

Ceci vous laisse trafiquer l'économiseur d'écran matériel (pas de grilles−pain, seulement un écran vide).

### showfont

showfont affiche des données sur une police. L'information montrée comprend les informations sur la police, ses propriétés, les métriques des caractères, et les images des caractères.

### showkey

showkey examine les scancodes et les keycodes transmis par le clavier.

### unicode_start

unicode_start met la console en mode Unicode.

### unicode_stop

unicode_stop annule le mode unicode pour le clavier et la console.

## Dépendances d'installation de Kbd

Dernière vérification effectuée auprès de la version 1.06.

Bash: sh Binutils: as, ld, strip Bison: bison Diffutils: cmp Fileutils: cp, install, ln, mv, rm Flex: flex Gettext: msgfmt, xgettext Gcc: cc1, collect2, cpp0, gcc Grep: grep Gzip: gunzip, gzip Make: make Patch: patch Sed: sed Sh−utils: uname

# Installation de Diffutils–2.8.1

```
Estimation du temps de construction :           0.31 SBU
Estimation de l'espace disque requis :  6 Mo
```

# Installation de Diffutils

Installez Diffutils en lançant les commandes suivantes :

```
./configure --prefix=/usr &&
make &&
make install
```

# Contenu de Diffutils

Dernière vérification effectuée auprès de la version 2.8.1.

## Programmes

cmp, diff, diff3 et sdiff

## Descriptions

### cmp et diff

cmp et diff comparent tous les deux deux fichiers et indiquent les différences. Tous les deux disposent d'options particulières pour comparer les fichiers dans des situations différentes.

### diff3

La différence entre diff et diff3 est que diff compare 2 fichiers, alors que diff3 en compare 3.

### sdiff

sdiff joint deux fichiers et affiche interactivement le résultat.

# Dépendances d'installation de Diffutils

Dernière vérification effectuée auprès de la version 2.7.

Bash: sh Binutils: ld, as Diffutils: cmp Fileutils: chmod, cp, install, mv, rm
Gcc: cc1, collect2, cpp0, gcc Grep: egrep, grep Make: make Sed: sed Sh–utils: date, hostname
Textutils: cat, tr

# Installation de E2fsprogs–1.27

```
Estimation du temps de construction :           0.80 SBU
Estimation de l'espace disque requis :  13 Mo
```

## Installation de E2fsprogs

Installez E2fsprogs en lançant les commandes suivantes :

```
mkdir ../e2fsprogs-build &&
cd ../e2fsprogs-build &&
../e2fsprogs-1.27/configure --prefix=/usr --with-root-prefix="" \
    --enable-elf-shlibs &&
make &&
make install &&
make install-libs &&
install-info /usr/share/info/libext2fs.info /usr/share/info/dir
```

## Explication des commandes

**`--with-root-prefix="":`** La raison de fournir cette option est dûe à la configuration du Makefile de e2fsprogs. Certains programmes sont essentiels pour l'utilisation du système quand, par exemple, /usr n'est pas monté (comme le programme e2fsck). Ces programmes et bibliothèques appartiennent donc à des répertoires tel que /lib et /sbin. Si l'option n'est pas déclarée lors de la configuration de E2fsprog, il va placer ces programmes dans /usr, ce que nous ne voulons pas.

**`--enable-elf-shlibs:`** Cela crée des bibliothèques partagées que quelques programmes dans ce package peuvent utiliser.

**`make install-libs:`** Cela installe les bibliothèques partagées qui sont construites.

## Contenu du package E2fsprogs

Dernière vérification effectuée auprès de la version 1.27.

### Programmes

badblocks, chattr, compile_et, debugfs, dumpe2fs, e2fsck, e2image, e2label, fsck, fsck.ext2, fsck.ext3, lsattr, mk_cmds, mke2fs, mkfs.ext2, mklost+found, resize2fs, tune2fs et uuidgen

### Descriptions

#### badblocks

badblocks est utilisé pour rechercher les secteurs défectueux sur un périphérique (en général une partition de disque).

#### chattr

chattr modifie les attributs d'un fichier sur un système de fichier Linux ext2.

### compile_et

compile_et est utilisé pour convertir un tableau, regroupant des codes d'erreur et messages associés, en un fichier source C exploitable avec la librairie com_err.

### debugfs

Le programme debugfs est un débogueur de système de fichiers. Il peut être utilisé pour examiner et modifier l'état d'un système de fichiers ext2.

### dumpe2fs

dumpe2fs affiche le super−bloc et les informations de groupe de blocs pour le système de fichiers présent sur un périphérique spécifié.

### e2fsck et fsck.ext2

e2fsck et fsck.ext2 sont utilisés pour vérifier, et parfois réparer, un système de fichiers Linux ext2.

### e2image

e2image est utilisé pour sauvegarder les données critiques d'un système de fichiers ext2 vers un fichier.

### e2label

e2label affiche ou modifie le nom du système de fichiers sur un système de fichiers ext2 situé sur le périphérique spécifié.

### fsck

fsck est utilisé pour vérifier, et éventuellement réparer, un système de fichiers Linux.

### fsck.ext3

fsck.ext3 est utilisé pour vérifier, et éventuellement réparer, un système de fichiers Linux ext3.

### lsattr

lsattr liste les attributs d'un fichier sur un système de fichiers ext2.

### mk_cmds

L'utilitaire mk_cmds prends un fichier de commandes en entrée et produit en sortie un fichier source en C, dont le but est d'être utilisé avec la bibliothèque libss.

### mke2fs et mkfs.ext2

mke2fs est utilisé pour créer un système de fichiers Linux ext2 sur un périphérique (en général une partition de disque). mkfs.ext2 fait la même chose que mke2fs.

### mkfs.ext3

mkfs.ext3 est utilisé pour créer un système de fichiers ext3.

### mklost+found

mklost+found est utilisé pour créer le dossier lost+found dans le répertoire de travail courant d'un système de fichiers Linux ext2. mklost+found pré−alloue des blocs de disque pour le répertoire, pour le rendre utilisable par e2fsck.

### resize2fs

resize2fs est utilisé pour redimensionner un système de fichier ext2.

### tune2fs

tune2fs ajuste les paramètres d'optimisation de système de fichiers sur un système de fichiers Linux ext2.

### uuidgen

Le programme uuidgen crée un nouvel identifiant universel unique (UUID) en utilisant la bibliothèque libuuid. Le nouvel UUID peut raisonnablement être considéré comme unique parmi tous les UUID créés, sur le système local et sur d'autres systèmes, dans le passé ou le futur.

## Bibliothèques

libcom_err.[a,so], libe2p.[a,so], libext2fs.[a,so], libss.[a,so], libuuid.[a,so]

## Descriptions

### libcom_err

La routine d'affichage des erreurs.

### libe2p

libe2p est utilisé par dumpe2fs, chattr, et lsattr.

### libext2fs

La bibliothèque ext2fs a été créée pour permettre à des programmes utilisateurs de manipuler un système de fichiers ext2.

### libss

libss est utilisé par debugfs.

### libuuid

La bibliothèque libuuid est utilisé pour générer des identifiants uniques pour les objets qui peuvent être accessibles en dehors du système local.

# Dépendances d'installation d'E2fsprogs

Dernière vérification effectuée auprès de la version 1.25.

Bash: sh Binutils: ar, as, ld, ranlib, strip Diffutils: cmp
Fileutils: chmod, cp, install, ln, mkdir, mv, rm, sync Gcc: cc, cc1, collect2, cpp0 Glibc: ldconfig
Grep: egrep, grep Gzip: gzip Make: make Gawk: awk Sed: sed
Sh−utils: basename, echo, expr, hostname, uname Texinfo: makeinfo Textutils: cat, tr

# Installer Grep–2.5

```
Estimation du temps de construction:           0.22 SBU
Estimation de l'espace disque requis:  5 Mo
```

## Installation de Grep

Installez Grep en lançant les commandes suivantes:

```
./configure --prefix=/usr --bindir=/bin &&
make &&
make install
```

## Contenu de Grep

Dernière vérification effectuée auprès de la version 2.5.

### Programmes

egrep (lien vers grep), fgrep (lien vers grep) et grep

### Descriptions

#### egrep

egrep affiche les lignes des fichiers correspondant au motif d'une expression régulière étendue.

#### fgrep

fgrep affiche les lignes des fichiers contenant une expression littérale.

#### grep

grep affiche les lignes des fichiers correspondant au motif d'une expression régulière basique.

## Dépendances d'installation de Grep

Dernière vérification effectuée auprès de la version 2.4.2.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh Binutils: as, ld Diffutils: cmp
Fileutils: chmod, install, ls, mkdir, mv, rm Gettext: msgfmt, xgettext
Gcc: cc, cc1, collect2, cpp0, gcc Glibc: getconf Grep: egrep, fgrep, grep M4: m4 Make: make
Gawk: gawk Sed: sed Sh–utils: basename, echo, expr, hostname, sleep, uname
Texinfo: install–info, makeinfo Textutils: cat, tr

# Installer Gzip–1.2.4a

```
Estimation du temps de construction:           0.03 SBU
Estimation de l'espace disque requis:  2 Mo
```

## Installation de Gzip

Installez Gzip en lançant les commandes suivantes:

```
patch -Np1 -i ../gzip-1.2.4b.patch &&
./configure --prefix=/usr &&
cp gzexe.in{,.backup} &&
sed 's%"BINDIR"%/bin%' gzexe.in.backup > gzexe.in &&
make &&
make install &&
mv /usr/bin/gzip /bin &&
rm /usr/bin/{gunzip,zcat} &&
ln -s gzip /bin/gunzip &&
ln -s gzip /bin/zcat &&
ln -s gunzip /bin/uncompress
```

## Explication de la commande

**`patch -Np1 -i ../gzip-1.2.4b.patch`:** Ce correctif corrige une erreur de 'buffer overflow' qui arrive lorsqu'un nom de fichier a plus de 1020 caractères.

## Contenu de Gzip

Dernière vérification effectuée auprès de la version 1.2.4a.

### Programmes

gunzip (lien vers gzip), gzexe, gzip, uncompress (lien vers gunzip), zcat (lien vers gzip), zcmp, zdiff, zforce, zgrep, zmore et znew

### Description

#### gunzip, uncompress

gunzip et uncompress décompressent les fichiers qui ont été compressés par gzip.

#### gzexe

gzexe vous permet de compresser des exécutables et de les décompresser puis de les lancer automatiquement (avec cependant une baisse des performances).

#### gzip

gzip réduit la taille des fichiers en utilisant l'algorithme de compression Lempel–Ziv (LZ77).

### zcat

zcat décompresse, et écrit sur la sortie standard, une liste de fichiers donnée en ligne de commande ou sur l'entrée standard en lecture.

### zcmp

zcmp a le même fonctionnement que le programme cmp mais sur des fichiers compressés.

### zdiff

zdiff a le même fonctionnement que le programme diff mais sur des fichiers compressés.

### zforce

zforce force l'extension .gz sur les fichiers compressés par gzip de façon à ce que gzip ne les compresse pas une deuxième fois. Ceci peut être utile pour des fichiers dont le nom a été tronqué après un transfert de fichier.

### zgrep

zgrep a le même fonctionnement que le programme grep mais sur des fichiers compressés.

### zmore

zmore est un programme permettant d'avoir le comportement du programme more aussi bien sur des fichiers normaux que compressés.

### znew

znew convertit les fichiers de format .Z (compress) en .gz (gzip).

## Dépendances d'installation de Gzip

Dernière vérification effectuée auprès de la version 1.2.4a.

Bash: sh Binutils: as, ld, nm Fileutils: chmod, cp, install, ln, mv, rm
Gcc: cc1, collect2, cpp, cpp0, gcc Grep: egrep, grep Make: make Sed: sed Sh−utils: hostname
Textutils: cat, tr

# Installer Man−1.5k

```
Estimation du temps de construction:           0.05 SBU
Estimation de l'espace disque requis:  2 Mo
```

## Installation de Man

Lancez les commandes suivantes pour installer man:

```
patch -Np1 -i ../man-1.5k.patch &&
PATH=$PATH:/usr/bin:/bin \
    ./configure -default -confdir=/etc &&
make &&
make install
```

Note: Si vous souhaitez désactiver les séquences d'échappement SGR, vous devez éditer le fichier man.conf et ajouter l'argument **-c** à nroff.

Vous voudrez peut−être jeter un oeil à l'astuce man sur http://hints.linuxfromscratch.org/hints/man.txt qui traite du formatage et de la compression pour les pages de manuel.

## Explication des commandes

**PATH=$PATH:/usr/bin:/bin ./configure -default**: Le chemin de certains programmes sont écrits dans les fichiers de manuel. Malheureusement, le script configure récupère le dernier emplacement dans PATH plutôt que le premier où ce programme est trouvé. En ajoutant en suffixe /usr/bin:/bin au PATH pour la commande ./configure, nous nous assurons que man n'utilise pas les versions /static de nos programmes.

**patch -Np1 -i ../man-1.5k.patch:** Ce correctif mets en commentaire un des fichiers dans le fichier `man.conf` (MANPATH /usr/man) car il créera des résultats redondants si on l'utilise avec des programmes comme **whatis**. Il ajoute aussi l'option *−R* à la variable *PAGER* de manière à ce que les pages man soient affichées correctement.

## Contenu de Man

Dernière vérification effectuée auprès de la version 1.5k.

### Programmes

apropos, makewhatis, man, man2dvi, man2html et whatis

### Descriptions

#### apropos

apropos recherche des mots clés dans un ensemble de fichiers base de données contenant une description brève des commandes système et affiche le résultat sur la sortie standard.

### makewhatis

makewhatis lit toutes les pages de manuel contenues dans les sections de manpath fournies, ou les pages préformatées contenues dans les sections de catpath données. Pour chaque page, il écrit une ligne dans la base de données whatis. Chaque ligne contient le nom de la page et une brève description, séparés par un tiret. La description est extraite du contenu de la section NAME de la page du manuel.

### man

man formate et affiche les pages de manuel en ligne.

### man2dvi

man2dvi exporte une page man au format dvi.

### man2html

man2html exporte une page man au format html.

### whatis

whatis recherche des mots clés dans un ensemble de fichiers base de données contenant une description brève des commandes système et affiche le résultat sur la sortie standard. Seules les correspondances de mots complets sont affichées.

## Dépendances d'installation de Man

Dernière vérification effectuée auprès de la version 1.5i2.

Bash: sh Binutils: as, ld Fileutils: chmod, cp, install, mkdir, rm Gcc: c11, collect2, cpp0, gcc
Grep: grep Make: make Gawk: awk Sed: sed Sh−utils: echo Textutils: cat

# Installer Lilo–22.2

```
Estimation du temps de construction:           0.08 SBU
Estimation de l'espace disque requis:  3 Mo
```

# Installation de Lilo

Nous avons choisi Lilo comme chargeur car nous nous sentons à l'aise avec, mais vous désirez peut–être en choisir un autre. Fabio Fracassi a écrit une astuce sur GRUB, qui peut se trouver sur la page http://hints.linuxfromscratch.org/hints/grub–howto.txt.

Installez Lilo en lançant les commandes suivantes:

```
make &&
make install
```

Il semble que la compilation de ce package échoue sur certaines machines quand l'option de compilation –g est utilisée. Si vous ne pouvez pas compiler Lilo du tout, vous devriez essayer de retirer la valeur –g de la variable CFLAGS dans le fichier `Makefile`.

A la fin de l'installation, le processus make install écrira un message disant que /sbin/lilo doit être exécuté pour complèter la mise à jour. Ne le faites pas car c'est inutile: le fichier /etc/lilo.conf n'est pas encore présent. Nous terminerons l'installation de Lilo au chapitre 8.

Le prompt standard de LILO, ainsi que le menu, peuvent être remplacés par n'importe quel logo, par exemple celui de LFS. Martin Imobersteg a écrit une astuce là–dessus, qui est disponible sur la page http://hints.linuxfromscratch.org/hints/bootlogo.txt.

# Contenu de Lilo

Dernière vérification effectuée auprès de la version 22.2.

## Programmes

lilo, mkrescue et keytab–lilo.pl

## Descriptions

### lilo

lilo installe le chargeur de boot de Linux, qui est utilisé pour démarrer un système Linux.

### mkrescue

mkrescue crée une disquette bootable de secours en utilisant un noyau existant et un ramdisk initial.

### keytab–lilo.pl

keytab–lilo.pl compile les tables de définition de codage de clavier dans un format que lilo peut utiliser pour initialiser le type de clavier lors du démarrage.

# Dépendances d'installation de Lilo

Dernière vérification effectuée auprès de la version 22.1.

Bash: sh Bin86: as86, ld86 Binutils: as, ld, strip Fileutils: cp, dd, ln Gcc: cc, cc1, collect2, cpp0
Make: make Sed: sed Textutils: cat

# Installer Make–3.79.1

```
Estimation du temps de construction:           0.22 SBU
Estimation de l'espace disque requis:  6 Mo
```

## Installation de Make

Installez Make en lançant les commandes suivantes:

```
./configure --prefix=/usr &&
make &&
make install &&
chgrp root /usr/bin/make &&
chmod 755 /usr/bin/make
```

## Explication des commandes

Par défaut, `/usr/bin/make` est installé avec un setgid kmem. Cela est nécessaire sur certains systèmes pour qu'il puisse vérifier la charge moyenne en utilisant `/dev/kmem`. Néanmoins, sur les systèmes Linux, setgid kmem n'est pas nécessaire, donc nous le supprimons de l'exécutable `make`. Ceci corrige aussi des problèmes quand `make` ignore certaines variables comme *LD_LIBRARY_PATH*.

## Contenu de Make

Dernière vérification effectuée auprès de la version 3.79.1.

### Programmes

make

### Descriptions

#### make

make détermine automatiquement quelles parties d'un gros programme ont besoin d'être recompilées, et exécute les commandes pour les recompiler.

## Dépendances d'installation de Make

Dernière vérification effectuée auprès de la version 3.79.1.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh Binutils: as, ld Diffutils: cmp
Fileutils: chgrp, chmod, install, ls, mv, rm Gcc: cc, cc1, collect2, cpp0, gcc Glibc: getconf
Grep: egrep, fgrep, grep M4: m4 Make: make Gawk: gawk Sed: sed
Sh–utils: basename, echo, expr, hostname, sleep, uname Texinfo: install–info, makeinfo
Textutils: cat, tr

# Installer Modutils–2.4.19

```
Estimation du temps de construction:           0.13 SBU
Estimation de l'espace disque requis:  3 Mo
```

# Installation de Modutils

Installez Modutils en lançant les commandes suivantes:

```
./configure &&
make &&
make install
```

# Contenu de Modutils

Dernière vérification effectuée auprès de la version 2.4.16.

## Programmes

depmod, genksyms, insmod, insmod_ksymoops_clean, kallsyms (lien vers insmod), kernelversion, ksyms (lien vers insmod), lsmod (lien vers insmod), modinfo, modprobe (lien vers insmod) et rmmod (lien vers insmod)

## Descriptions

### depmod

depmod gère la description des dépendances entre les modules du noyau.

### genksyms

genksyms examine (sur l'entrée standard) la sortie de gcc –E source.c et génère un fichier contenant des informations de version.

### insmod

insmod installe un module chargeable dans le noyau en cours d'utilisation.

### insmod_ksymoops_clean

insmod_ksymoops_clean efface les fichiers ksyms et les modules sauvegardés n'ayant pas été accédés depuis 2 jours.

### kallsyms

kallsyms extrait tous les symboles pour le débuggage du noyau.

### kernelversion

kernelversion affiche la version majeure du noyau en cours d'utilisation.

### ksyms

ksyms affiche les symboles exportés du noyau.

### lsmod

lsmod affiche des informations sur les modules chargés.

### modinfo

modinfo examine un fichier objet associé à un module noyau pour afficher toutes les informations qu'il peut glaner.

### modprobe

modprobe utilise un fichier de dépendances du style de make, créé par depmod, pour automatiquement charger les modules associés situés dans les arborescences prévues à cet effet.

### rmmod

rmmod décharge les modules chargeables du noyau en cours d'utilisation.

## Dépendances d'installation de Modutils

Dernière vérification effectuée auprès de la version 2.4.12.

Bash: sh Binutils: ar, as, ld, ranlib, strip Bison: bison Diffutils: cmp
Fileutils: chmod, install, ln, mkdir, mv, rm Flex: flex Gcc: cc, cc1, collect2, cpp0, gcc
Grep: egrep, grep Make: make Sed: sed Sh−utils: basename, expr, hostname, uname Textutils: cat, tr

# Installer Netkit−base−0.17

```
Estimation du temps de construction:           0.03 SBU
Estimation de l'espace disque requis:  1 Mo
```

# Installation de Netkit−base

Installez Netkit−base en lançant les commandes suivantes:

```
./configure &&
make &&
make install &&
cp etc.sample/{services,protocols} /etc
```

Il y a d'autres fichiers dans le répertoire `etc.sample` qui pourraient vous intéresser.

# Contenu de Netkit−base

Dernière vérification effectuée auprès de la version 0.17.

## Program Files

inetd et ping

## Descriptions

### inetd

inetd est le père de tous les démons. Il est à l'écoute des connections, et transfère l'appel au démon approprié.

### ping

ping envoie des paquets ICMP ECHO_REQUEST à un hôte et détermine son temps de réponse.

# Dépendances d'installation de Netkit−base

Dernière vérification effectuée auprès de la version 0.17.

Bash: sh Binutils: as, ld, strip Fileutils: cp, install, rm Make: make Gcc: cc1, collect2, cpp0, gcc
Sed: sed Sh−utils: date Textutils: cat

# Installer Patch–2.5.4

```
Estimation du temps de construction:           0.10
Estimation de l'espace disque requis:  2 Mo
```

# Installation de Patch

Installez Patch en lançant les commandes suivantes:

```
CPPFLAGS=-D_GNU_SOURCE \
    ./configure --prefix=/usr &&
make &&
make install
```

# Contenu de Patch

Dernière vérification effectuée auprès de la version 2.5.4.

## Programmes

patch

## Descriptions

### patch

Le programme patch modifie un fichier en fonction d'un fichier de patch, aussi appelé correctif. Ce correctif est habituellement une liste créée par le programme diff et contient les instructions pour modifier le fichier original. Patch est très souvent utilisé pour la modification de code source car il fait économiser du temps et de la taille mémoire. Imaginez un package faisant 1 Mo. La version suivante de ce package n'a modifié que deux fichiers de la précédente version. On peut alors le considérer entièrement comme un nouveau package d'1 Mo ou juste sous la forme d'un correctif d'1 Ko qui viendra mettre à jour la précédente version pour la rendre identique à la dernière version réalisée. Ainsi, si ma première version avait déjà été téléchargée, un correctif évite un deuxième téléchargement conséquent.

# Dépendances d'installation de Patch

Dernière vérification effectuée auprès de la version 2.5.4.

Bash: sh Binutils: as, ld Diffutils: cmp Fileutils: chmod, install, mv, rm
Gcc: cc, cc1, collect2, cpp0, gcc Glibc: getconf Grep: egrep, grep Make: make Sed: sed
Sh–utils: echo, expr, hostname, uname Textutils: cat, tr

# Installer Procinfo–18

```
Estimation du temps de construction:           0.02 SBU
Estimation de l'espace disque requis:  168 Ko
```

## Installation de Procinfo

Installez Procinfo en lançant les commandes suivantes:

```
make LDLIBS=-lncurses &&
make install
```

## Explication des Commandes

**`make LDLIBS=-lncurses` :** Ceci utilisera –lncurses à la place de –ltermcap lors de la construction de procinfo. Ceci est fait car libtermcap est déclarée obsolète en faveur de libncurses.

## Contenu de Procinfo

Dernière vérification effectuée auprès de la version 18.

### Programmes

lsdev, procinfo et socklist

### Descriptions

#### lsdev

lsdev récupère des informations sur le matériel installé dans votre ordinateur à partir des fichiers sur les interruptions, ports d'entrées/sorties et ports dma dans le répertoire /proc, vous donnant un rapide aperçu de qui utilise quoi (quel matériel utilise tel adresse d'entrée/sortie et tel IRQ et tels canaux DMA).

#### procinfo

procinfo récupère des informations à partir du répertoire /proc et les affiche proprement formatté sur la sortie standard.

#### socklist

Il s'agit d'un script Perl vous donnant une liste de tous les sockets ouverts, énumérant leurs types, port, inode, identifiant utilisateur, numéro de processus, identifiant du fichier ainsi que le programme à qui ils appartiennent.

## Dépendances d'installation de Procinfo

Dernière vérification effectuée auprès de la version 18.

Binutils: as, ld Fileutils: install, mkdir Gcc: cc1, collect2, cpp0, gcc Make: make

# Installer Procps–2.0.7

```
Estimation du temps de construction:           0.14 SBU
Estimation de l'espace disque requis:  2 Mo
```

## Installation de Procps

Ce programme nécessite que son correctif lui soit appliqué. Assurez–vous que celui–ci a bien été déballé avant de lancer les instructions d'installation.

Installez Procps en lançant les commandes suivantes:

```
patch -Np1 -i ../procps-2.0.7.patch &&
make &&
make XSCPT="" install &&
mv /usr/bin/kill /bin
```

## Explication des Commandes

**`patch -Np1 -i ../procps-2.0.7.patch:`** Ce correctif corrige un problème avec locale qui fait s'arrêter brutalement top avec certains paramétrages de locale.

**`make XSCPT="" install:`** Cela positionne la variable XSCPT du Makefile à une valeur nulle afin que l'installation de XConsole soit désactivée. Sinon "Make install" essaye de copier le fichier XConsole dans /usr/X11R6/lib/X11/app–defaults. Et ce répertoire n'existe pas, car X n'est pas installé.

## Contenu de Procps

Dernière vérification effectuée auprès de la version 2.0.7.

### Programmes

free, kill, oldps, pgrep, pkill, ps, skill, snice, sysctl, tload, top, vmstat, w et watch

### Descriptions

#### free

free affiche le nombre total de mémoire, physique ou swap, disponible ou utilisée dans le système. Il affiche également la mémoire partagée et les tampons utilisés par le noyau.

#### kill

kill envoie des signaux aux processus.

#### oldps et ps

ps donne un état des processus en cours.

### pgrep

pgrep cherche les processus à partir de leur nom et d'autres attributs.

### pkill

pkill envoie des signaux en se basant sur leur nom et d'autres attributs.

### skill

skill envoie des signaux aux processus remplissant certains critères.

### snice

snice change la priorité d'ordonnancement des processus remplissant certains critères.

### sysctl

sysctl modifie les paramètres noyau au lancement.

### tload

tload affiche un graphique de la charge moyenne du système en cours au terminal spécifié (ou au tty du processus tload si aucun n'est spécifié).

### top

top affiche l'activité du processeur en temps réel.

### vmstat

vmstat fournit des informations sur les processus, la mémoire, la pagination, les entrées−sorties, les interruptions et l'activité du processeur.

### w

w affiche des informations sur les utilisateurs actuellement connectés, et leurs processus.

### watch

watch exécute une commande répétitivement, en affichant ses sorties (au premier écran plein).

## Bibliothèques

libproc.so

## Descriptions

### libproc

libproc est une bibliothèque avec laquelle la plupart des programmes de ce package ont été lié pour sauver de l'espace disque, en implémentant des fonctions communes une seule fois.

## Dépendances d'installation de Procps

Dernière vérification effectuée auprès de la version 2.0.7.

Bash: sh Binutils: as, ld, strip Fileutils: install, ln, mv, rm Gcc: cc1, collect2, cpp0, gcc Grep: grep
Make: make Gawk: awk Sed: sed Sh−utils: basename, pwd Textutils: sort, tr

# Installer Psmisc–21

```
Estimation du temps de construction:           0.11 SBU
Estimation de l'espace disque requis:  2 Mo
```

## Installation de Psmisc

Installez Psmisc en lançant les commandes suivantes:

```
./configure --prefix=/usr --exec-prefix=/ &&
make &&
make install
```

Psmisc installe la page de man `/usr/share/man/man1/pidof.1`, mais le programme pidof de Psmisc n'est pas installé par défaut. Généralement ceci n'est pas un problème car nous installerons le package sysvinit plus tard, qui nous fournit un meilleur programme pidof.

C'est à vous de décider maintenant si vous allez utiliser le package Sysvinit qui fournit un programme pidof, ou non. Si vous le souhaitez, vous devriez enlever la page man pidof de Psmisc en exécutant:

```
rm /usr/share/man/man1/pidof.1
```

Si vous n'utilisez pas Sysvinit, vous devriez complèter l'installation de ce package en créant le lien symbolique `/bin/pidof` en exécutant:

```
ln -s killall /bin/pidof
```

## Explication des Commandes

**`--exec-prefix=/`:** Cela fera que les programmes seront installés dans `/bin` plutôt que dans `/usr/bin`. Les programmes de ce package sont souvent utilisés dans des scripts de démarrage, alors ils devraient être dans le répertoire `/bin` afin qu'ils puissent être utilisés quand la partition `/usr` n'est pas encore montée.

## Contenu de Psmisc

Dernière vérification effectuée auprès de la version 21.

### Programmes

fuser, killall et pstree

A noter que dans LFS nous n'installons pas le lien pidof par défaut car nous utilisons pidof de sysvinit à la place.

### Descriptions

#### fuser

fuser affiche les PIDs des processus qui utilisent les fichiers ou les systèmes de fichiers spécifiés.

### killall

killall envoie un signal à tous les processus faisant tourner l'une des commandes spécifiées.

### pstree

pstree présente les processus tournant sous forme d'arborescence.

## Dépendances d'installation de Psmisc

Dernière vérification effectuée auprès de la version 20.2.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh Bison: bison Binutils: as, ld
Diffutils: cmp Fileutils: chmod, install, ls, mkdir, mv, rm Gettext: msgfmt, xgettext
Gcc: cc, cc1, collect2, cpp0, gcc Grep: egrep, grep M4: m4 Make: make Gawk: gawk Sed: sed
Sh−utils: basename, echo, expr, hostname, sleep, uname Texinfo: makeinfo Textutils: cat, tr

# Installer Shadow–4.0.3

```
Estimation du temps de construction:           0.88 SBU
Estimation de l'espace disque requis:  7 Mo
```

## Installation de Shadow Password Suite

Avant d'installer ce package, vous devriez jeter un oeil à l'astuce LFS sur Shadow. Elle examine la façon de rendre son système plus sûr en ce qui concerne les mots de passe, par exemple en activant des mots de passe MD5 plus sécurisés, et comment tirer le meilleur parti de ce package. Cette astuce est disponible à l'adresse suivante: http://hints.linuxfromscratch.org/hints/shadowpasswd_plus.txt.

Installez la Shadow Password Suite en lançant les commandes suivantes:

```
./configure --prefix=/usr \
    --libdir=/usr/lib --enable-shared &&
make &&
make install &&
cp etc/{limits,login.access} /etc &&
sed 's%/var/spool/mail%/var/mail%' \
    etc/login.defs.linux > /etc/login.defs &&
ln -s vipw /usr/bin/vigr &&
rm /bin/vipw &&
mv /bin/sg /usr/bin &&
mv /usr/lib/lib{shadow,misc}.so.0* /lib &&
ln -sf ../../lib/libshadow.so.0 /usr/lib/libshadow.so &&
ln -sf ../../lib/libmisc.so.0 /usr/lib/libmisc.so
```

Sh–utils et Shadow Password Suite installent tous les deux un programme `groups`. Si vous voulez, vous pouvez supprimer le programme installé par la Shadow Password Suite en lançant la commande suivante:

```
rm /bin/groups
```

## Explication des commandes

**`cp limits login.access /etc:`** Ces fichiers n'ont pas été installé pendant l'installation du package, alors nous les copions manuellement car ces fichiers sont utilisés pour configurer les processus d'identification sur le système.

**`sed "s%/var/spool/mail%/var/mail%" login.defs.linux > /etc/login.defs`**: /var/spool/mail est l'ancien emplacement des boites aux lettres utilisateur. L'emplacement utilisé de nos jours est /var/mail.

**`ln -s vipw vigr`**: D'après la page man de vipw, vigr devrait être un lien sur lui. Comme la procédure d'installation de shadow ne crée pas les liens symboliques, nous les créons manuellement.

## Contenu de Shadow

Dernière vérification effectuée auprès de la version 4.0.3.

### Programmes

chage, chfn, chpasswd, chsh, dpasswd, expiry, faillog, gpasswd, groupadd, groupdel, groupmod, grpck,

grpconv, grpunconv, lastlog, login, logoutd, mkpasswd, newgrp, newusers, passwd, pwck, pwconv, pwunconv, sg (lien vers newgrp), su, useradd, userdel, usermod, vigr (lien vers vipw) et vipw

## Descriptions

### chage

chage change le nombre de jours entre chaque changement de mot de passe et la date du dernier changement de mot de passe.

### chfn

chfn change le nom complet de l'utilisateur ainsi que d'autres informations (les coordonnées professionnelles, le numéro de téléphone au bureau et le numéro de téléphone personnel).

### chpasswd

chpasswd lit des paires nom d'utilisateur/mot de passe depuis l'entrée standard pour mettre à jour un groupe d'utilisateurs déjà existants.

### chsh

chsh change le shell de login d'un utilisateur.

### dpasswd

dpasswd ajoute, efface ou met à jour les mots de passe téléphoniques pour les shells de login d'un utilisateur.

### expiry

expiry vérifie et applique les règles d'expiration des mots de passe.

### faillog

faillog met en forme le contenu du journal d'erreur, /var/log/faillog, et tient à jour le compte et les limites de ces erreurs.

### gpasswd

gpasswd administre le fichier /etc/group.

### groupadd

La commande groupadd crée un nouveau groupe en utilisant les valeurs entrées en ligne de commande et les valeurs par défaut du système.

### groupdel

La commande groupdel modifie les fichiers de gestion des comptes du système en effaçant toutes les entrées se référant au groupe spécifié.

### groupmod

La commande groupmod modifie les fichiers de gestion des comptes du système en fonction des entrées en ligne de commande.

### groups

groups affiche le groupe d'appartenance de l'utilisateur.

### grpck

grpck vérifie l'intégrité des informations du système d'authentification.

### grpconv

grpconv convertit des fichiers de groupes normaux en fichiers de groupe de type shadow.

### grpunconv

grpunconv convertit des fichiers de groupes de type shadow en fichiers de groupes normaux.

### lastlog

lastlog met en forme et affiche le contenu du journal des dernières connections, /var/log/lastlog. Le nom de login, le port et l'heure de dernière connection seront affichés.

### login

login permet d'établir une nouvelle session sur le système.

### logoutd

logoutd applique les restrictions de temps de connection et de port de connection définis dans le fichier /etc/porttime.

### mkpasswd

mkpasswd lit un fichier au format spécifié en option et le convertit dans un format de fichier de base de données correspondant.

### newgrp

newgrp permet de changer l'identifiant du groupe en cours d'utilisation.

### newusers

newusers lit un fichier de noms d'utilisateur, efface leur mot de passe et utilise ces informations pour mettre à jour ou pour créer de nouveaux utilisateurs.

### passwd

passwd change les mots de passe des comptes d'utilisateurs et de groupes.

### pwck

pwck vérifie l'intégrité des fichiers de mots de passe.

### pwconv

pwconv convertit un fichier de mots de passe normal en fichier de mots de passe de type shadow.

### pwunconv

pwunconv convertit un fichier de mots de passe de type shadow en fichier de mots de passe normal.

### sg

sg positionne l'identifiant de groupe de l'utilisateur à celui du groupe donné, ou exécute une commande donnée en tant que membre du groupe spécifié.

### useradd

useradd crée un nouvel utilisateur ou met à jour les informations par défaut d'un utilisateur.

### userdel

userdel modifie les fichiers du système de gestion de comptes en effaçant toutes les entrées relative au nom de login spécifié.

### usermod

usermod modifie les fichiers du système de gestion de comptes en fonction des informations données en ligne de commande.

### vipw and vigr

vipw et vigr éditeront les fichiers /etc/passwd et /etc/group, respectivement. Avec l'option –s, ces deux commandes pourront éditer les versions de type shadow de ces mêmes fichiers, /etc/shadow et /etc/gshadow, respectivement.

## Bibliothèques

libmisc.[a,so], libshadow.[a,so]

## Descriptions

### libmisc

Aucune description encore disponible.

### libshadow

libshadow apporte des fonctions communes aux programmes shadow.

## Dépendances d'installation de Shadow

Dernière vérification effectuée auprès de la version 20001016.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh Binutils: ar, as, ld, nm, ranlib Diffutils: cmp Fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm, rmdir Gettext: msgfmt, xgettext Gcc: cc1, collect2, cpp0, gcc Glibc: ldconfig Grep: egrep, grep M4: m4 Make: make Gawk: gawk Net−tools: hostname Sed: sed Sh−utils: basename, echo, expr, sleep, uname Texinfo: makeinfo Textutils: cat, sort, tr, uniq

# Installer Sysklogd–1.4.1

```
Estimation du temps de construction:           0.03 SBU
Estimation de l'espace disque requis:  472 Ko
```

# Installation de Sysklogd

Installez Sysklogd en lançant les commandes suivantes:

```
make &&
make install
```

# Contenu de Sysklogd

Dernière vérification effectuée auprès de la version 1.4.1.

## Programmes

klogd et syslogd

## Descriptions

### klogd

klogd est un démon système qui intercepte et journalise les messages du noyau linux.

### syslogd

syslogd fournit un système de journalisation que la plupart des programmes récents utilise. Chaque message journalisé contient au minimum les champs heure et nom de machine, normalement un champ réservé au nom du programme concerné mais cela dépend des options de journalisation de ce programme.

# Dépendances d'installation de Sysklogd

Dernière vérification effectuée auprès de la version 1.4.1.

Binutils: as, ld, strip Fileutils: install Gcc: cc1, collect2, cpp0, gcc Make: make

# Installer Sysvinit–2.84

```
Estimation du temps de construction:           0.06 SBU
Estimation de l'espace disque requis:  1 Mo
```

# Installation de Sysvinit

Lorsque les niveaux d'exécution sont changés (par exemple lors d'un arrêt système), le programme init va envoyer les signaux TERM et KILL à tous les processus démarrés par init. Mais init écrit un message sur l'écran disant : "sending all processes the TERM signal" et la même chose pour le signal KILL. Cela implique que init envoie ce signal à tous les processus en cours d'exécution, ce qui n'est pas le cas. Pour éviter cette confusion, vous devez changer le fichier init.c afin que la phrase devienne "sending all processes started by init the TERM signal", en exécutant la commande suivante. Si vous ne voulez pas la changer, passez cette commande.

```
cp src/init.c{,.backup} &&
sed 's/Sending processes/Sending processes started by init/g' \
    src/init.c.backup > src/init.c
```

Installez Sysvinit en lançant les commandes suivantes:

```
make -C src &&
make -C src install
```

# Contenu de Sysvinit

Dernière vérification effectuée auprès de la version 2.84.

## Programmes

halt, init, killall5, last, lastb (lien vers last), mesg, pidof (lien vers killall5), poweroff (lien vers halt), reboot (lien vers halt), runlevel, shutdown, sulogin, telinit (lien vers init), utmpdump et wall

## Descriptions

### halt

halt enregistre la demande d'arrêt du système dans le fichier /var/log/wtmp et envoie ensuite au noyau les commandes halt, reboot ou poweroff. Si halt ou reboot sont appelés alors que le système ne se trouve pas dans un état de fonctionnement 0 ou 6, la commande shutdown sera exécutée (avec l'option –h ou –r).

### init

init est le père de tous les processus. Son rôle premier est de créer les processus décrits dans le fichier /etc/inittab. Ce fichier contient en général les entrées pour lancer gettys sur toutes les lignes pouvant réceptionner les demandes de connection des utilisateurs. Il contrôle également les processus requis par un système particulier.

### killall5

killall5 est l'équivalent SystemV de la commande killall. Elle envoie un signal à tous les processus excepté ceux de sa propre session, ainsi elle ne tuera pas le shell qui a lancé le script appelant cette commande.

### last

last recherche dans le fichier /var/log/wtmp (ou dans le fichier désigné après l'option −f) et affiche une liste des utilisateurs qui se sont connectés (et déconnectés) depuis la création de ce fichier.

### lastb

lastb est identique à last, sauf que par défaut il affiche le journal du fichier /var/log/btmp, lequel contient tous les essais infructueux de connection.

### mesg

mesg contrôle l'accès au terminal des utilisateurs par les autres utilisateurs. C'est typiquement utilisé pour permettre ou interdire aux autres utilisateurs d'écrire sur son propre terminal.

### pidof

pidof recherche les identifiants de processus (PIDs) des programmes données.

### poweroff

poweroff est l'équivalent de shutdown −h −p now. Il arrête l'ordinateur et l'éteint (avec un BIOS compatible APM et APM validé dans le noyau).

### reboot

reboot est l'équivalent de shutdown −r now. Il redémarre l'ordinateur.

### runlevel

runlevel lit le fichier système utmp (typiquement /var/run/utmp) pour localiser l'enregistrement de l'état de fonctionnement, puis affiche l'état de fonctionnement précédent et celui en cours sur la sortie standard, séparés par un espace.

### shutdown

shutdown arrête le système en toute sécurité. Tous les utilisateurs connectés sont notifiés de l'arrêt, et les nouvelles connections sont bloquées.

### sulogin

sulogin est employé par init lorsque le système démarre en mode single−user (ceci est déterminé dans le fichier /etc/inittab). init essaie aussi d'exécuter sulogin lorsque le chargeur de boot l'appelle avec l'option −b (par exemple LILO).

### telinit

telinit envoie les signaux appropriés à init pour lui spécifier l'état de fonctionnement dans lequel basculer.

### utmpdump

utmpdump affiche le contenu d'un fichier (habituellement /var/run/utmp) sur la sortie standard dans un format explicite pour l'utilisateur.

### wall

wall envoie un message à tous les utilisateurs connectés qui ont positionné leur permission mesg à oui.

## Dépendances d'installation de Sysvinit

Dernière vérification effectuée auprès de la version 2.84.

Bash: sh Binutils: as, ld Fileutils: chown, cp, install, ln, mknod, rm Gcc: cc, cc1, collect2, cpp0
Make: make Sed: sed

# Installer Tar−1.13

```
Estimation du temps de construction:           0.26 SBU
Estimation de l'espace disque requis:  6 Mo
```

## Installation de Tar

Pour pouvoir utiliser directement les fichiers bzip2 avec tar, utiliser le correctif tar disponible depuis le site FTP LFS. Ce correctif ajoutera à Tar l'option −j, qui fonctionne de manière identique à l'option −z de tar (qui permet d'utiliser directement les fichiers gzip).

Installez le correctif en lançant la commande suivante :

```
patch -Np1 -i ../tar-1.13.patch
```

Installez Tar en lançant les commandes suivantes :

```
./configure --prefix=/usr \
    --libexecdir=/usr/bin --bindir=/bin &&
make &&
make install
```

## Contenu de Tar

Dernière vérification effectuée auprès de la version 1.13.

### Programmes

rmt et tar

### Descriptions

#### rmt

rmt est un programme utilisé par les programmes distants dump et restore pour manipuler un lecteur de bandes magnétiques via une connection de communication entre processus.

#### tar

tar est un programme de sauvegarde destiné à stocker et à extraire des fichiers depuis un format d'archive connu sous le nom de fichier tar.

## Dépendances d'installation de Tar

Dernière vérification effectuée auprès de la version 1.13.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh Binutils: ar, as, ld, ranlib
Diffutils: cmp Fileutils: chmod, install, ls, mv, rm Gettext: msgfmt, xgettext
Gcc: cc, cc1, collect2, cpp0, gcc Glibc: getconf Grep: egrep, fgrep, grep M4: m4 Make: make
Gawk: gawk Net−tools: hostname Patch: patch Sed: sed Sh−utils: basename, echo, expr, sleep, uname
Texinfo: install−info, makeinfo Textutils: cat, tr

# Installer Util–linux–2.11u

```
Estimation du temps de construction:           0.38 SBU
Estimation de l'espace disque requis:  10 Mo
```

# Notes sur la conformité FHS

Le FHS nous recommande d'utiliser /var/lib/hwclock comme emplacement du fichier adjtime, à la place de l'habituel /etc. Pour rendre hwclock, qui fait partie du package util–linux, compatible FHS, exécutez les commandes suivantes:

```
cp hwclock/hwclock.c{,.backup} &&
sed 's%etc/adjtime%var/lib/hwclock/adjtime%' \
    hwclock/hwclock.c.backup > hwclock/hwclock.c &&
mkdir -p /var/lib/hwclock
```

# Installation de Util–linux

Installez Util–linux en lançant les commandes suivantes:

```
./configure &&
make HAVE_SLN=yes &&
make HAVE_SLN=yes install
```

# Explication des commandes

**`HAVE_SLN=yes`**: Nous ne construisons pas ce programme car il est déjà installé par Glibc.

# Contenu de Util–linux

Dernière vérification effectuée auprès de la version 2.11t.

## Programmes

agetty, arch, blockdev, cal, cfdisk, chkdupexe, col, colcrt, colrm, column, ctrlaltdel, cytune, ddate, dmesg, elvtune, fdformat, fdisk, fsck.cramfs, fsck.minix, getopt, hexdump, hwclock, ipcrm, ipcs, isosize, line, logger, look, losetup, mcookie, mkfs, mkfs.bfs, mkfs.cramfs, mkfs.minix, mkswap, more, mount, namei, parse.bash, parse.tcsh, pg, pivot_root, ramsize (lien vers rdev), raw, rdev, readprofile, rename, renice, rev, rootflags (lien vers rdev), script, setfdprm, setsid, setterm, sfdisk, swapoff (lien vers swapon), swapon, test.bash, test.tcsh, tunelp, ul, umount, vidmode (lien vers rdev), whereis et write

## Descriptions

### agetty

agetty ouvre un port tty, affiche une invite pour un nom de login et fait appel à la commande /bin/login.

### arch

arch affiche l'architecture de la machine.

### blockdev

blockdev permet les appels ioctl pour les périphériques de type bloc depuis la ligne de commande.

### cal

cal affiche un simple calendrier.

### cfdisk

cfdisk est un gestionnaire de table des partitions de disque, basé sur la bibliothèque libncurses.

### chkdupexe

chkdupexe cherche les exécutables dupliqués.

### col

col filtre les sauts de ligne inverses sur l'entrée standard.

### colcrt

colcrt filtre la sortie de nroff pour CRT.

### colrm

colrm enlève des colonnes d'un fichier.

### column

column met une liste sous forme de colonnes.

### ctrlaltdel

ctrlaltdel initialise l'action de la combinaison des touches CTRL+ALT+DEL (reset matériel ou logiciel).

### cytune

cytune demande et modifie les interruptions associées au périphériques Cyclades.

### ddate

ddate convertit les dates Grégoriennes en dates Discordiennes.

### dmesg

dmesg examine ou contrôle le buffer circulaire du noyau (messages de démarrage du noyau).

### elvtune

elvtune permet d'affiner les entrées/sorties des périphériques de type bloc.

### fdformat

fdformat lance le formattage bas−niveau d'une disquette.

### fdisk

fdisk est un manipulateur de tables de partition disque.

### fsck.cramfs

Aucune description encore disponible.

### fsck.minix

fsck.minix exécute une vérification logique d'une système de fichier Linux MINIX.

### getopt

getops gère les options de commande de la même façon que la fonction C getopt.

### hexdump

hexdump affiche le fichier spécifié, ou l'entrée standard, dans le format spécifié par l'utilisateur (ascii, décimal, hexadécimal, octal).

### hwclock

hwclock affiche et met à jour l'horloge matérielle (aussi appelée horloge RTC ou BIOS).

### ipcrm

ipcrm supprime la ressource spécifiée.

### ipcs

ipcs fournit des informations sur les ressources IPC.

### isosize

isosize indique la taille d'un système de fichiers iso9660

### line

line copie une ligne (jusqu'à une nouvelle ligne) de l'entrée standard et l'écrit sur la sortie standard.

### logger

logger crée une entrée dans le système de journalisation.

### look

look affiche les lignes commençant par une chaîne de caractères donnée.

### losetup

losetup active et contrôle les périphériques de type loop.

### mcookie

mcookie génère un magic−cookie pour xauth.

### mkfs

mkfs construit un système de fichiers Linux sur un périphérique, habituellement une partition d'un disque dur.

### mkfs.bfs

mkfs.bfs crée un système de fichiers SCO bfs sur un périphérique, habituellement une partition d'un disque dur.

### mkfs.cramfs

Aucune description encore disponible.

### mkfs.minix

mkfs.minix crée un système de fichiers Linux MINIX sur un périphérique, habituellement une partition d'un disque dur.

### mkswap

mkswap crée une zone de swap Linux swap sur un périphérique ou dans un fichier.

### more

more affiche du texte page par page.

### mount

mount monte le système de fichiers d'un périphérique dans un répertoire (point de montage).

### namei

namei parcourt un nom de chemin jusqu'à ce qu'un point terminal soit trouvé.

### parse.bash, parse.tcsh, test.bash, test.tcsh

Ces exemples de scripts, BASH et TCSH, utilisent le programme getopt.

### pg

Aucune description encore disponible.

### pivot_root

pivot_root déplace le système de fichiers root pour le processus courant.

### ramsize

ramsize affiche et initialise une taille de disque mémoire.

### raw

raw est utilisé pour faire correspondre un périphérique caractère Linux vers un périphérique bloc.

### rdev

rdev affiche et positionne l'image du périphérique racine, le périphérique de swap, la taille de disque mémoire, ou le mode vidéo.

### readprofile

readprofile lit les informations de fonctionnement du noyau.

### rename

rename renomme les fichiers.

### renice

renice modifie les priorités des processus en cours.

### rev

rev inverse les lignes d'un fichier.

### rootflags

rootflags affiche et met à jour les informations supplémentaires utilisées lorsque la partition racine est montée.

### script

script crée un fichier de tout ce qui est affiché dans un terminal.

### setfdprm

setfdprm initialise les paramètres des disquettes.

### setsid

setsid exécute les programmes dans une nouvelle session.

### setterm

setterm positionne les attributs d'un terminal.

### sfdisk

sfdisk est un manipulateur de tables de partition disque.

### swapoff

swapoff désactive les périphériques et les fichiers de pagination et de swap.

### swapon

swapon active les périphériques et les fichiers de pagination et de swap.

### tunelp

tunelp initialise de nombreux paramètres pour le périphérique LP.

### ul

ul lit un fichier et transforme les occurrences soulignées en séquences donnant l'aspect souligné pour le terminal utilisé.

### umount

umount démonte un système de fichiers monté.

### vidmode

vidmode affiche et positionne le mode vidéo.

### whereis

whereis localise un exécutable, un fichier source ou une page de manuel de commande.

### write

write envoie un message à un autre utilisateur.

# Dépendances d'installation de Util−linux

Dernière vérification effectuée auprès de la version 2.11n.

Bash: sh Binutils: as, ld Diffutils: cmp Fileutils: chgrp, chmod, cp, install, ln, mv, rm
Gettext: msgfmt, xgettext Gcc: cc, cc1, collect2, cpp, cpp0 Glibc: rpcgen Grep: grep Make: make
Sed: sed Sh−utils: uname, whoami textutils: cat

# Installation de LFS−Bootscripts−1.10

```
Estimation du temps de construction :           0.01 SBU
Estimation de l'espace disque requis :  420 Ko
```

## Installation de LFS−Bootscripts

Nous allons utiliser des scripts de démarrage de style SysV. Nous avons choisi ce style car il est utilisé couramment et que nous les connaissons bien. Si vous préférez essayer autre chose, Marc Heerdink a écrit une astuce LFS à propos des scripts de démarrage de style BSD, qui peut être trouvée sur la page http://hints.linuxfromscratch.org/hints/bsd−init.txt.

Si vous décidez d'utiliser le style BSD, ou tout autre style de scripts, vous pourrez sauter le chapitre 7 lorsque vous y arriverez et passer directement au chapitre 8.

Installez LFS−Bootscripts en lançant les commandes suivantes :

```
cp -a rc.d sysconfig /etc &&
chown -R root:root /etc/rc.d /etc/sysconfig
```

## Contenu de LFS−bootscripts

Dernière vérification effectuée auprès de la version 1.10.

### Scripts

checkfs, cleanfs, functions, halt, ifdown, ifup, loadkeys, localnet, mountfs, mountproc, network, rc, reboot, sendsignals, setclock, swap, sysklogd et template

### Descriptions

#### checkfs

Le script checkfs vérifie les systèmes de fichiers un peu avant qu'ils ne soient montés à l'exception des systèmes de fichiers journalisés ou réseau).

#### cleanfs

Le script cleanfs supprime tous les fichiers qui ne devraient pas être conservés entre les redémarrages, fichiers tels que /var/run/*, /var/lock/*. Il recrée /var/run/utmp et supprime les éventuels fichiers /etc/nologin, /fastboot et /forcefsck.

#### functions

Le script functions contient les fonctions partagées entre les différents scripts, telles que la vérification d'erreur, la vérification de l'état, etc...

#### halt

Le script halt arrête le système.

### ifdown, ifup

Les scripts ifdown et ifup assistent le script network avec les périphériques réseaux.

### loadkeys

Le script loadkeys charge le plan de clavier, correspondant à votre clavier.

### localnet

Le script localnet initialise le nom de l'hôte ainsi que le périphérique loopback.

### mountfs

Le script mountfs monte tous les systèmes de fichiers qui ne sont pas indiqués noauto et qui ne sont pas réseau.

### mountproc

Le script mountproc est utilisé pour monter le système de fichiers proc.

### network

Le script network met en place les interfaces réseau (telles que les cartes réseau) et initialise la passerelle par défaut si nécessaire.

### rc

Le script rc est le script maître de contrôle du niveau d'exécution, responsable pour lancer tous les autres scripts un par un dans une séquence spécifiée.

### reboot

Le script reboot relance le système.

### sendsignals

Le script sendsignals s'assure que chaque processus est bien arrêté avant que le système ne s'arrête ou ne redémarre la machine.

### setclock

Le script setclock réinitialise l'horloge du noyau à l'heure locale au cas où l'horloge matérielle n'est pas positionnée sur l'heure GMT.

### swap

Le script swap active et désactive les fichiers et partitions de swap.

### sysklogd

Le script sysklogd lance et arrête les démons système et kernel de log.

### template

Le script template est un modèle que vous pouvez utiliser pour créer vos propres scripts de démarrage pour d'autres démons.

## Dépendances d'installation de Bootscripts

Dernière vérification effectuée auprès de la version 1.10.

Fileutils: chown, cp

# Configurer les logiciels essentiels

Maintenant que tous les logiciels sont installés, tout ce dont nous avons besoin pour que les programmes tournent correctement est de créer leurs fichiers de configuration.

## Configurer Vim

Par défault, vim fonctionne en mode compatible vi. Certaines personnes l'aiment comme cela, mais nous avons une forte préférence pour faire tourner Vim en mode Vim (autrement nous n'aurions pas inclus vim dans ce livre mais l'original vi). Créez `/root/.vimrc` en exécutant ce qui suit:

```
cat > /root/.vimrc << "EOF"
" Debut de /root/.vimrc

set nocompatible
set bs=2

" Fin de /root/.vimrc
EOF
```

## Configurer Glibc

Nous avons besoin de créer le fichier /etc/nsswitch.conf. Bien que glibc puisse fournir un paramétrage par défault lorsque ce fichier manque ou est corrompu, celle−ci ne fonctionne pas correctement avec les réseaux, que nous traiterons dans un prochain chapitre. Notre zone horaire a également besoin d'être configurée

Créez un nouveau fichier`/etc/nsswitch.conf` en exécutant ce qui suit:

```
cat > /etc/nsswitch.conf << "EOF"
# Debut de /etc/nsswitch.conf

passwd: files
group: files
shadow: files

publickey: files

hosts: files dns
networks: files

protocols: db files
services: db files
ethers: db files
rpc: db files

netgroup: db files

# Fin de /etc/nsswitch.conf
EOF
```

Le script **`tzselect`** doit être lancé et vous devez répondre aux questions correspondant à votre zone horaire. Lorsque vous l'aurez fait, le script vous donnera l'adresse du fichier de zone horaire nécessaire.

Créez le lien symbolique `/etc/localtime` en lançant:

```
ln -sf ../usr/share/zoneinfo/<tzselect's output> /etc/localtime
```

Les sorties de tzselect peuvent être du style *EST5EDT* ou *Canada/Eastern*.

Le lien symbolique que vous avez créé serait:

```
ln -sf ../usr/share/zoneinfo/EST5EDT /etc/localtime
```

Ou:

```
ln -sf ../usr/share/zoneinfo/Canada/Eastern /etc/localtime
```

# Configurer le chargeur dynamique

Par défaut, le chargeur dynamique (`/lib/ld-linux.so.2`) cherche à travers `/lib` et `/usr/lib` pour des bibliothèques dynamiques dont certains programmes ont besoin lorsque vous les lancez. Néanmoins, si certaines bibliothèques appartiennent à des répertoires autres que `/lib` et `/usr/lib`, vous devez les ajouter dans le fichier `/etc/ld.so.conf` de façon à ce que le chargeur dynamique puisse les trouver. Deux répertoires contenant habituellement des bibliothèques supplémentaires sont `/usr/local/lib` et `/opt/lib`, donc nous ajoutons ces répertoires au chemin de recherche du chargeur dynamique.

Créez un nouveau fichier `/etc/ld.so.conf` en exécutant ce qui suit:

```
cat > /etc/ld.so.conf << "EOF"
# Begin /etc/ld.so.conf

/usr/local/lib
/opt/lib

# End /etc/ld.so.conf
EOF
```

# Configurer Sysklogd

Créez un nouveau fichier `/etc/syslog.conf` en exécutant ce qui suit:

```
cat > /etc/syslog.conf << "EOF"
# Debut de /etc/syslog.conf

auth,authpriv.* -/var/log/auth.log
*.*;auth,authpriv.none -/var/log/sys.log
daemon.* -/var/log/daemon.log
kern.* -/var/log/kern.log
mail.* -/var/log/mail.log
user.* -/var/log/user.log
*.emerg *

# Fin de /etc/syslog.conf
EOF
```

# Configurer Shadow Password Suite

Ce package contient des utilitaires pour modifier les mots de passe des utilisateurs, ajouter ou supprimer des utilisateurs et groupes, plus encore. Nous n'allons pas expliquer ce que 'password shadowing' veut dire. Un explication complète peut être lu dans le fichier doc/HOWTO à l'intérieur du répertoire des sources. Vous devez avoir une chose à l'esprit, si vous décidez d'activer le support de shadow. Les programmes qui ont besoin de vérifier des mots de passes (comme xdm, les démons ftp, pop3, etc) ont besoin d'être compatible

avec shadow, c'est–à–dire qu'ils doivent être capable de travailler avec des mots de passe shadow.

Pour activer les mots de passe shadow, lancez la commande suivante:

```
/usr/sbin/pwconv
```

# Configurer Sysvinit

Créez un nouveau fichier `/etc/inittab` en exécutant ce qui suit :

```
cat > /etc/inittab << "EOF"
# Begin /etc/inittab

id:3:initdefault:

si::sysinit:/etc/rc.d/init.d/rc sysinit

l0:0:wait:/etc/rc.d/init.d/rc 0
l1:S1:wait:/etc/rc.d/init.d/rc 1
l2:2:wait:/etc/rc.d/init.d/rc 2
l3:3:wait:/etc/rc.d/init.d/rc 3
l4:4:wait:/etc/rc.d/init.d/rc 4
l5:5:wait:/etc/rc.d/init.d/rc 5
l6:6:wait:/etc/rc.d/init.d/rc 6

ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now

su:S016:once:/sbin/sulogin

1:2345:respawn:/sbin/agetty tty1 9600
2:2345:respawn:/sbin/agetty tty2 9600
3:2345:respawn:/sbin/agetty tty3 9600
4:2345:respawn:/sbin/agetty tty4 9600
5:2345:respawn:/sbin/agetty tty5 9600
6:2345:respawn:/sbin/agetty tty6 9600

# End /etc/inittab
EOF
```

# Configuration du clavier

Rien n'est plus ennuyant que d'utiliser Linux avec un mauvais plan de codage pour votre clavier. Si vous disposez du clavier US par défaut, vous pouvez passer cette section. Le fichier US est utilisé par défaut.

Pour indiquer le fichier de plan de codage par défaut, créez le lien symbolique `/usr/share/kbd/keymaps/defkeymap.map.gz` en lançant les commandes suivantes:

```
ln -s <path/to/keymap> /usr/share/kbd/keymaps/defkeymap.map.gz
```

Remplacez <path/to/keymap> avec le fichier pour votre clavier. Par exemple, si vous utilisez un clavier hollandais, lancez la commande:

```
ln -s i386/qwerty/nl.map.gz /usr/share/kbd/keymaps/defkeymap.map.gz
```

Une deuxième option pour configurer votre plan de codage de votre clavier est de compiler celui–ci directement dans le noyau. Ceci vous assurera que votre clavier fonctionnera toujours correctement, même si vous avez démarré en mode maintenance (en passant 'init=/bin/sh' au noyau), mode qui ne lance pas le script de démarrage qui configure votre clavier.

Lancer la commande suivante pour corriger le bon plan de codage dans les sources du noyau. Vous devrez répéter cette commande à chaque fois vous déballerez un nouveau noyau:

```
loadkeys -m /usr/share/kbd/keymaps/defkeymap.map.gz > \
    /usr/src/linux/drivers/char/defkeymap.c
```

# Créer les fichiers /var/run/utmp, /var/log/wtmp et /var/log/btmp

Les programmes tels que login, shutdown, uptime et les autres lisent depuis et écrivent vers les fichiers /var/run/utmp, /var/log/btmp et /var/log/wtmp. Ces derniers contiennent des informations sur les connections en cours. Ils contiennent aussi des informations relatives aux dates du dernier démarrage/arrêt de l'ordinateur ainsi que les tentatives infructueuses de login.

Créez ces fichiers avec les permissions appropriées en lançant les commandes suivantes :

```
touch /var/run/utmp /var/log/{btmp,lastlog,wtmp} &&
chmod 644 /var/run/utmp /var/log/{btmp,lastlog,wtmp}
```

# Créer un mot de passe pour root

Choisissez un mot de passe pour l'utilisateur root et créez–le en lançant la commande suivante:

```
passwd root
```

# Chapitre 7. Mise en place des scripts de démarrage

# Introduction

Ce chapitre va permettre la configuration des scripts de démarrage que vous avez installé lors du chapitre 6. La plupart de ces scripts fonctionneront sans modification, mais un petit nombre vont nécessiter le paramétrage de fichiers de configurations supplémentaires, car ils ont affaire à des informations dépendant du matériel.

# Comment fonctionne le processus de démarrage utilisant ces scripts?

Linux utilise un système de démarrage nommé SysVinit. Il est basé sur le concept de *niveaux d'exécution*. Il peut être extrêmement différent d'un système à l'autre, on ne peut donc pas supposer que si cela a fonctionné avec <tel distribution>, cela devrait aussi fonctionner de la même manière avec LFS. LFS fait les choses à sa manière, mais respecte généralement les standards établis.

SysVinit (que nous nommerons dorénavant *init*) se base pour fonctionner sur un système de niveaux d'exécution. Ils sont au nombre de 7 (de 0 à 6) (en réalité, il y a plus de niveaux d'exécution que cela, mais ils sont réservés à des cas spéciaux et ne sont généralement pas utilisés. La page man de init décrit ces détails), et chacun d'eux correspond à ce que l'ordinateur est supposé faire lorsqu'il démarre. Le niveau d'exécution par défaut est le niveau 3. Voici la description des différents niveaux d'exécution tels qu'ils sont fréquemment implémentés:

0: arrête l'ordinateur 1: mode mono−utilisateur 2: mode multi−utilisateur sans réseau
3: mode multi−utilisateur avec le réseau
4: réservé à la personnalisation, sinon identique au niveau 3
5: identique au 4, utilisé généralement pour une connexion graphique (comme
l'xdm de X ou kdm de KDE) 6: redémarre l'ordinateur

L'instruction utilisée pour changer de niveau d'exécution est **`init <niveau d'exécution>`** où <niveau d'exécution> est le niveau d'exécution désiré. Par exemple, pour redémarrer l'ordinateur, un utilisateur lancera l'instruction init 6. L'instruction reboot est simplement un alias, tout comme l'instruction halt est un alias d'init 0.

Il existe un certain nombre de répertoires sous /etc qui ressemblent à rc?.d où ? est le niveau d'exécution et rcsysinit.d qui contient un certain nombre de liens symboliques. Certains commencent par un K, les autres par un S et tous ont deux chiffres après la lettre initiale. Le K signifie arrêter (kill) un service, et le S (start) en démarrer un. Les chiffres déterminent l'ordre d'exécution des scripts, de 00 à 99 ; plus un nombre est petit, plus tôt il sera exécuté. Lorsque init passe à un autre niveau d'exécution, les services appropriés sont arrêtés et d'autres sont démarrés.

Les véritables scripts sont dans /etc/rc.d/init.d. Ils font tout le travail et les liens symboliques pointent vers eux. Les liens de d'arrêt et de démarrage pointent vers le même script dans /etc/rc.d/init.d. Ceci est dû au fait que le script peut être appellé avec des paramètres différents tels que start, stop, restart, reload, status. Quand un lien K est rencontré, le script approprié est exécuté avec l'argument stop. Quand un lien S est rencontré, le script idoine est exécuté avec l'argument start.

Il existe une exception. les liens commençant par un S dans les répertoires rc0.d et rc6.d ne feront rien démarrer. Ils seront appelés avec le paramètre *stop* pour arrêter quelque chose. La logique derrière cela est que lorsque l'on est sur le point de redémarrer ou d'arrêter le système, on ne désire rien démarrer, seulement arrêter le système.

Voici la description de ce que ces arguments font réaliser aux scripts :

- *start*: le service est démarré.
-

*stop*: le service est arrêté.

- *restart*: le service est arrêté puis démarré à nouveau.

- *reload*: la configuration du service est mise à jour. Ceci est utilisé après la modification du fichier de configuration d'un service, lorsque le service n'a pas besoin d'être redémarré.

- *status*: indique si le service est lancé et avec quel PID.

Modifiez librement la manière dont le système de démarrage fonctionne (après tout, il s'agit de votre système LFS). Les fichiers donnés ici sont simplement un exemple de la manière de faire tout cela proprement (quoi qu'il en soit, vous pouvez détester ce que nous considérons comme propre).

# Configuration du script setclock

Le script setclock lit l'heure à partir de l'horloge matérielle (aussi connu en tant qu'horloge BIOS ou CMOS) et soit convertit cette heure en temps local en utilisant le fichier `/etc/localtime` (si l'horloge matérielle est réglée sur GMT), soit ne le fait pas (si votre horloge matérielle est déjà réglé sur l'heure locale). Il n'existe pas de moyens pour détecter automatiquement si votre horloge est réglée sur l'heure GMT ou locale, donc nous avons besoin de le configurer nous−même.

Changer la valeur de la variable *UTC* ci−dessous en indiquant *0* (zéro) si votre horloge matérielle n'utilise pas l'heure GMT.

Créez un nouveau fichier `/etc/sysconfig/clock` en lançant la commande suivante:

```
cat > /etc/sysconfig/clock << "EOF"
# Begin /etc/sysconfig/clock

UTC=1

# End /etc/sysconfig/clock
EOF
```

Maintenant, vous voudrez peut−être jeter un oeil sur cette excellente astuce expliquant comment on gère le temps avec LFS: http://hints.linuxfromscratch.org/hints/time.txt. Il explique des points concernant les fuseaux horaires UTC et la variable d'environnement TZ.

# Ai–je besoin du script loadkeys?

Si vous avez décidé de compiler votre plan de codage du clavier directement dans le noyau à la fin du chapitre 6, alors vous n'avez pas besoin de lancer le script loadkeys car le noyau a déjà initialisé le plan de codage pour vous. Vous pouvez toujours le lancer si vous voulez, cela ne posera pas de problème. Cela peut même vous être avantageux de le garder au cas où vous posséderiez différents noyaux et que vous ne vous rappelliez pas, ou que vous ne vouliez pas compiler le plan de codage dans chaque noyau installé.

Si vous décidez que vous n'en avez pas besoin ou que vous ne voulez pas utiliser le script loadkeys, supprimer le lien symbolique `/etc/rc.d/rcsysinit.d/S70loadkeys` .

# Configuration du script sysklogd

Le script `sysklogd` invoque le programme **`syslogd`** avec l'option *−m 0*. Cette option désactive la marque périodique que syslogd écrit dans les fichiers de log toutes les vingt minutes par défaut. Si vous préférez l'activer, éditez le script `sysklogd` et faites les changements adéquats. Voir la page de manuel **`man syslogd`** pour plus d'informations.

# Configurer le script localnet

Une partie de ce script configure le nom du système. Ce nom doit être indiqué dans le fichier /etc/sysconfig/network.

Créez le fichier /etc/sysconfig/network et entrez le nom du système en lançant:

```
echo "HOSTNAME=lfs" > /etc/sysconfig/network
```

<< lfs>> doit être remplacé par le nom de l'ordinateur. Vous ne devez pas entrer le FQDN (Fully Qualified Domain Name, nom de domaine pleinement qualifié) ici. Cette information sera rentrée dans le fichier `/etc/hosts` un peu plus tard.

# Création du fichier /etc/hosts

Si une carte réseau doit être configurée, vous devez choisir une adresse IP, le nom de domaine pleinement qualifié et les alias possibles à déclarer dans le fichier /etc/hosts. La syntaxe est la suivante:

```
<adresse IP> mon-hôte.mon-domaine.org alias
```

Vous devez vous assurer que l'adresse IP se trouve dans la plage d'adresses réservée aux réseaux privés. Les plages valides sont:

```
        Classes Réseaux
        A     10.0.0.0
        B     172.16.0.0 à 172.31.0.0
        C     192.168.0.0 à 192.168.255.0
```

Une adresse IP valide pourrait être 192.168.1.1. Un nom de domaine pleinement qualifié pour cette adresse IP pourrait être www.linuxfromscratch.org.

Si vous ne possédez pas de carte réseau, vous devez néanmoins déclarer un nom de domaine pleinement qualifié. Cela est nécessaire à certains programmes pour fonctionner correctement.

Si aucune carte réseau ne doit être configurée, créez le fichier `/etc/hosts` en lançant la commande:

```
cat > /etc/hosts << "EOF"
# Début de /etc/hosts (version sans carte réseau)

127.0.0.1 www.mon-domaine.com <nom d'hôte> localhost

# Fin de /etc/hosts (version sans carte réseau)
EOF
```

Si une carte réseau doit être configurée, créez le fichier `/etc/hosts` en lançant la commande:

```
cat > /etc/hosts << "EOF"
# Début de /etc/hosts (version avec carte réseau)

127.0.0.1 localhost.localdomain localhost
192.168.1.1 www.mon-domaine.org <nom d'hôte>

# Fin de /etc/hosts (version avec carte réseau)
EOF
```

Bien évidemment, les valeurs 192.168.1.1 et www.mon−domaine.org doivent être changées selon vos souhaits (ou selon les valeurs données par l'administrateur système/réseau si cette machine doit être connectée à un réseau existant).

# Configuration du script network

Cette section n'est utile que si vous souhaitez configurer une carte réseau.

Si vous ne possédez aucune carte réseau, vous pouvez ne pas créer les fichiers de configuration en rapport avec celles–ci. Si c'est le cas, vous devez supprimer les liens symboliques de `network` de tous les répertoires des niveaux d'exécution (`/etc/rc.d/rc*.d`)

## Configuration de la passerelle par défaut

Si vous êtes sur un réseau, vous pouvez avoir besoin de configurer la passerelle par défaut de cette machine. Ceci se fait en ajoutant les bonnes valeurs au fichier /etc/sysconfig/network, en lançant les commandes suivantes:

```
cat >> /etc/sysconfig/network << "EOF"
GATEWAY=192.168.1.2
GATEWAY_IF=eth0
EOF
```

Les valeurs pour GATEWAY et GATEWAY_IF doivent être changées pour correspondre à la configuration de votre réseau. GATEWAY contient l'adresse IP de la passerelle par défaut, et GATEWAY_IF contient l'interface réseau par laquelle la passerelle par défaut peut être jointe.

## Création des fichiers de configuration d'interfaces

Quelles interfaces doivent être activées ou non dépendent des fichiers du répertoire /etc/sysconfig/network–devices. Ce répertoire contient des fichiers de la forme ifconfig.xyz, où xyz est le nom de l'interface réseau (comme eth0 ou eth0:1).

Si vous décidez de renommer ou changer le répertoire /etc/sysconfig/network–devices, assurez–vous de modifier le fichier /etc/sysconfig/rc et de mettre à jour la variable network_devices en lui indiquant le nouveau chemin.

Maintenant, de nouveaux fichiers sont créés dans ce répertoire avec les informations suivantes. La commande suivante crée un fichier d'exemple ifconfig.eth0:

```
cat > /etc/sysconfig/network-devices/ifconfig.eth0 << "EOF"
ONBOOT=yes
IP=192.168.1.1
NETMASK=255.255.255.0
BROADCAST=192.168.1.255
EOF
```

Bien sûr, les valeurs de ces variables doivent être changé dans chaque fichier pour correspondre à votre propre configuration. Si la variable ONBOOT est mise à yes, le script network l'activera lors du démarrage du système. Pour une quelconque autre valeur, cette interface sera ignorée par le script network et ne sera donc pas activée.

# Chapitre 8. Rendre le système LFS démarrable

# Introduction

Ce chapitre permettra de rendre LFS démarrable. Ce chapitre traite de la création d'un nouveau fichier fstab, de la construction d'un nouveau noyau pour le nouveau système LFS et de l'ajout d'entrées propres à LILO afin que le système LFS puisse être sélectionné lors du démarrage de LILO.

# Créer le fichier /etc/fstab

On utilise le fichier /etc/fstab afin que certains programmes puissent déterminer où certaines partitions sont supposées être montées par défaut. Créez un nouveau fichier `/etc/fstab` en lançant cette commande:

```
cat > /etc/fstab << "EOF"
# Début de /etc/fstab

# emplacement du système de fichier     point de montage        type de système options

/dev/*périphérique partition-LFS*       /                       *fs-type*       defaults 1 1
/dev/*périphérique partition-swap*      swap                    swap            pri=1 0 0
proc                                    /proc                   proc            defaults 0 0

# Fin de /etc/fstab
EOF
```

**`*périphérique partition-LFS*`**, **`*périphérique partition-swap*`** et **`*fs-type*`** doivent être remplacés par les valeurs adéquates (/dev/hda2, /dev/hda5 et reiserfs par exemple).

Lorsqu'on ajoute une partition reiserfs, le **`1 1`** à la fin de la ligne doit être remplacé par **`0 0`**.

Pour plus d'informations sur les différents champs présents dans le fichier fstab, voir **`man 5 fstab`**.

Il y a d'autres lignes que vous pourriez ajouter à votre fichier fstab. L'exemple suivant est la ligne que vous devriez avoir si vous utilisez devpts:

```
devpts          /dev/pts        devpts    gid=4,mode=620  0 0
```

Un autre exemple est la ligne à utiliser pour employer des périphériques USB:

```
usbdevfs                /proc/bus/usb   usbdevfs defaults        0 0
```

Ces deux options ne fonctionneront que si vous avez compilé le support adéquat dans votre noyau.

# Installer Linux–2.4.19

```
Temps de construction estimé:           Toutes les options par défaut: 4.20 SBU
Espace disque nécessaire estimé:  Toutes les options par défaut: 181 Mo
```

Construire un noyau implique quelques étapes: le configurer et le compiler. Il y a plusieurs façons de configurer un noyau. Si la manière décrite dans ce livre ne vous convient pas, lisez le fichier `README` qui est inclus dans l'arborescence des sources du noyau, et cherchez quelles sont les autres options.

Une chose que vous pourriez faire, est de prendre le fichier `.config` des sources du noyau de votre distribution hôte et de le copier dans `$LFS/usr/src/linux-2.4.19`. De cette façon, vous n'aurez pas à configurer la totalité du noyau depuis le début et pourrez utiliser vos valeurs actuelles. Si vous choisissez de le faire, commencez par lancer la commande **`make mrproper`**, ensuite copiez le fichier `.config` par dessus, et enfin lancez **`make menuconfig`** (**`make oldconfig`** peut être préférable dans certaines situations. Voir le fichier `README` pour plus de détails sur l'utilisation du **`make oldconfig`**).

Si vous avez l'intention d'utiliser des modules pour le noyau, vous aurez besoin d'un fichier `/etc/modules.conf`. Les informations pertinentes pour le noyau et pour la configuration du noyau en général peuvent être trouvées dans la documentation du noyau, qui est stockée dans `/usr/src/linux-2.4.19/Documentation`. La page de manuel de modules.conf et le HOWTO concernant le noyau, disponible à l'adresse http://www.tldp.org/HOWTO/Kernel–HOWTO.html, peuvent aussi vous être utile.

Les commandes suivantes sont utilisées pour construire le noyau:

```
make mrproper &&
make menuconfig &&
make dep &&
make bzImage &&
make modules &&
make modules_install &&
cp arch/i386/boot/bzImage /boot/lfskernel &&
cp System.map /boot
```

Note: le chemin arch/i386/boot/bzImage peut varier suivant la plateforme.

# Dépendances d'installation de Linux

Dernière vérification effectuée auprès de la version 2.4.17.

Bash: sh Binutils: ar, as, ld, nm, objcopy Fileutils: cp, ln, mkdir, mv, rm, touch
Findutils: find, xargs Gcc: cc1, collect2, cpp0, gcc Grep: grep Gzip: gzip Make: make Gawk: awk
Modutils: depmod, genksyms Net–tools: dnsdomainname, hostname Sed: sed
Sh–utils: basename, date, expr, pwd, stty, uname, whoami, yes Textutils: cat, md5sum, sort, tail

# Rendre son système LFS démarrable

Afin de pouvoir démarrer son système LFS, nous avons besoin de mettre à jour notre chargeur de démarrage. Nous avons supposé que votre système hôte utilise LILO (car c'est le chargeur le plus commun à l'heure actuelle).

Nous n'allons pas lancer le programme LILO à l'intérieur du chroot. Lancer lilo à l'intérieur du chroot peut avoir comme conséquence de rendre votre MBR inutile et vous auriez besoin d'une disquette de démarrage pour être capable de lancer tout système Linux (le système de base ainsi que le système LFS).

Premièrement nous sortirons du chroot et copirons le fichier lfskernel sur le système de base:

```
logout
cp $LFS/boot/lfskernel /boot
```

L'étape suivante est d'ajouter l'entrée dans /etc/lilo.conf ce qui nous permettra de choisir LFS lorsque nous démarrerons l'ordinateur:

```
cat >> /etc/lilo.conf << "EOF"
image=/boot/lfskernel
        label=lfs
        root=<partition>
        read-only
EOF
```

<partition> doit être remplacé par l'emplacement de la partition LFS.

Notez également que si vous utilisez reiserfs comme partition racine, la ligne **`read-only`** devra être changée en **`read-write`**.

Maintenant, mettez à jour le chargeur en lancant:

```
/sbin/lilo -v
```

La dernière étape est la synchronisation des fichiers de configuration du système de base avec ceux du système LFS:

```
cp /etc/lilo.conf $LFS/etc &&
cp $(grep "image.*=" /etc/lilo.conf | cut -f 2 -d "=") $LFS/boot
```

# Chapitre 9. La fin

# La fin

Bien joué! Vous avez terminé d'installer votre système LFS. Cela vous a peut−être pris du temps, mais c'est une bonne chose que de l'avoir fait. Nous vous souhaitons de bien vous amuser avec votre nouveau système Linux rutilant.

Maintenant c'est le bon moment pour nettoyer les binaires de tous les symboles de débogage sur votre système LFS. Si vous n'êtes pas un programmeur et ne prévoyez pas de déboguer vos logiciels, alors vous serez certainement content de savoir que l'on peut gagner quelques dizaines de mégas en enlevant les symboles de débogage. Ce processus n'a pas d'autre inconvénient que de vous empêcher de déboguer votre logiciel à l'avenir, ce qui n'a pas d'importance si vous ne saviez pas comment faire.

Précision: 98% des personnes qui utilisent la commande ci−dessous n'ont jamais eu de problème. Mais faites une sauvegarde de votre système LFS avant de lancer cette commande. Il y a une chance infime que cela se retourne contre vous et rende votre système inutilisable (essentiellement en détruisant les modules du noyau et les bibliothèques dynamiques partagées). Cela est plus souvent dû à des erreurs de frappes qu'à des problèmes avec les commandes utilisées.

Cela étant dit, l'option −−strip−debug que nous utilisons est plutôt sans dommage dans des circonstances normales. Cela ne débarrasse pas les fichiers de quoique ce soit de vital. Il n'est sûr d'utiliser −−strip−all que sur des programmes classiques (ne pas utiliser sur des bibliothèques − elles seraient détruites) mais pas aussi sûr et l'espace gagné n'est pas plus grand. Mais si vous êtes limité en espace disque, quelques octets de plus peuvent aider, alors décidez vous−même. Référez vous, s'il vous plait, à la page man pour d'autres options de strip que vous pourriez utiliser. L'idée générale est de ne pas lancer strip sur des bibliothèques (autre que −−strip−debug) pour rester su côté sûr.

```
find $LFS/{,usr/,usr/local/}{bin,sbin,lib} -type f \
   -exec /usr/bin/strip --strip-debug '{}' ';'
```

Cela peut être une bonne idée de créer le fichier $LFS/etc/lfs. Avec ce fichier, il devient très facile pour vous (et pour nous si vous venez à nous demander de l'aide sur quelque chose) de trouver quelle version de LFS vous avez d'installée sur votre système. Créez le fichier $LFS/etc/lfs en lançant la commande suivante:

```
echo 4.0 > $LFS/etc/lfs
```

# Enregistrez–vous

Vous voulez être enregistré comme utilisateur de LFS maintenant que vous avez terminé le livre ? Allez directement à http://linuxfromscratch.org/cgi–bin/lfscounter.cgi et enregistrez–vous comme utilisateur LFS en entrant votre nom et la première version de LFS que vous avez utilisée.

Replongeons nous maintenant dans LFS...

# Redémarrer le système

Maintenant que tous les logiciels ont été installés, les scripts de démarrage créés, il est temps de redémarrer l'ordinateur. Avant de redémarrer, démontons $LFS/proc et la partition LFS elle−même en exécutant:

```
umount $LFS/proc &&
umount $LFS
```

Si vous décidez de créer de multiples partitions, vous avez besoin de démonter les autres partitions avant de démonter $LFS, comme ceci:

```
umount $LFS/proc &&
umount $LFS/usr &&
umount $LFS/home &&
umount $LFS
```

Et vous pouvez redémarrer votre système en lançant quelque chose comme:

```
/sbin/shutdown -r now
```

Au prompt LILO: assurez−vous que vous lui dites de démarrer *lfs* et pas l'option par défaut qui redémarrera votre système hôte.

Après avoir redémarré, votre système LFS est prêt à l'emploi et vous pouvez commencer à ajouter vos propres logiciels.

Une dernière chose que vous voudriez peut−être faite est de lancer lilo, maintenant que vous avez démarré avec LFS. Ainsi vous placerez la version LFS de LILO dans le MBR plutôt que celle qu'il y a sur votre système hôte. En fonction de l'âge de la distribution de votre hôte, la version LFS peut disposer de fonctionnalités plus avancées que vous pourriez ou devriez utiliser.

Dans tous les cas, lancer ce qui suit pour activer la version LFS de lilo:

```
/sbin/lilo
```

Vous pouvez maintenent supprimer le répertoire static. Si vous pensez avoir à refaire le chapitre 5, alors vous pouvez souhaiter faire une copie de ce répertoire avant de le supprimer. Pour supprimer le répertoire static, lancez la commande suivante:

```
rm -rf /static
```

Maintenant que vous avez terminé l'installation de votre système LFS, vous pouvez vous demander comment installer d'autres logiciels, comme un navigateur web. Commencez par le projet Beyond Linux From Scratch qui est disponible à l'adresse http://beyond.linuxfromscratch.org/. Les astuces LFS peuvent aussi être utiles et sont disponibles à l'adresse http://hints.linuxfromscratch.org/hints.shtml. De la même manière, si vous n'êtes pas seulement un nouveau sur LFS, mais aussi avec Linux en général, vous pouvez trouver l'astuce newbie très intéressante. Elle est disponible sur http://hints.linuxfromscratch.org/hints/newbie.txt.

N'oubliez pas qu'il y a plusieurs listes de diffusions LFS aux quelles vous pouvez vous abonner en cas de besoin d'aide, conseils, etc. Pour plus d'informations, se référer à Chapter 1 − Listes de diffusion et archives.

Nous vous remercions encore d'avoir utilisé le livre LFS et espérons que vous l'avez trouvé utile et qu'il vous a fait gagner du temps.

# III. Troisième Partie – Annexes

***Table des matières***

*A. Descriptions des packages et dépendances*
*B. Ressources*

# Annexe A. Descriptions des packages et dépendances

# Introduction

Dans cette annexe, les aspects suivants des packages installés dans ce livre sont décrits:

- le site officiel de téléchargement pour le package,
- ce que le package contient,
- ce que chaque programme du package réalise,
- ce que chaque package a besoin pour compiler.

Beaucoup d'informations sur ces packages (principalement les descriptions) proviennent des pages man, contenues dans ces packages. Nous n'incluons pas la page man entière, mais simplement les éléments principaux pour permettre la compréhension de l'utilité du package. Pour disposer de tous les détails sur un programme, rérérez−vous à sa page man ou à sa page info.

Certains packages sont documentés plus en profondeur que d'autres, parce que nous en savons plus sur ceux−là. Si quelque chose doit être ajoutés aux descriptions, n'hésitez surtout pas à nous envoyer un email à la mailing list. Nous avons l'intention que cette liste contienne une description en profondeur de chaque package installé, mais nous ne pouvons pas le faire sans aide.

Remarquez aussi qu'actuellement, seul ce que fait un package est décrit. Pourquoi il doit être installé n'est pas encore couvert. Ceci pourra être ajouté plus tard.

Toutes les dépendances de chaque package installé avec ce livre sont décrites. Seront inclus les programmes de chaque package nécessaires pour compiler le package à installer.

Ce ne sont pas des dépendances lors de l'exécution, c'est−à−dire ceux dont vous avez besoin pour les lancer. Uniquement ceux nécessaire pour la compilation.

La liste de dépendances peut ne pas être à jour de temps en temps à cause de la version courant du package. Vérifier les dépendances demande un travail important, donc il peut y avaoir du temps avec une mise à jour. Mais souvent avec des mises à jour mineures d'un package, les dépendances d'installation changent peu, donc elles seront correctes dans la plupart des cas. Quand nous mettons à jour vers une nouvelle version majeure, nous nous assurons que les dépendances ont aussi été vérifié.

# Autoconf

## Site officiel de téléchargement

Autoconf (2.53):  ftp://ftp.gnu.org/gnu/autoconf/

## Contenu de Autoconf

Dernière vérification effectuée auprès de la version 2.53.

### Programmes

autoconf, autoheader, autom4te, autoreconf, autoscan, autoupdate et ifnames

### Descriptions

#### autoconf

autoconf est un outil destiné à produire des scripts shell qui configurent automatiquement les packages de code source logiciel pour les adapter à de nombreuses sortes de systèmes de type Unix. Les scripts de configuration produits par autoconf sont indépendants d'autoconf lorsqu'ils sont exécutés, aussi leurs utilisateurs n'ont pas besoin d'avoir Autoconf installé.

#### autoheader

Le programme autoheader permet de créer un fichier modèle des déclarations C #define, pour l'usage de configure.

#### autom4te

autom4te lance M4 de GNU sur des fichiers.

#### autoreconf

Si il y a beaucoup de scripts configure générés par autoconf, le programme autoreconf peut vous faire gagner du temps. Il exécute autoconf et autoheader (si nécessaire) de façon répétée pour reconstruire les scripts configure d'autoconf et les modèles d'entête de configuration dans l'arborescence rattachée au répertoire courant.

#### autoscan

Le programme autoscan peut aider à créer un fichier configure.in pour un package logiciel. autoscan examine les fichiers sources dans l'arborescence de dossiers. Si aucun répertoire n'est fourni sur la ligne de commande, il utilise le répertoire courant. Il cherche dans les fichiers sources les problèmes communs de portabilité et crée un fichier configure.scan qui est un préliminaire à configure.in pour ce package.

### autoupdate

Le programme autoupdate met à jour le fichier configure.in qui appelle les macros autoconf par leurs anciens noms, afin qu'il utilise les noms de macros actuels.

### ifnames

ifnames peut être utile lors de l'écriture d'un configure.in pour un package. Il écrit les identifiants déjà utilisés par le package dans des conditions pour le pré−processeur C. Si un package a déjà été configuré pour avoir une certaine portabilité, ce programme peut aider à identifier ce que son script configure doit vérifier. Ceci peut aider à combler certains manques dans un fichier configure.in généré par autoscan.

## Dépendances d'installation d'Autoconf

Dernière vérification effectuée auprès de la version 2.52.

Bash: sh Diffutils: cmp Fileutils: chmod, install, ln, ls, mkdir, mv, rm Grep: fgrep, grep M4: m4
Make: make Gawk: gawk Sed: sed Sh−utils: echo, expr, hostname, sleep, uname Texinfo: install−info
Textutils: cat, tr

# Automake

## Site Officiel de Téléchargement

Automake (1.6.3):  ftp://ftp.gnu.org/gnu/automake/

## Contenu de Automake

Dernière vérification effectuée auprès de la version 1.6.2.

### Programmes

acinstall, aclocal, aclocal−1.6, automake, automake−1.6, compile, config.guess, config.sub, depcomp, elisp−comp, install−sh, mdate−sh, missing, mkinstalldirs, py−compile, ylwrap

### Descriptions

#### acinstall

acinstall est un script qui installe les fichiers M4 style aclocal.

#### aclocal, aclocal−1.6

automake inclut un certain nombre de macros Autoconf qui peuvent être utilisées dans les packages, certaines d'entre elles sont nécessaires par automake dans certaines situations. Ces macros doivent être définies dans le fichier aclocal.m4, sinon elles ne seront pas vues par autoconf.

Le programme aclocal générera automatiquement des fichiers aclocal.m4 basés sur le contenu de configure.in. Ceci fournit un moyen commode d'obtenir des macros fournies par automake, sans avoir à rechercher ici ou là. De plus, le mécanisme aclocal est extensible à l'usage par d'autres packages.

#### automake, automake−1.6

Pour créer tous les fichiers makefile.in d'un package, exécutez le programme automake dans le répertoire racine de ce package, sans argument. automake trouvera automatiquement chaque Makefile.am approprié (en parcourant configure.in) et générera les Makefile.in correspondants.

#### compile

compile est un script qui agit comme un lien avec les compilateurs.

#### config.guess

config.guess est un script qui essaie de deviner le nom canonique du système.

### config.sub

config.sub est un script contenant une sous−routine de validation de configuration.

### depcomp

depcomp est un script qui compile un programme en générant les dépendances comme effets de bord.

### elisp−comp

elisp−comp est un script qui compile les fichiers .el.

### install−sh

install−sh est un script qui installe un programme, script ouf fichier de données.

### mdate−sh

mdate−sh est un script qui affiche l'heure de modification d'un fichier ou d'un répertoire.

### missing

missing est un script qui agit comme un tube commun pour les quelques programmes GNU manquants lors d'une installation.

### mkinstalldirs

mkinstalldirs est un script qui crée une hiérarchie de répertoires.

### py−compile

py−compile est un script qui compile un programme Python.

### ylwrap

ylwrap est un script qui agit comme un lien en invoquant lex/yacc.

## Dépendances d'installation d'Automake

Dernière vérification effectuée auprès de la version 1.5.

Bash: sh Diffutils: cmp Fileutils: chmod, install, ls, mkdir, mv, rm, rmdir Grep: fgrep, grep
Make: make Perl: perl Sed: sed Sh−utils: echo, expr, hostname, sleep Texinfo: install−info
Textutils: cat, tr

# Bash

## Site Officiel de Téléchargements

Bash (2.05a):  ftp://ftp.gnu.org/gnu/bash/

## Contenu de Bash

Dernière vérification effectuée auprès de la version 2.05a.

### Programmes

bash, sh (lien vers bash) et bashbug

### Descriptions

#### bash

bash est le Bourne−Again SHell, un interpréteur de commandes très largement répandu sur les systèmes Unix. bash est un programme qui lit depuis l'entrée standard, le clavier. L'utilisateur tape quelque chose et le programme évaluera ce qu'il a tapé et fera quelque chose avec, comme lancer un programme.

#### bashbug

bashbug est un shell de script afin d'aider l'utilisateur à composer et à expédier par courrier électronique sous un format standard les rapports de bogues concernant bash.

#### sh

sh est un lien symbolique vers le programme bash. Quand bash est invoqué en tant que sh, il essaye de simuler le comportement du démarrage des versions historiques de sh aussi similairement que possible, tout en restant conforme au standard POSIX aussi.

## Dépendances d'installation de Bash

Dernière vérification effectuée auprès de la version 2.05a.

Bash: bash, sh Binutils: ar, as, ld, ranlib, size Diffutils: cmp
Fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm Gcc: cc, cc1, collect2, cpp0, gcc
Grep: egrep, grep Make: make Gawk: awk Sed: sed
Sh−utils: basename, echo, expr, hostname, sleep, uname Texinfo: install−info Textutils: cat, tr, uniq

# Bin86

## Site Officiel de Téléchargement

Bin86 (0.16.3):  http://www.cix.co.uk/~mayday/

## Contenu de Bin86

Dernière vérification effectuée auprès de la version 0.16.3.

### Programmes

as86, as86_encap, ld86, nm86 (lien vers objdump86), objdump86 et size86 (lien vers objdump86)

### Descriptions

#### as86

as86 est un assembleur pour les processeurs 8086...80386.

#### as86_encap

as86_encap est un script shell pour appeller as86 et convertir le binaire créé en un fichier C prog.v à inclure ou à lier avec des programmes comme les installeurs de blocs de démarrage.

#### ld86

ld86 ne comprends que les fichiers objets produits par l'assembleur as86. Il peut les lier dans un exécutable I&D soit impur, soit séparé.

#### nm86

La table des symboles du fichier binaire.

#### objdump86

Il affiche des informations détaillées sur un fichier binaire.

#### size86

Il résume les tailles des données dans un fichier binaire.

## Dépendances d'installation de Bin86

Dernière vérification effectuée auprès de la version 0.16.0.

Bash: sh Binutils: as, ld, strip Fileutils: chmod, install, ln, mv Gcc: cc, cc1, collect2, cpp0
Make: make Sed: sed

# Binutils

## Site Officiel de Téléchargement

Binutils (2.13):  ftp://ftp.gnu.org/gnu/binutils/

## Contenu de Binutils

Dernière vérification effectuée auprès de la version 2.12.1.

### Programmes

addr2line, ar, as, gasp, gprof, ld, nm, objcopy, objdump, ranlib, readelf, size, strings et strip

### Descriptions

#### addr2line

addr2line traduit les adresses d'un programme en nom de fichiers et numéros de ligne. A partir d'une adresse et d'un exécutable, il utilise les informations de déboguage présentes dans l'exécutable pour trouver quel nom de fichier et quel numéro de ligne sont associés à une adresse donnée.

#### ar

Le programme ar crée, modifie et extrait des données depuis des archives. Une archive est un fichier contenant un ensemble d'autres fichiers dans une structure qui permet de retrouver les fichiers individuels d'origine (appelés membres de l'archive).

#### as

as a d'abord été prévu pour assembler la sortie du compilateur GNU C, gcc, pour qu'elle puisse être utilisée par l'éditeur de liens ld.

#### gasp

gasp est le préprocesseur de macros assembleur.

#### gprof

gprof affiche les données de profil sous forme de graphe d'appel.

#### ld

ld combine plusieurs fichiers objets et archives, modifie l'emplacement de leurs données et lie les références des symboles. Souvent, la dernière étape dans la construction d'un nouveau programme compilé à exécuter est un appel à ld.

### nm

nm liste les symboles des fichiers objet.

### objcopy

L'utilitaire objcopy copie le contenu d'un fichier objet dans un autre. objcopy utilise la bibliothèque GNU BFD pour lire et écrire les fichiers objet. Il peut écrire le fichier objet destination dans un format différent de celui du fichier objet source.

### objdump

objdump affiche des informations sur un ou plusieurs fichiers objet. Les options déterminent quelles informations spécifiques sont à afficher. Ces informations sont essentiellement utiles pour les programmeurs qui travaillent sur des outils de compilation, à l'inverse des programmeurs qui veulent juste compiler leur programme pour qu'il fonctionne.

### ranlib

ranlib génère un index du contenu d'une archive, et le stocke dans l'archive. L'index liste chaque symbole défini par un membre d'une archive, qui est un fichier objet re−localisable.

### readelf

readelf affiche des informations sur les binaires de type elf.

### size

size liste les tailles des sections, et la taille totale, pour chaque fichier objet de sa liste d'arguments. Par défaut, une ligne est générée en sortie pour chaque fichier objet ou chaque module dans une archive.

### strings

Pour chaque fichier donné, strings affiche les séquences de caractères imprimables qui ont au moins une longueur de quatre caractères (ou le nombre spécifié en option du programme) et qui sont suivies d'un caractère non imprimable. Par défaut, il affiche seulement les chaînes des parties initialisées et chargées des fichiers objet. Pour les autres types de fichier, il affiche les chaînes de tout le fichier.

strings est principalement utile pour déterminer le contenu des fichiers binaires.

### strip

strip débarrasse les fichiers objet de tous leurs symboles, ou de certains en particulier. La liste des fichiers objet peut inclure des archives. Au moins un fichier objet doit être fourni. strip modifie les fichiers fournis en argument, plutôt que d'écrire les copies modifiées sous d'autres noms.

## Bibliothèques

libbfd.[a,so] et libopcodes.[a,so]

## Descriptions

### libbfd

libbfd est une bibliothèqes de description des fichiers binaires.

### libopcodes

libopcodes est une bibliothèque native pour traiter les opcodes et est utilisé dans le but de construire des utilitaires comme ojbdump. Les opcodes correspondent à une version texte facilement lisible d'instructions processeurs.

# Dépendances d'installation de Binutils

Dernière vérification effectuée auprès de la version 2.11.2.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh
Binutils: ar, as, ld, nm, ranlib, strip Diffutils: cmp
Fileutils: chmod, cp, ln, ls, mkdir, mv, rm, rmdir, touch Flex: flex
Gcc: cc, cc1, collect2, cpp0, gcc Glibc: ldconfig Grep: egrep, fgrep, grep M4: m4 Make: make
Gawk: gawk Sed: sed Sh−utils: basename, echo, expr, hostname, sleep, true, uname
Texinfo: install−info, makeinfo Textutils: cat, sort, tr, uniq

# Bison

## Site officiel de téléchargement

Bison (1.35):  ftp://ftp.gnu.org/gnu/bison/

## Contenu de bison

Dernière vérification effectuée auprès de la version 1.35.

### Programmes

bison et yacc

### Descriptions

#### bison

bison est un générateur d'analyse syntaxique, en remplacement de YACC. YACC signifie Encore un Autre Compilateur de Compilateurs (Yet Another Compiler Compiler). Qu'est−ce que bison alors ? Ce programme génère un autre programme en analysant la structure d'un fichier texte. Au lieu d'écrire le programme, l'utilisateur spécifie comment les choses doivent être reliées et, avec ces règles, un programme est construit pour analyser le fichier texte.

Il y a de nombreux exemples où une structuration est nécessaire, l'un d'eux est une calculatrice.

Soit la chaine :

```
1 + 2 * 3
```

Un humain peut facilement arriver au résultat de 7. Pourquoi ? A cause de la structure. Notre cerveau sait comment interpréter la chaîne. L'ordinateur ne le sait pas, et bison est un outil pour l'aider à comprendre cela, en présentant la chaîne sous la forme suivante au compilateur :

```
   +         / \         *   1        / \
2   3
```

En commençant en bas de l'arbre, et en arrivant aux nombres 2 et 3, qui sont liés par un symbole de multiplication, l'ordinateur multiplie 2 par 3. Le résultat de cette multiplication est mémorisé et la chose suivante que l'ordinateur voit est le résultat de 2*3 et le nombre 1 qui sont reliés par le symbole d'addition. L'ajout de 1 au résultat précédent donne 7. En calcul, les opérations les plus complexes peuvent être exprimées sous ce format d'arbre, et l'ordinateur commence simplement en bas, et travaille en cheminant vers le haut, pour arriver à la réponse correcte. Bien entendu, bison n'est pas utilisé uniquement pour les calculatrices.

### yacc

Nous créons un script nommé yacc qui appelle bison avec l'option −y. Ceci est fait pour des raisons de compatibilité avec des programmes qui utilisent yacc au lieu de bison.

## Dépendances d'installation de Bison

Dernière vérification effectuée auprès de la version 1.31.

Bash: sh Binutils: ar, as, ld, ranlib Diffutils: cmp
Fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm, rmdir Gcc: cc, cc1, collect2, cpp0, gcc
Grep: egrep, fgrep, grep Make: make Sed: sed
Sh−utils: basename, dirname, echo, expr, hostname, sleep, uname Texinfo: install−info
Textutils: cat, head, tr, uniq

# Bzip2

## Site officiel de

Bzip2 (1.0.2):  ftp://sourceware.cygnus.com/pub/bzip2/

## Contenu de Bzip2

Dernière vérification effectuée auprès de la version 1.0.2.

### Programmes

bunzip2 (lien vers bzip2), bzcat (lien vers bzip2), bzcmp, bzdiff, bzegrep, bzfgrep, bzgrep, bzip2, bzip2recover, bzless et bzmore

### Descriptions

#### bunzip2

bunzip2 décompresse les fichiers compressés avec bzip2.

#### bzcat

bzcat (ou bzip2 −dc) décompresse les fichiers spécifiés sur la sortie standard.

#### bzcmp, bzdiff

bzcmp et bzdiff sont utilisés pour appeler les programmes cmp et diff avec des fichiers compressés avec bzip2.

#### bzegrep, bzfgrep, bzgrep

bzegrep, bzfgrep et bzgrep appellent respectivement egrep, fgrep, or grep sur des fichiers compressés avec bzip2.

#### bzip2

bzip2 compresse les fichiers grâce à l'algorithme Burrows−Wheeler et au codage Huffman. La compression est généralement bien meilleure à celle obtenue avec des compresseurs plus conventionnels basés sur LZ77/LZ78, et approche les performances de la famille PPM des compresseurs statistiques.

#### bzip2recover

bzip2recover recupère les données de fichiers bzip2 endommagés.

#### bzless

bzless est un filtre permettant d'examiner des fichiers textes compressés, un écran à la fois, sur un terminal, de la même manière que less.

#### bzmore

bzmore est un filtre permettant d'examiner des fichiers textes compressés, un écran à la fois, sur un terminal, de la même manière que more.

### Bibliothèques

libbz2.a, libbz2.so (lien vers libbz2.so.1.0), libbz2.so.1.0 (lien vers libbz2.so.1.0.2) et libbz2.so.1.0.2

#### libbz2

libbz2 est la bibliothèque implémentant l'algorithme de compression Burrows−Wheeler.

## Dépendances d'installation de Bzip2

Dernière vérification effectuée auprès de la version 1.0.1.

Bash: sh Binutils: ar, as, ld, ranlib Fileutils: cp, ln, rm Gcc: cc1, collect2, cpp0, gcc Make: make

# Diffutils

## Site officiel de

Diffutils (2.8.1):  ftp://ftp.gnu.org/gnu/diffutils/

## Contenu de Diffutils

Dernière vérification effectuée auprès de la version 2.8.1.

### Programmes

cmp, diff, diff3 et sdiff

### Descriptions

#### cmp et diff

cmp et diff comparent tous les deux deux fichiers et indiquent les différences. Tous les deux disposent d'options particulières pour comparer les fichiers dans des situations différentes.

#### diff3

La différence entre diff et diff3 est que diff compare 2 fichiers, alors que diff3 en compare 3.

#### sdiff

sdiff joint deux fichiers et affiche interactivement le résultat.

## Dépendances d'installation de Diffutils

Dernière vérification effectuée auprès de la version 2.7.

Bash: sh Binutils: ld, as Diffutils: cmp Fileutils: chmod, cp, install, mv, rm
Gcc: cc1, collect2, cpp0, gcc Grep: egrep, grep Make: make Sed: sed Sh−utils: date, hostname
Textutils: cat, tr

# E2fsprogs

## Site officiel de téléchargement

E2fsprogs (1.27): ftp://download.sourceforge.net/pub/sourceforge/e2fsprogs/
http://download.sourceforge.net/e2fsprogs/

## Contenu du package E2fsprogs

Dernière vérification effectuée auprès de la version 1.27.

### Programmes

badblocks, chattr, compile_et, debugfs, dumpe2fs, e2fsck, e2image, e2label, fsck, fsck.ext2, fsck.ext3, lsattr, mk_cmds, mke2fs, mkfs.ext2, mklost+found, resize2fs, tune2fs et uuidgen

### Descriptions

#### badblocks

badblocks est utilisé pour rechercher les secteurs défectueux sur un périphérique (en général une partition de disque).

#### chattr

chattr modifie les attributs d'un fichier sur un système de fichier Linux ext2.

#### compile_et

compile_et est utilisé pour convertir un tableau, regroupant des codes d'erreur et messages associés, en un fichier source C exploitable avec la librairie com_err.

#### debugfs

Le programme debugfs est un débogueur de système de fichiers. Il peut être utilisé pour examiner et modifier l'état d'un système de fichiers ext2.

#### dumpe2fs

dumpe2fs affiche le super−bloc et les informations de groupe de blocs pour le système de fichiers présent sur un périphérique spécifié.

#### e2fsck et fsck.ext2

e2fsck et fsck.ext2 sont utilisés pour vérifier, et parfois réparer, un système de fichiers Linux ext2.

### e2image

e2image est utilisé pour sauvegarder les données critiques d'un système de fichiers ext2 vers un fichier.

### e2label

e2label affiche ou modifie le nom du système de fichiers sur un système de fichiers ext2 situé sur le périphérique spécifié.

### fsck

fsck est utilisé pour vérifier, et éventuellement réparer, un système de fichiers Linux.

### fsck.ext3

fsck.ext3 est utilisé pour vérifier, et éventuellement réparer, un système de fichiers Linux ext3.

### lsattr

lsattr liste les attributs d'un fichier sur un système de fichiers ext2.

### mk_cmds

L'utilitaire mk_cmds prends un fichier de commandes en entrée et produit en sortie un fichier source en C, dont le but est d'être utilisé avec la bibliothèque libss.

### mke2fs et mkfs.ext2

mke2fs est utilisé pour créer un système de fichiers Linux ext2 sur un périphérique (en général une partition de disque). mkfs.ext2 fait la même chose que mke2fs.

### mkfs.ext3

mkfs.ext3 est utilisé pour créer un système de fichiers ext3.

### mklost+found

mklost+found est utilisé pour créer le dossier lost+found dans le répertoire de travail courant d'un système de fichiers Linux ext2. mklost+found pré−alloue des blocs de disque pour le répertoire, pour le rendre utilisable par e2fsck.

### resize2fs

resize2fs est utilisé pour redimensionner un système de fichier ext2.

### tune2fs

tune2fs ajuste les paramètres d'optimisation de système de fichiers sur un système de fichiers Linux ext2.

### uuidgen

Le programme uuidgen crée un nouvel identifiant universel unique (UUID) en utilisant la bibliothèque libuuid. Le nouvel UUID peut raisonnablement être considéré comme unique parmi tous les UUID créés, sur le système local et sur d'autres systèmes, dans le passé ou le futur.

## Bibliothèques

libcom_err.[a,so], libe2p.[a,so], libext2fs.[a,so], libss.[a,so], libuuid.[a,so]

## Descriptions

### libcom_err

La routine d'affichage des erreurs.

### libe2p

libe2p est utilisé par dumpe2fs, chattr, et lsattr.

### libext2fs

La bibliothèque ext2fs a été créée pour permettre à des programmes utilisateurs de manipuler un système de fichiers ext2.

### libss

libss est utilisé par debugfs.

### libuuid

La bibliothèque libuuid est utilisé pour générer des identifiants uniques pour les objets qui peuvent être accessibles en dehors du système local.

# Dépendances d'installation d'E2fsprogs

Dernière vérification effectuée auprès de la version 1.25.

Bash: sh Binutils: ar, as, ld, ranlib, strip Diffutils: cmp
Fileutils: chmod, cp, install, ln, mkdir, mv, rm, sync Gcc: cc, cc1, collect2, cpp0 Glibc: ldconfig
Grep: egrep, grep Gzip: gzip Make: make Gawk: awk Sed: sed
Sh−utils: basename, echo, expr, hostname, uname Texinfo: makeinfo Textutils: cat, tr

# Ed

## Site officiel de téléchargement

Ed (0.2):  ftp://ftp.gnu.org/gnu/ed/ Correctif Ed (0.2): ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/ http://ftp.linuxfromscratch.org/lfs−packages/4.0/

## Contenu de Ed

Dernière vérification effectuée auprès de la version 0.2.

### Programmes

ed et red (lien vers ed)

### Description

#### ed

ed est un éditeur de texte par ligne. Il est utilisé pour créer, afficher, modifier et réaliser d'autres manipulations sur les fichiers texte.

#### red

red est un ed resteint : il ne peut que éditer les fichiers du répertoire courant et ne peut pas exécuter des commandes du shell.

## Dépendances d'installation de Ed

Dernière vérification effectuée auprès de la version 0.2.

Bash: sh Binutils: ar, as, ld, ranlib Diffutils: cmp Fileutils: chmod, cp, install, ln, mv, rm, touch Gcc: cc1, collect2, cpp0, gcc Grep: egrep, grep Make: make Sed: sed Sh−utils: hostname Textutils: cat, tr

# File

## Site officiel de téléchargement

File (3.39):  ftp://ftp.gw.com/mirrors/pub/unix/file/

## Contenu de File

Dernière vérification effectuée auprès de la version 3.39.

### Programmes

file

### Description

#### file

file teste le fichier spécifié et essaie de le classifier. Il y a trois séries de tests, exécutées dans cet ordre : tests du système de fichiers, tests du nombre magique et tests de langage. Le premier test à s'exécuter avec succès affiche le type du fichier traité.

## Dépendances d'installation de File

Dernière vérification effectuée auprès de la version 3.37.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh Binutils: as, ld Diffutils: cmp Fileutils: chmod, install, ln, ls, mv, rm, touch Gcc: cc1, collect2, cpp0, gcc Grep: egrep, grep M4: m4 Make: make Gawk: gawk Sed: sed Sh−utils: echo, expr, hostname, sleep Texinfo: makeinfo Textutils: cat, tr

# Fileutils

## Site officiel de téléchargement

Fileutils (4.1): ftp://ftp.gnu.org/gnu/fileutils/ Correctif Fileutils (4.1): ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/ http://ftp.linuxfromscratch.org/lfs−packages/4.0/

## Contenu de Fileutils

Dernière vérification effectuée auprès de la version 4.1.

### Programmes

chgrp, chmod, chown, cp, dd, df, dir, dircolors, du, install, ln, ls, mkdir, mkfifo, mknod, mv, rm, rmdir, shred, sync, touch et vdir

### Descriptions

#### chgrp

chgrp remplace le groupe des propriétaires des fichiers spécifiés par celui dont le nom ou le numéro est passé en paramètre.

#### chmod

chmod change les permissions des fichiers spécifiés en fonction du mode, ce mode pouvant être une représentation symbolique des changements à effectuer, ou une représentation octale d'un nombre représentant le champ de bits des nouvelles permissions.

#### chown

chown change le propriétaire et/ou le groupe de propriétaires des fichiers spécifiés.

#### cp

cp copie des fichiers d'un répertoire à un autre.

#### dd

dd copie un fichier (de l'entrée standard vers la sortie standard, par défaut) en fonction d'une taille de bloc et des options de conversion du fichier données par l'utilisateur.

#### df

df affiche l'espace disque disponible du système de fichiers contenant les fichiers passés en argument. Sans argument, df indiquera les quantités correspondant à tous les systèmes de fichiers montés.

### dir, ls et vdir

dir et vdir sont des versions de ls avec des formats d'affichage différents. Ces programmes listent tous les fichiers ou répertoires donnés en argument. Le contenu des répertoires est trié alphabétiquement. Pour ls, les fichiers sont affichés en colonnes par défaut et triés verticalement si la sortie standard est un terminal; ils sont affichés ligne par ligne dans le cas contraire. Pour dir, les fichiers sont par défaut affichés en colonnes et triés verticalement. Pour vdir, l'affichage par défaut est le format long.

### dircolors

dircolors affiche une liste de commandes permettant de définir la variable d'environnement LS_COLOR. La variable LS_COLOR est utilisée pour changer les couleurs d'affichage par défaut de la commande ls et d'autres utilitaires.

### du

du affiche l'espace disque utilisé par chaque fichier et répertoire listés en argument sur la ligne de commande et pour chacun de leurs sous−répertoires.

### install

install copie les fichiers puis positionne leurs permissions et, si possible, leur propriétaire et leur groupe.

### ln

ln crée des liens matériels ou symboliques entre fichiers.

### mkdir

mkdir crée un répertoire correspondant à chacun des noms mentionnés en argument.

### mkfifo

mkfifo crée un fichier FIFO avec les noms passés en argument.

### mknod

mknod crée un fichier FIFO, fichier spécial en mode caractère ou fichier spécial en mode bloc, avec le nom passé en argument.

### mv

mv déplace des fichiers d'un répertoire à un autre ou les renomme, selon les arguments qui lui sont passés.

### rm

rm efface des fichiers ou des répertoires.

### rmdir

rmdir efface des répertoires, s'ils sont vides.

### shred

shred efface un fichier de façon sûre, en écrasant son contenu afin qu'il ne puisse pas être récupéré.

### sync

sync force l'écriture des blocs modifiés et met à jour le super bloc.

### touch

touch remplace le temps d'accès et de modification des fichiers passés en argument par l'heure et la date courantes. S'ils n'existent pas, des fichiers vides sont créés.

## Dépendances d'installation de Fileutils

Dernière vérification effectuée auprès de la version 4.1.

Bash: sh Binutils: ar, as, ld, ranlib Diffutils: cmp
Fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm, rmdir Gettext: msgfmt, xgettext
Gcc: cc, cc1, collect2, cpp0, gcc Grep: egrep, fgrep, grep Make: make Perl: perl Sed: sed
Sh−utils: basename, echo, expr, hostname, sleep, uname Texinfo: install−info Textutils: cat, tr

# Findutils

## Site officiel de téléchargement

Findutils (4.1): ftp://ftp.gnu.org/gnu/findutils/ Correctif Findutils (4.1): ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/ http://ftp.linuxfromscratch.org/lfs−packages/4.0/

## Contenu du package Findutils

Dernière vérification effectuée auprès de la version 4.1.

### Programmes

bigram, code, find, frcode, locate, updatedb et xargs

### Descriptions

#### bigram

bigram est utilisé avec la commande code pour produire des bases de données de localisation ancienne formule. Pour en savoir plus sur ces trois derniers programmes, lire les pages de manuel updatedb.5.

#### code

code est l'ancêtre de frcode. Il était utilisé dans les anciennes bases de données de localisation.

#### find

Le programme find recherche des fichiers dans une arborescence selon certains critères. Si aucun critère n'est défini, il liste tous les fichiers du répertoire courant et de ses sous−répertoires.

#### frcode

frcode est lancé par updatedb pour compresser la liste des fichiers de façon à réduire la taille de la base de données d'un facteur 4 à 5.

#### locate

locate scanne une base de données contenant tous les fichiers et répertoires d'un système de fichiers. Ce programme liste les fichiers et répertoires de cette base de données selon certains critères de concordance. Si un utilisateur recherche un fichier, ce programme scannera la base de données pour lui donner l'emplacement exact où ces fichiers sont situés. Ceci n'a de sens que si la base de données est régulièrement mise à jour, autrement cela ne fournirait que des informations dépassées.

#### updatedb

Le programme updatedb met à jour la base de données de localisation. Il parcourt entièrement les systèmes de fichiers (sans contre indications, il parcourt tous les systèmes de fichiers montés) et met à jour la base de

données qui sera ensuite utilisée par le programme locate pour retrouver ces informations. Lancer cette commande quotidiennement est une bonne pratique pour retrouver des informations à jour au moment opportun.

### xargs

La commande xargs applique une commande à une liste de fichiers. S'il faut appliquer la même commande à de multiples fichiers, une liste peut être créée pour contenir la liste de tous ces fichiers (un par ligne) et xargs exécuté une commande sur chacun d'eux.

## Dépendances d'installation de Findutils

Dernière vérification effectuée auprès de la version 4.1.

Bash: sh Binutils: ar, as, ld, ranlib Diffutils: cmp Fileutils: chmod, cp, install, mv, rm
Grep: egrep, grep Gcc: cc1, collect2, cpp0, gcc Make: make Patch: patch Sed: sed
Sh−utils: basename, date, echo, hostname Textutils: cat, tr

# Flex

## Site officiel de téléchargement

Flex (2.5.4a):  ftp://ftp.gnu.org/non−gnu/flex/

## Contenu de Flex

Dernière vérification effectuée auprès de la version 2.5.4a.

### Programmes

flex, flex++ (lien vers flex) et lex

### Descriptions

#### flex

flex est un utilitaire destiné à générer des programmes pour la reconnaissance de motifs dans des textes. La reconnaissance de motifs est très utile dans de nombreuses applications. Un utilisateur définit des règles à rechercher et flex créera un programme destiné à reconnaître ces motifs. Utiliser flex est un avantage car il est beaucoup plus facile de définir des règles de recherche que d'écrire un programme pour repérer le texte voulu.

#### flex++

flex++ invoque une version de flex utilisé exclusivement par les scanners C++.

#### lex

Nous créons un script bash, nommé lex, appellant flex avec l'option −l. Ceci est fait pour des raisons de compatibilité avec les programmes utilisant lex au lieu de flex.

### Bibliothèques

libfl.a

### Descriptions

#### libfl

libfl est la bibliothèque flex.

## Dépendances d'installation de Flex

Dernière vérification effectuée auprès de la version 2.5.4a.

Bash: sh Binutils: ar, as, ld, ranlib Bison: bison Diffutils: cmp
Fileutils: chmod, cp, install, ln, mv, rm, touch Gcc: cc1, collect2, cpp0, gcc Grep: egrep, grep
Make: make Sed: sed Sh−utils: echo, hostname Textutils: cat, tr

# Gawk

## Site officiel de téléchargement

Gawk (3.1.1): ftp://ftp.gnu.org/pub/gnu/gawk/ Correctif Gawk (3.1.1−2): ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/ http://ftp.linuxfromscratch.org/lfs−packages/4.0/

## Contenu de Gawk

Dernière vérification effectuée auprès de la version 3.1.1.

### Programmes

awk, gawk, gawk−3.1.1, grcat, igawk, pgawk, pgawk−3.1.1, pwcat

### Descriptions

#### awk

awk est un lien symbolique vers gawk.

#### gawk, gawk−3.1.1

gawk est l'implémentation GNU de awk, un langage basé sur le traitement et la reconnaissance de chaînes de caractères.

#### grcat

grcat concatène la base de données des groupes, /etc/group.

#### igawk

igawk est un script shell qui donne à gawk la possibilité d'inclure des fichiers.

#### pgawk, pgawk−3.1.1

pgawk est la version de débuggage de gawk.

#### pwcat

pwcat concatène la base de données des utilisateurs, /etc/passwd.

## Dépendances d'installation de Gawk

Dernière vérification effectuée auprès de la version 3.1.0.

(Aucune dépendance encore vérifiée)

# GCC

## Site officiel de téléchargement

GCC (3.2):  ftp://ftp.gnu.org/pub/gnu/gcc/ Correctif GCC (3.2): ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/ http://ftp.linuxfromscratch.org/lfs−packages/4.0/ Correctif nofixincludes pour GCC (3.2): ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/ http://ftp.linuxfromscratch.org/lfs−packages/4.0/

## Contenu de GCC

Dernière vérification effectuée auprès de la version 3.1.

### Programmes

c++, c++filt, cc (lien vers gcc), cc1, cc1plus, collect2, cpp, cpp0, g++, gcc, gccbug, gcov, tradcpp0

### Descriptions

#### cc, cc1, cc1plus, gcc

Ce sont les compilateurs C. Un compilateur transforme le code source au format texte dans un format que l'ordinateur comprends. Après qu'un code source ait été compilé en un fichier objet, un éditeur de lien va créer un fichier exécutable à partir d'un ou plusieurs de ces fichiers générés par le compilateur.

#### c++, cc1plus, g++

Ce sont les compilateurs C++, l'équivalent de cc et gcc, etc...

#### c++filt

Le langage C++ procure la surcharge de fonctions, c'est−à−dire qu'il est possible d'écrire plusieurs fonctions possédant le même nom, à condition que chaque fonction utilise des paramètres de différents types. Tous les noms des fonctions C++ sont codés avec un label assembleur bas niveau (ce procédé est aussi connu sous le nom de mangling). Le programme c++filt réalise l'opération inverse; il décode les noms bas niveau en des noms de niveau utilisateur, de manière à ce que l'éditeur de liens ne confonde pas les fonctions surchargées.

#### collect2

collect2 assiste à la compilation de constructeurs.

#### cpp, cpp0

cpp réalise un travail préliminaire sur un fichier source, comme inclure le contenu des fichiers d'entêtes dans le code source. Il suffit d'insérer une ligne comme #include <filename> dans votre fichier source. Le préprocesseur va insérer le contenu de ce fichier dans le fichier source.

#### gccbug

gccbug est un script shell, utilisé pour simplifier la création de rapports de bugs.

#### gcov

gcov analyse les programmes pour aider à créer des codes plus efficaces, plus rapides avec une optimisation.

#### tradcpp0

Aucune description actuellement disponible.

### Bibliothèques

libgcc.a, libgcc_eh.a, libgcc_s.so, libiberty.a, libstdc++.[a,so], libsupc++.a

### Descriptions

#### libgcc, libgcc_eh, libgcc_s

Fichiers de support pour gcc.

#### libiberty

libiberty est une collection de sous−routines utilisées par différents programmes GNU, comme getopt, obstack, strerror, strtol et strtoul.

#### libstdc++

libstdc++ est la bibliothèque C++. Elle est utilisée par les programmes C++ et contient des fonctions couramment utilisées par eux. De cette façon, le développeur n'a pas besoin d'écrire certaines fonctions (comme écrire une ligne de texte à l'écran) à partir de rien, à chaque fois qu'il crée un programme.

#### libsupc++

libsup++ procure une aide au langage de programmation c++. Entre autres choses, libsup++ contient des routines pour la gestion des exceptions.

## Dépendances d'installation de GCC

Dernière vérification effectuée auprès de la version 2.95.3.

Bash: sh Binutils: ar, as, ld, nm, ranlib Diffutils: cmp
Fileutils: chmod, cp, ln, ls, mkdir, mv, rm, touch Find: find Gcc: cc, cc1, collect2, cpp0, gcc
Grep: egrep, grep Make: make Patch: patch Sed: sed
Sh−utils: basename, dirname, echo, expr, hostname, sleep, true, uname Tar: tar
Texinfo: install−info, makeinfo Textutils: cat, tail, tr

# Gettext

## Site officiel de téléchargement

Gettext (0.11.5):  ftp://ftp.gnu.org/gnu/gettext/

## Contenu de Gettext

Dernière vérification effectuée auprès de la version 0.11.2.

### Programmes

config.charset, config.rpath, gettext, gettextize, hostname, msgattrib, msgcat, , msgcmp, msgcomm, msgconv, msgen, msgexec, msgfilter, msgfmt, msggrep, msginit, msgmerge, msgunfmt, msguniq, ngettext, project−id, team−address, trigger, urlget, user−email et xgettext

### Descriptions

#### config.charset

Le script config.charset affiche une table, dépendant du système, des alias de caractères.

#### config.rpath

Le script config.rpath affiche un ensemble de variables, dépendant du système, décrivant comment initialiser le chemin de recherche des bibliothèques dynamiques dans un exécutable.

#### gettext

Le package gettext est utilisé pour l'internationalisation (aussi nommée i18n) et pour la localisation (connue aussi sous le nom de l10n). Les programmes peuvent ainsi être compilés avec le support de langue native (Native Language Support) pour permettre l'affichage dans la langue native de l'utilisateur plutôt qu'en anglais qui est la langue par défaut.

#### gettextize

Le programme gettextize copie tous les fichiers au standard gettext dans un répertoire. Ceci est utile pour créer un package de traductions gettext.

#### hostname

Le programme hostname affiche le nom réseau de l'hôte, dans plusieurs formats.

#### msgattrib

Le programme msgattrib filtre les messages d'un catalogue de traduction d'après leurs attributs, et manipule les attributs.

### msgcat

Le programme msgcat trouve les messages qui sont communs à plusieurs traductions.

### msgcmp

Le programme msgcmp compare deux fichiers de traduction.

### msgcomm

Le programme msgcomm recherche les messages qui apparaissent dans plusieurs fichiers .po. Ceci est utile pour comparer l'état de la traduction.

### msgconv

Le programme msgconv convertit un catalogue de traduction dans un encodage de caractères différent.

### msgen

Le programme msgen crée un catalogue de traductions anglaises.

### msgexec

Le programme msgexec applique une commande à toutes les traductions d'un catalogue de traductions.

### msgfilter

Le programme msgfilter applique un filtre à toutes les traductions d'un catalogue de traductions.

### msgfmt

Le programme msgftm compile les traductions en code machine. Ceci est utile pour créer le fichier final de traduction du programme/package.

### msggrep

Le programme msggrep extrait tous les messages d'un catalogue de traduction qui suive un certain modèle ou appartienne à des fichiers sources donnés.

### msginit

Le programme msginit crée un nouveau fichier PO, initialisant les informations meta avec les valeurs de l'environnement de l'utilisateur.

### msgmerge

Le programme msgmerge combine deux traductions en un fichier. Cela permet de mettre à jour les fichiers de traduction à partir de la source.

### msgunfmt

Le programme msgunfmt décompile les fichiers de traduction en leur version texte. Ceci est utilisé uniquement dans le cas où les version compilées sont disponibles.

### msguniq

Le programme msguniq unifie des traductions dupliquées dans un catalogue de traduction.

### ngettext

Le programme ngettext affiche la traduction dans le langage natif d'un message textuel dont la forme grammaticale dépend d'un nombre.

### project−id

Le script project−id affiche l'identification d'un package ou la version d'un package.

### team−address

Le script team−address envoie l'adresse de l'équipe sur stdout et affiche quelques informations supplémentaires.

### trigger

Le script trigger teste si le package courant est un package GNOME ou KDE.

### urlget

Le programme urlget récupère le contenu d'une URL.

### user−email

Le script user−email affiche l'adresse email de l'utilisateur, avec la confirmation de l'utilisateur.

### xgettext

Le programme xgettext extrait les lignes du message à partir des sources C. Cela permet la création du fichier modèle de traduction.

## Bibliothèques

libgettextlib[a,so], libgettextsrc[a,so]

## Descriptions

### libgettextlib

Aucune description actuellement disponible.

### libgettextsrc

Aucune description actuellement disponible.

## Dépendances d'installation de Gettext

Dernière vérification effectuée auprès de la version 0.10.40.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh
Binutils: ar, as, ld, nm, ranlib, strip Bison: bison Diffutils: cmp
Fileutils: chmod, install, ln, ls, mkdir, mv, rm, rmdir Gcc: cc, cc1, collect2, cpp0, gcc
Grep: egrep, fgrep, grep M4: m4 Make: make Gawk: gawk Sed: sed
Sh−utils: basename, echo, expr, hostname, sleep, uname Texinfo: install−info, makeinfo
Textutils: cat, sort, tr, uniq

# Glibc

## Site officiel de téléchargement

Glibc (2.2.5): ftp://ftp.gnu.org/gnu/glibc/ Glibc−linuxthreads (2.2.5): ftp://ftp.gnu.org/gnu/glibc/ Correctif Glibc (2.2.5−2): ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/ http://ftp.linuxfromscratch.org/lfs−packages/4.0/

## Contenu de Glibc

Dernière vérification effectuée auprès de la version 2.2.5.

### Programmes

catchsegv, gencat, getconf, getent, glibcbug, iconv, iconvconfig, ldconfig, ldd, lddlibc4, locale, localedef, mtrace, nscd, nscd_nischeck, pcprofiledump, pt_chown, rpcgen, rpcinfo, sln, sprof, tzselect, xtrace, zdump et zic

### Descriptions

#### catchsegv

catchsegv peut être utilisé pour créer une trace de la pile lorsqu'un programme s'interrompt avec une erreur 'segmentation fault'.

#### gencat

gencat génère des catalogues de messages.

#### getconf

getconf affiche les valeurs de configuration du système pour les variables spécifiques aux systèmes de fichiers.

#### getent

getent récupère des entrées d'une base de données administrative.

#### glibcbug

glibcbug crée un rapport de bug sur glibc et l'envoie par courrier électronique à l'adresse email pour les bugs.

#### iconv

iconv réalise des conversions de jeux de caractères.

### iconvconfig

iconvconfig crée un fichier de configuration chargé rapidement pour le module iconv.

### ldconfig

ldconfig crée un cache des bibliothèques dynamiques pour l'éditeur de liens dynamiques.

### ldd

ldd affiche les bibliothèques partagées requis par chaque programme ou bibliothèque partagée spécifié sur la ligne de commande.

### lddlibc4

lddlibc4 assiste ldd avec les fichiers objets.

### locale

locale est un programme Perl indiquant au compilateur d'autoriser (ou non) l'utilisation des variables locales POSIX pour les opérations intégrées.

### localedef

localedef compile les spécifications pour locale.

### mtrace

mtrace affiche les chemins 'multicast' d'une source à un récepteur (une requête des traces IP).

### nscd

nscd est un démon procurant un cache pour les requêtes DNS les plus courantes.

### nscd_nischeck

nscd_nischeck vérifie si un mode de sécurité est nécessaire pour les recherches NIS+.

### pcprofiledump

pcprofiledump affiche l'information générée par 'PC profiling'.

### pt_chown

pt_chown indique l'utilisateur, le groupe et les permissions d'accès au terminal du pseudo terminal esclave correspondant au pseudo terminal maître passé sur le descripteur de fichier 3. C'est le programme d'aide pour la fonction 'grantpt'. Il n'a pas été conçu pour être lancé directement en ligne de commande.

### rpcgen

rpcgen génère le code C pour implémenter le protocole RPC.

### rpcinfo

rpcinfo fait un appel RPC vers un serveur RPC.

### sln

sln crée le lien symbolique entre une source et sa destination. C'est lié statiquement, sans nécessiter de lien dynamique. Donc, sln est utilisé pour créer des liens symboliques avec des bibliothèques dynamiques si, pour quelque raison que ce soit, le système de liens dynamiques n'était pas fonctionnel.

### sprof

sprof lit et affiche les données d'objets partagés profilés.

### tzselect

tzselect questionne l'utilisateur sur sa position géographique courante et affiche la description de fuseau horaire résultante sur la sortie standard.

### xtrace

xtrace trace l'exécution de programmes en affichant la fonction actuellement exécutée.

### zdump

zdump est le dumper du fuseau horaire.

### zic

zic est le compilateur de fuseau horaire.

## Bibliothèques

ld.so, libBrokenLocale.[a,so], libBrokenLocale_p.a, libSegFault.so, libanl.[a,so], libanl_p.a, libbsd−compat.a, libc.[a,so], libc_nonshared.a, libc_p.a, libcrypt.[a,so], libcrypt_p.a, libdl.[a,so], libdl_p.a, libg.a, libieee.a, libm.[a,so], libm_p.a, libmcheck.a, libmemusage.so, libnsl.a, libnsl_p.a, libnss_compat.so, libnss_dns.so, libnss_files.so, libnss_hesiod.so, libnss_nis.so, libnss_nisplus.so, libpcprofile.so, libpthread.[a,so], libpthread_p.a, libresolv.[a,so], libresolv_p.a, librpcsvc.a, librpcsvc_p.a, librt.[a,so], librt_p.a, libthread_db.so, libutil.[a,so] et libutil_p.a

## Descriptions

### ld.so

ld.so est le programme d'aide pour les exécutables utilisant des bibliothèques partagées.

### libBrokenLocale, libBrokenLocale_p

Utilisé par des logiciels, tels que Mozilla, pour résoudreles locales manquantes.

### libSegFault

libSegFault est un gestionnaire pour le signal 'segmentation fault'. Il essaie de capturer ces signaux.

### libanl, libanl_p

libanl est une bibliothèque de recherche de 'asynchronous name'.

### libbsd−compat

libbsd−compat procure une portabilité nécessaire pour lancer certains programmes sous Linux.

### libc, libc_nonshared, libc_p

Ces fichiers constituent la principale bibliothèque C. Celle−ci est une collection de fonctions couramment utilisées dans les programmes. Cela évite au développeur d'écrire ses propres fonctions pour toutes sortes de tâches. Les plus communes telles que l'affichage d'une chaîne de caractères à l'écran sont déjà présentes et à disposition du programmeur.

La bibliothèque C (comme presque toutes les bibliothèques) est disponible en deux versions : dynamique et statique. En résumé, lorsqu'un programme utilise une bibliothèque C statique, le code de cette bibliothèque est copié dans l'exécutable. Lorsqu'un programme utilise une bibliothèque dynamique, cet exécutable ne contient pas le code de la bibliothèque, mais une routine qui charge la fonction de la bibliothèque au moment où le programme l'utilise. Cela réduit d'une façon significative la taille du programme. La documentation fournie avec la bibliothèque C décrit ce mécanisme plus en détails, il est trop compliqué de l'expliquer ici en une ou deux lignes.

### libcrypt, libcrypt_p

libcrypt est la bibliothèque de cryptographie.

### libdl, libdl_p

libdl est la bibliothèque d'interface pour le chargeur dynamique de liens.

### libg

libg est une bibliothèque de lancement pour g++.

### libieee

libieee est la bibliothèque des nombres flottants IEEE.

### libm, libm_p

libm est la bibliothèque de mathématiques.

### libmcheck

libmcheck contient du code exécuté au démarrage.

### libmemusage

libmemusage est utilisé par memusage pour aider à récupérer des informations sur l'utilisation de la mémoire par un programme.

### libnsl, libnsl_p

libnsl est la bibliothèque des services réseau.

### libnss_compat, libnss_dns, libnss_files, libnss_hesiod, libnss_nis, libnss_nisplus

L'idée principale est de mettre l'implémentation des différents services offerts pour accéder aux bases de données dans des modules séparés. Ceci a un certain nombre d'avantages:

- les contributeurs peuvent ajouter de nouveaux services sans les ajouter à la bibliothèque GNU C.
- les modules peuvent être mis à jour séparement.
- l'image de la bibliothèque C est plus petite.

### libpcprofile

Code utilisé par le noyau pour surveiller le temps CPU passé dans les fonctions, les lignes de codes sources et les instructions.

### libpthread, libpthread_p

La bibliothèques POSIX des threads.

### libresolv, libresolv_p

Les fonctions de cette bibliothèque permettent la création, l'envoi et l'interprétation des paquets des serveurs de noms Internet.

### librpcsvc, librpcsvc_p

Les fonctions de cette bibliothèque apportent différents services RPC.

### librt, librt_p

Les fonctions de cette bibliothèque apportent la plupart des interfaces spécifiées par l'extension temps réel POSIX.1b.

### libthread_db

Les fonctions de cette bibliothèque sont utiles pour construire des debuggers pour les programmes multi−tâches.

### libutil, libutil

Contient du code pour les fonctions standards utilisé pour les différents utilitaires Unix.

## Dépendances d'installation de Glibc

Dernière vérification effectuée auprès de la version 2.2.5.

Bash: sh Binutils: ar, as, ld, ranlib, readelf Diffutils: cmp
Fileutils: chmod, cp, install, ln, mknod, mv, mkdir, rm, touch Gcc: cc, cc1, collect2, cpp, gcc
Grep: egrep, grep Gzip: gzip Make: make Gawk: gawk Sed: sed
Sh−utils: date, expr, hostname, pwd, uname Texinfo: install−info, makeinfo
Textutils: cat, cut, sort, tr

# Grep

## Site officiel de téléchargement

Grep (2.5):  ftp://ftp.gnu.org/gnu/grep/

## Contenu de Grep

Dernière vérification effectuée auprès de la version 2.5.

### Programmes

egrep (lien vers grep), fgrep (lien vers grep) et grep

### Descriptions

#### egrep

egrep affiche les lignes des fichiers correspondant au motif d'une expression régulière étendue.

#### fgrep

fgrep affiche les lignes des fichiers contenant une expression littérale.

#### grep

grep affiche les lignes des fichiers correspondant au motif d'une expression régulière basique.

## Dépendances d'installation de Grep

Dernière vérification effectuée auprès de la version 2.4.2.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh Binutils: as, ld Diffutils: cmp
Fileutils: chmod, install, ls, mkdir, mv, rm Gettext: msgfmt, xgettext
Gcc: cc, cc1, collect2, cpp0, gcc Glibc: getconf Grep: egrep, fgrep, grep M4: m4 Make: make
Gawk: gawk Sed: sed Sh−utils: basename, echo, expr, hostname, sleep, uname
Texinfo: install−info, makeinfo Textutils: cat, tr

# Groff

## Site officiel de téléchargement

Groff (1.18):  ftp://ftp.gnu.org/gnu/groff/

## Contenu de Groff

Dernière vérification effectuée auprès de la version 1.17.2.

### Programmes

addftinfo, afmtodit, eqn, geqn (lien vers eqn), grn, grodvi, groff, grog, grolbp, grolj4, grops, grotty, gtbl (lien vers tbl), hpftodit, indxbib, lkbib, lookbib, mmroff, neqn, nroff, pfbtops, pic, post−grohtml, pre−grohtml, refer, soelim, tbl, tfmtodit, troff et zsoelim (lien vers soelim)

### Descriptions

#### addftinfo

addftinfo lit un fichier de polices troff et ajoute des informations de description des polices utilisées par le système groff.

#### afmtodit

afmtodit crée un fichier de polices utilisables par groff et grops.

#### eqn

eqn traduit les descriptions d'équations contenues dans des fichiers sources en commandes compréhensibles par troff.

#### geqn

geqn est l'implémentation GNU de eqn.

#### grn

grn est un préprocesseur groff pour les fichiers gremlin.

#### grodvi

grodvi est un driver de groff qui produit un format TeX dvi.

#### groff

groff est l'interface du système de formatage de documents groff. Normalement, il exécute le programme troff et le post−processeur approprié en fonction du périphérique sélectionné.

### grog

grog lit les fichiers et détermine quelles options de groff −e, −man, −me, −mm, −ms, −p, −s et −t sont nécessaires pour l'impression des fichiers, et affiche la commande groff avec ces options sur la sortie standard.

### grolbp

grolbp est un driver pour groff qui permet l'impression sur des imprimantes Canon CAPSL (imprimantes laser série LBP−4 et LBP−8).

### grolj4

grolj4 est un driver pour groff qui permet l'impression au format PCL5 pour une imprimante HP Laserjet 4.

### grops

grops convertit la sortie troff GNU en Postscript.

### grotty

grotty convertit la sortie troff GNU en un format compatible pour les périphériques de type 'typewriter'.

### gtbl

gtbl est l'implémentation GNU de tbl.

### hpftodit

hpftodit crée un fichier de police utilisable avec groff −Tlj4 depuis un fichier de type 'HP tagged police metric'.

### indxbib

indxbib crée un index inversé pour la base de données bibliographique d'un fichier donné de façon à pouvoir utiliser refer, lookbib, et lkbib.

### lkbib

lkbib cherche les références dans les bases de données bibliographiques qui contiennent certaines clefs et affiche toutes les références trouvées sur la sortie standard.

### lookbib

lookbib affiche une invite sur le flux d'erreur standard (à moins que l'entrée standard ne soit pas un terminal), lit depuis l'entrée standard une suite de mots clefs, recherche dans les bases de données bibliographiques du fichier spécifié les références contenant ces mots clefs, les affiche sur la sortie standard, et répète ce processus jusqu'à ce qu'il n'y ait plus d'entrée.

### mmroff

mmroff est un préprocesseur basic pour groff.

### neqn

Le script neqn formate les équations pour une sortie ASCII.

### nroff

Le script nroff émule la commande nroff en utilisant groff.

### pfbtops

pfbtops convertit une police Postscript au format .pfb en ASCII.

### pic

pic compile des descriptions d'images contenues dans des fichiers sources en commandes compréhensibles par TeX ou troff.

### pre−grohtml et post−grohtml

pre− et post−grohtml traduisent la sortie de GNU troff au format html.

### refer

refer copie le contenu d'un fichier sur la sortie standard, sauf les parties situées entre .[ et .] interprétées comme des citations, et celles situées entre .R1 et .R2 qui sont interprétées comme des commandes de traitement de ces citations.

### soelim

soelim examine des fichiers et remplace les lignes du style *.so fichier* par le contenu de *fichier*.

### tbl

tbl traduit les descriptions de tables contenues dans des fichiers sources en commandes compréhensibles par troff.

### tfmtodit

tfmtodit crée un fichier police utilisable par **`groff -Tdvi`**.

### troff

troff est quasi identique à la commande Unix troff. Normalement il devrait être appelé par la commande groff, qui exécutera les préprocesseurs et post− processeurs dans l'ordre approprié et avec les options appropriées.

### zsoelim

zsoelim est l'implémentation GNU de soelim.

## Dépendances d'installation de Groff

Dernière vérification effectuée auprès de la version 1.17.2.

Bash: sh Binutils: ar, as, ld, ranlib Bison: bison Diffutils: cmp
Fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm, touch
Gcc: cc1, cc1plus, collect2, cpp0, g++, gcc Grep: egrep, grep Make: make Gawk: awk Sed: sed
Sh−utils: basename, date, echo, expr, hostname, uname Textutils: cat, tr

# Gzip

## Site officiel de téléchargement

Gzip (1.2.4a):  ftp://ftp.gnu.org/gnu/gzip/

## Contenu de Gzip

Dernière vérification effectuée auprès de la version 1.2.4a.

### Programmes

gunzip (lien vers gzip), gzexe, gzip, uncompress (lien vers gunzip), zcat (lien vers gzip), zcmp, zdiff, zforce, zgrep, zmore et znew

### Description

#### gunzip, uncompress

gunzip et uncompress décompressent les fichiers qui ont été compressés par gzip.

#### gzexe

gzexe vous permet de compresser des exécutables et de les décompresser puis de les lancer automatiquement (avec cependant une baisse des performances).

#### gzip

gzip réduit la taille des fichiers en utilisant l'algorithme de compression Lempel–Ziv (LZ77).

#### zcat

zcat décompresse, et écrit sur la sortie standard, une liste de fichiers donnée en ligne de commande ou sur l'entrée standard en lecture.

#### zcmp

zcmp a le même fonctionnement que le programme cmp mais sur des fichiers compressés.

#### zdiff

zdiff a le même fonctionnement que le programme diff mais sur des fichiers compressés.

#### zforce

zforce force l'extension .gz sur les fichiers compressés par gzip de façon à ce que gzip ne les compresse pas une deuxième fois. Ceci peut être utile pour des fichiers dont le nom a été tronqué après un transfert de fichier.

### zgrep

zgrep a le même fonctionnement que le programme grep mais sur des fichiers compressés.

### zmore

zmore est un programme permettant d'avoir le comportement du programme more aussi bien sur des fichiers normaux que compressés.

### znew

znew convertit les fichiers de format .Z (compress) en .gz (gzip).

## Dépendances d'installation de Gzip

Dernière vérification effectuée auprès de la version 1.2.4a.

Bash: sh Binutils: as, ld, nm Fileutils: chmod, cp, install, ln, mv, rm
Gcc: cc1, collect2, cpp, cpp0, gcc Grep: egrep, grep Make: make Sed: sed Sh−utils: hostname
Textutils: cat, tr

# Kbd

## Site officiel de téléchargement

Kbd (1.06): ftp://ftp.win.tue.nl/pub/linux−local/utils/kbd/ Correctif Kbd (1.06−3): ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/ http://ftp.linuxfromscratch.org/lfs−packages/4.0/

## Contenu de Kbd

Dernière vérification effectuée auprès de la version 1.06.

### Programmes

chvt, deallocvt, dumpkeys, fgconsole, getkeycodes, getunimap, kbd_mode, kbdrate, loadkeys, loadunimap, mapscrn, openvt, psfaddtable (lien vers psfxtable), psfgettable (lien vers psfxtable), psfstriptable (lien vers psfxtable), psfxtable, resizecons, setfont, setkeycodes, setleds, setlogcons, setmetamode, setvesablank, showfont, showkey, unicode_start, et unicode_stop

### Descriptions

#### chvt

chvt change le terminal virtuel de premier plan.

#### deallocvt

deallocvt désalloue les terminaux virtuels inutilisés.

#### dumpkeys

dumpkeys affiche les tables de conversion du clavier.

#### fgconsole

fgconsole affiche le numéro du terminal virtuel actif.

#### getkeycodes

getkeycodes affiche la table de correspondance de scancode en keycode du noyau.

#### getunimap

getunimap affiche le unimap actuellement utilisé.

#### kbd_mode

kbd_mode indique ou positionne le mode du clavier.

### kbdrate

kbdrate positionne les taux de répétition et de temporisation du clavier.

### loadkeys

loadkeys charge les tables de conversion du clavier.

### loadunimap

loadunimap charge la table de correspondance d'unicode en police du noyau.

### mapscrn

mapscrn charge une table de correspondance des caractères de sortie définie par l'utilisateur dans le pilote de la console. Notez qu'il est obsolète et que ses fonctionnalités sont reprises dans setfont.

### openvt

openvt lance un programme sur un nouveau terminal virtuel (VT).

### psfaddtable, psfgettable, psfstriptable, psfxtable

C'est un ensemble d'outils pour gérer les tables de caractères Unicode pour les polices de console.

### resizecons

resizeicons modifie la représentation de la taille de la console pour le noyau.

### setfont

Ceci vous permet de modifier les polices EGA/VGA dans la console.

### setkeycodes

setkeycodes charge les entrées de la table de correspondance scancode en keycode du noyau.

### setleds

setleds positionne les LED du clavier. Beaucoup de personnes trouvent utile d'avoir numlock activé par défaut, et, en utilisant ce programme, vous pourrez y arriver.

### setlogcons

setlogcons sends kernel messages to the console.

### setmetamode

setmetamode définit la gestion de la touche méta du clavier.

### setvesablank

Ceci vous laisse trafiquer l'économiseur d'écran matériel (pas de grilles−pain, seulement un écran vide).

### showfont

showfont affiche des données sur une police. L'information montrée comprend les informations sur la police, ses propriétés, les métriques des caractères, et les images des caractères.

### showkey

showkey examine les scancodes et les keycodes transmis par le clavier.

### unicode_start

unicode_start met la console en mode Unicode.

### unicode_stop

unicode_stop annule le mode unicode pour le clavier et la console.

## Dépendances d'installation de Kbd

Dernière vérification effectuée auprès de la version 1.06.

Bash: sh Binutils: as, ld, strip Bison: bison Diffutils: cmp Fileutils: cp, install, ln, mv, rm Flex: flex Gettext: msgfmt, xgettext Gcc: cc1, collect2, cpp0, gcc Grep: grep Gzip: gunzip, gzip Make: make Patch: patch Sed: sed Sh−utils: uname

# Less

## Site officiel de téléchargement

Less (374):  ftp://ftp.gnu.org/gnu/less/

## Contenu de Less

Dernière vérification effectuée auprès de la version 374.

### Programmes

less, lessecho et lesskey

### Description

#### less

Le programme less est un paginateur de fichier (ou un afficheur de texte). Il affiche le contenu d'un fichier et permet le défilement du texte. Less est une amélioration du paginateur habituel appelé << more>> . Less permet également le défilement arrière dans les fichiers et n'a pas besoin de lire tout le fichier quand il démarre, ce qui le rend plus rapide lors de la lecture de gros fichiers.

#### lessecho

lessecho est utile pour étendre les méta caractères, comme * et ?, dans les noms de fichiers sur les systèmes Unix.

#### lesskey

lesskey est utilisé pour spécifier les associations de touches à less.

## Dépendances d'installation de Less

Dernière vérification effectuée auprès de la version 358.

Bash: sh Binutils: as, ld Diffutils: cmp Fileutils: chmod, install, mv, rm, touch Grep: egrep, grep
Gcc: cc1, collect2, cpp0, gcc Make: make Sed: sed Sh−utils: expr, hostname, uname Textutils: cat, tr

# LFS−Bootscripts

## Site officiel de

LFS−Bootscripts (1.10):
ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/
http://ftp.linuxfromscratch.org/lfs−packages/4.0/http://ftp.linuxfromscratch.org/lfs−packages/4.0/

## Contenu de LFS−bootscripts

Dernière vérification effectuée auprès de la version 1.10.

### Scripts

checkfs, cleanfs, functions, halt, ifdown, ifup, loadkeys, localnet, mountfs, mountproc, network, rc, reboot, sendsignals, setclock, swap, sysklogd et template

### Descriptions

#### checkfs

Le script checkfs vérifie les systèmes de fichiers un peu avant qu'ils ne soient montés à l'exception des systèmes de fichiers journalisés ou réseau).

#### cleanfs

Le script cleanfs supprime tous les fichiers qui ne devraient pas être conservés entre les redémarrages, fichiers tels que /var/run/*, /var/lock/*. Il recrée /var/run/utmp et supprime les éventuels fichiers /etc/nologin, /fastboot et /forcefsck.

#### functions

Le script functions contient les fonctions partagées entre les différents scripts, telles que la vérification d'erreur, la vérification de l'état, etc...

#### halt

Le script halt arrête le système.

#### ifdown, ifup

Les scripts ifdown et ifup assistent le script network avec les périphériques réseaux.

#### loadkeys

Le script loadkeys charge le plan de clavier, correspondant à votre clavier.

### localnet

Le script localnet initialise le nom de l'hôte ainsi que le périphérique loopback.

### mountfs

Le script mountfs monte tous les systèmes de fichiers qui ne sont pas indiqués noauto et qui ne sont pas réseau.

### mountproc

Le script mountproc est utilisé pour monter le système de fichiers proc.

### network

Le script network met en place les interfaces réseau (telles que les cartes réseau) et initialise la passerelle par défaut si nécessaire.

### rc

Le script rc est le script maître de contrôle du niveau d'exécution, responsable pour lancer tous les autres scripts un par un dans une séquence spécifiée.

### reboot

Le script reboot relance le système.

### sendsignals

Le script sendsignals s'assure que chaque processus est bien arrêté avant que le système ne s'arrête ou ne redémarre la machine.

### setclock

Le script setclock réinitialise l'horloge du noyau à l'heure locale au cas où l'horloge matérielle n'est pas positionnée sur l'heure GMT.

### swap

Le script swap active et désactive les fichiers et partitions de swap.

### sysklogd

Le script sysklogd lance et arrête les démons système et kernel de log.

### template

Le script template est un modèle que vous pouvez utiliser pour créer vos propres scripts de démarrage pour d'autres démons.

# Dépendances d'installation de Bootscripts

Dernière vérification effectuée auprès de la version 1.10.

Fileutils: chown, cp

# Libtool

## Site officiel de téléchargement

Libtool (1.4.2): ftp://ftp.gnu.org/gnu/libtool/

## Contenu de libtool

Dernière vérification effectuée auprès de la version 1.4.2.

### Programmes

libtool et libtoolize

### Descriptions

#### libtool

libtool fournit des services généraux de support de construction de bibliothèques.

#### libtoolize

libtoolize fournit un moyen standard d'ajouter un support libtool à un package.

### Bibliothèques

libltdl.a, libltdl.so (lien vers libltdl.so.3.1.0), libltdl.so.3 (lien vers libltdl.so.3.1.0) et libltdl.so.3.1.0

### Descriptions

#### libltdl, libltdl.so.3, libltdl.so.3.1.0

libltdl est une petite bibliothèque qui a pour but de masquer aux programmeurs les difficultés variées liées à l'ouverture dynamique des bibliothèques.

## Dépendances d'installation de Libtool

Dernière vérification effectuée auprès de la version 1.4.2.

Bash: sh Binutils: ar, as, ld, nm, ranlib, strip Diffutils: cmp
Fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm, rmdir Gcc: cc, cc1, collect2, cpp0
Glibc: ldconfig Grep: egrep, fgrep, grep Make: make Sed: sed
Sh−utils: echo, expr, hostname, sleep, uname Texinfo: install−info Textutils: cat, sort, tr, uniq

# Lilo

## Site officiel de téléchargement

Lilo (22.2): ftp://ibiblio.org/pub/Linux/system/boot/lilo/ http://ibiblio.org/pub/Linux/system/boot/lilo/

## Contenu de Lilo

Dernière vérification effectuée auprès de la version 22.2.

### Programmes

lilo, mkrescue et keytab−lilo.pl

### Descriptions

#### lilo

lilo installe le chargeur de boot de Linux, qui est utilisé pour démarrer un système Linux.

#### mkrescue

mkrescue crée une disquette bootable de secours en utilisant un noyau existant et un ramdisk initial.

#### keytab−lilo.pl

keytab−lilo.pl compile les tables de définition de codage de clavier dans un format que lilo peut utiliser pour initialiser le type de clavier lors du démarrage.

## Dépendances d'installation de Lilo

Dernière vérification effectuée auprès de la version 22.1.

Bash: sh Bin86: as86, ld86 Binutils: as, ld, strip Fileutils: cp, dd, ln Gcc: cc, cc1, collect2, cpp0
Make: make Sed: sed Textutils: cat

# Linux (le noyau)

## Site officiel de téléchargement

Linux (2.4.19):  ftp://ftp.kernel.org/pub/linux/kernel/

## Contenu de Kernel

Dernière vérification effectuée auprès de la version 2.4.18.

### Programmes

Le noyau linux et les entêtes du noyau linux

### Descriptions

#### Noyau linux

Le noyau Linux est au coeur de chaque système Linux. C'est lui qui fait tourner Linux. Quand vous allumez votre ordinateur et démarrez un système Linux, la toute première partie de logiciel Linux qui est chargée est le noyau. Le noyau initialise les composants matériels du système comme les ports série, les ports parallèles, les cartes son, les cartes réseau, les contrôleurs IDE, les contrôleurs SCSI et beaucoup d'autres choses. En bref, le noyau rend le matériel disponible pour que le logiciel puisse fonctionner.

#### En−têtes du noyau linux

Nous copions ces fichiers dans /usr/include/{linux,asm} lors du chapitre 5. Ils doivent correspondre à ceux avec lesquels glibc a été compilé et ne doivent *pas* être remplacé lors d'une mise à jour du kernel. Ils sont essentiels pour compiler un grand nombre de logiciels.

## Dépendances d'installation de Linux

Dernière vérification effectuée auprès de la version 2.4.17.

Bash: sh Binutils: ar, as, ld, nm, objcopy Fileutils: cp, ln, mkdir, mv, rm, touch
Findutils: find, xargs Gcc: cc1, collect2, cpp0, gcc Grep: grep Gzip: gzip Make: make Gawk: awk
Modutils: depmod, genksyms Net−tools: dnsdomainname, hostname Sed: sed
Sh−utils: basename, date, expr, pwd, stty, uname, whoami, yes Textutils: cat, md5sum, sort, tail

# M4

## Site officiel de téléchargement

M4 (1.4):  ftp://ftp.gnu.org/gnu/m4/

## Contenu de M4

Dernière vérification effectuée auprès de la version 1.4.

### Programmes

m4

### Descriptions

#### m4

m4 est un processeur de macro. Il copie l'entrée sur la sortie en développant les macros au fur et à mesure. Les macros sont soit prédéfinies, soit définies par l'utilisateur, et peuvent prendre un nombre quelconque d'arguments. En plus de développer les macros, m4 a des fonctions prédéfinies pour inclure des fichiers nommés, lancer des commandes Unix, faire de l'arithmétique d'entiers, manipuler du texte de diverses manières, faire de la récursivité, etc. m4 peut être utilisé soit comme frontal d'un compilateur, ou comme processeur de macros à part.

## Dépendances d'installation de M4

Dernière vérification effectuée auprès de la version 1.4.

Bash: sh Binutils: ar, as, ld, ranlib Diffutils: cmp Fileutils: chmod, cp, install, mv, rm Make: make
Gcc: cc1, collect2, cpp0, gcc Grep: egrep, grep Sed: sed Sh−utils: date, echo, hostname
Textutils: cat, tr

# Make

## Site officiel de téléchargement

Make (3.79.1):  ftp://ftp.gnu.org/gnu/make/

## Contenu de Make

Dernière vérification effectuée auprès de la version 3.79.1.

### Programmes

make

### Descriptions

#### make

make détermine automatiquement quelles parties d'un gros programme ont besoin d'être recompilées, et exécute les commandes pour les recompiler.

## Dépendances d'installation de Make

Dernière vérification effectuée auprès de la version 3.79.1.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh Binutils: as, ld Diffutils: cmp
Fileutils: chgrp, chmod, install, ls, mv, rm Gcc: cc, cc1, collect2, cpp0, gcc Glibc: getconf
Grep: egrep, fgrep, grep M4: m4 Make: make Gawk: gawk Sed: sed
Sh−utils: basename, echo, expr, hostname, sleep, uname Texinfo: install−info, makeinfo
Textutils: cat, tr

# MAKEDEV

## Site officiel de téléchargement

MAKEDEV (1.7): ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/
http://ftp.linuxfromscratch.org/lfs−packages/4.0/

## Contenu de MAKEDEV

Dernière vérification effectuée auprès de la version 1.5.

### Programmes

MAKEDEV

### Descriptions

#### MAKEDEV

`MAKEDEV` est un script qui crée les fichiers périphériques statiques nécessaires, qui se trouvent généralement dans le répertoire `/dev`. Plus d'informations sur les fichiers périphériques sont disponibles à partir de la hiérarchie des sources du noyau dans le fichier `Documentation/devices.txt`.

## Dépendances d'installation de MAKEDEV

Dernière vérification effectuée auprès de la version 1.5.

Bash: sh Fileutils: chmod, chown, cp, ln, mknod, mv, rm Grep: grep Sh−utils: expr, id

# Man

## Site officiel de téléchargement

Man (1.5k):  ftp://ftp.win.tue.nl/pub/linux−local/utils/man/ Correctif Man (1.5k): ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/ http://ftp.linuxfromscratch.org/lfs−packages/4.0/

## Contenu de Man

Dernière vérification effectuée auprès de la version 1.5k.

### Programmes

apropos, makewhatis, man, man2dvi, man2html et whatis

### Descriptions

#### apropos

apropos recherche des mots clés dans un ensemble de fichiers base de données contenant une description brève des commandes système et affiche le résultat sur la sortie standard.

#### makewhatis

makewhatis lit toutes les pages de manuel contenues dans les sections de manpath fournies, ou les pages préformatées contenues dans les sections de catpath données. Pour chaque page, il écrit une ligne dans la base de données whatis. Chaque ligne contient le nom de la page et une brève description, séparés par un tiret. La description est extraite du contenu de la section NAME de la page du manuel.

#### man

man formate et affiche les pages de manuel en ligne.

#### man2dvi

man2dvi exporte une page man au format dvi.

#### man2html

man2html exporte une page man au format html.

#### whatis

whatis recherche des mots clés dans un ensemble de fichiers base de données contenant une description brève des commandes système et affiche le résultat sur la sortie standard. Seules les correspondances de mots complets sont affichées.

# Dépendances d'installation de Man

Dernière vérification effectuée auprès de la version 1.5i2.

Bash: sh Binutils: as, ld Fileutils: chmod, cp, install, mkdir, rm Gcc: c11, collect2, cpp0, gcc
Grep: grep Make: make Gawk: awk Sed: sed Sh−utils: echo Textutils: cat

# Man−pages

## Site officiel de téléchargement

Man−pages (1.52):  ftp://ftp.kernel.org/pub/linux/docs/manpages/

## Contents of Man−pages

Dernière vérification effectuée auprès de la version 1.52.

### Fichiers support

Plusieurs pages de manuel qui ne viennent pas avec les packages.

### Descriptions

#### manual pages

Les exemples de pages de manuel fournis décrivent toutes les fonctions C et C++, quelques fichiers /dev/ importants et autres.

## Dépendances d'installation de Man−pages

Dernière vérification effectuée auprès de la version 1.47.

Bash: sh Fileutils: install Make: make

# Modutils

## Site officiel de téléchargement

Modutils (2.4.19):  ftp://ftp.kernel.org/pub/linux/utils/kernel/modutils/

## Contenu de Modutils

Dernière vérification effectuée auprès de la version 2.4.16.

### Programmes

depmod, genksyms, insmod, insmod_ksymoops_clean, kallsyms (lien vers insmod), kernelversion, ksyms (lien vers insmod), lsmod (lien vers insmod), modinfo, modprobe (lien vers insmod) et rmmod (lien vers insmod)

### Descriptions

#### depmod

depmod gère la description des dépendances entre les modules du noyau.

#### genksyms

genksyms examine (sur l'entrée standard) la sortie de gcc −E source.c et génère un fichier contenant des informations de version.

#### insmod

insmod installe un module chargeable dans le noyau en cours d'utilisation.

#### insmod_ksymoops_clean

insmod_ksymoops_clean efface les fichiers ksyms et les modules sauvegardés n'ayant pas été accédés depuis 2 jours.

#### kallsyms

kallsyms extrait tous les symboles pour le débuggage du noyau.

#### kernelversion

kernelversion affiche la version majeure du noyau en cours d'utilisation.

#### ksyms

ksyms affiche les symboles exportés du noyau.

### lsmod

lsmod affiche des informations sur les modules chargés.

### modinfo

modinfo examine un fichier objet associé à un module noyau pour afficher toutes les informations qu'il peut glaner.

### modprobe

modprobe utilise un fichier de dépendances du style de make, créé par depmod, pour automatiquement charger les modules associés situés dans les arborescences prévues à cet effet.

### rmmod

rmmod décharge les modules chargeables du noyau en cours d'utilisation.

## Dépendances d'installation de Modutils

Dernière vérification effectuée auprès de la version 2.4.12.

Bash: sh Binutils: ar, as, ld, ranlib, strip Bison: bison Diffutils: cmp
Fileutils: chmod, install, ln, mkdir, mv, rm Flex: flex Gcc: cc, cc1, collect2, cpp0, gcc
Grep: egrep, grep Make: make Sed: sed Sh−utils: basename, expr, hostname, uname Textutils: cat, tr

# Ncurses

## Site officiel de téléchargement

Ncurses (5.2):  ftp://ftp.gnu.org/gnu/ncurses/ Correctif Ncurses (5.2): ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/ http://ftp.linuxfromscratch.org/lfs−packages/4.0/

## Contenu de Ncurses

Dernière vérification effectuée auprès de la version 5.2.

### Programmes

captoinfo (lien vers tic), clear, infocmp, infotocap (lien vers tic), reset (lien vers tset), tack, tic, toe, tput et tset.

### Descriptions

#### captoinfo

captoinfo convertit une description termcap en une description au format terminfo.

#### clear

Le programme clear efface l'écran si cela est possible. Il utilise les variables d'environnement pour déterminer le type de terminal puis se renseigne dans la base de données terminfo pour trouver les commandes d'effacement de l'écran.

#### infocmp

Le programme infocmp peut être utilisé pour comparer deux entrées binaires de terminfo, ré−écrire une description de terminfo avec l'avantage du champ terminfo use=, ou afficher une description terminfo depuis le fichier binaire (term) dans une variété de formats (fonctionnement inverse de tic).

#### infotocap

infotocap convertit une description terminfo en une description au format termcap.

#### reset

reset met en place les modes cooked et echo, désactive les modes cbreak et raw, active la prise en compte des retours chariot et réinitialise tous les caractères spéciaux non mis en place à leur valeurs par défaut avant d'initialiser le terminal de la même façon que tset.

#### tack

tack vérifie les actions de terminfo.

### tic

tic est le compilateur des entrées de description de terminfo. Le programme traduit un fichier terminfo d'un format source en format binaire pour être utilisé par les routines de la bibliothèque ncurses. Les fichiers terminfo contiennent la description des possibilités d'un terminal.

### toe

Le programme toe liste tous les types de terminaux disponibles selon leur nom primaire avec pour chacun leur description.

### tput

Le programme tput utilise la base de données terminfo pour créer les valeurs associées aux capacités d'un terminal et les informations disponibles pour le shell, pour initialiser un terminal, ou retourner le nom complet d'un type du terminal requis.

### tset

Le programme tset initialise les terminaux de façon à ce qu'ils puissent être utilisés, mais cette fonctionnalité n'est plus tellement utilisée. Elle est fourni pour une compatibilité avec BSD4.4.

## Bibliothèques

libcurses.[a,so] (lien vers libncurses.[a,so]), libform.[a,so], libform_g.a, libmenu.[a,so], libmenu_g.a, libncurses++.a, libncurses.[a,so], libncurses_g.a, libpanel.[a,so] et libpanel_g.a

### libcurses, libncurses++, libncurses, libncurses_g

Les bibliothèques libncurses sont la base du système et sont utilisées pour afficher du texte à l'écran. Un exemple où ncurses est utilisé est dans le programme << make menuconfig>> .

### libform, libform_g

libform est utilisé pour implémenter des formes dans ncurses.

### libmenu, libmenu_g

libmenu est utilisé pour implémenter les menus dans ncurses.

### libpanel, libpanel_g

libpanel est utilisé pour implémenter des panneaux dans ncurses.

# Dépendances d'installation de Ncurses

Dernière vérification effectuée auprès de la version 5.2.

Bash: sh Binutils: ar, as, ld, ranlib Diffutils: cmp Fileutils: chmod, cp, install, ln, mkdir, mv, rm
Gcc: c++, cc1, cc1plus, collect2, cpp0, gcc Glibc: ldconfig Grep: egrep, fgrep, grep Make: make

Gawk: gawk Sed: sed Sh−utils: basename, date, echo, expr, hostname, uname
Textutils: cat, sort, tr, wc

# Netkit−base

## Site officiel de téléchargement

Netkit−base (0.17):  ftp://ftp.uk.linux.org/pub/linux/Networking/netkit/

## Contenu de Netkit−base

Dernière vérification effectuée auprès de la version 0.17.

### Program Files

inetd et ping

### Descriptions

#### inetd

inetd est le père de tous les démons. Il est à l'écoute des connections, et transfère l'appel au démon approprié.

#### ping

ping envoie des paquets ICMP ECHO_REQUEST à un hôte et détermine son temps de réponse.

## Dépendances d'installation de Netkit−base

Dernière vérification effectuée auprès de la version 0.17.

Bash: sh Binutils: as, ld, strip Fileutils: cp, install, rm Make: make Gcc: cc1, collect2, cpp0, gcc
Sed: sed Sh−utils: date Textutils: cat

# Net−tools

## Site officiel de téléchargement

Net−tools (1.60):  http://www.tazenda.demon.co.uk/phil/net−tools/

## Contenu de Net−tools

Dernière vérification effectuée auprès de la version 1.60.

### Programmes

arp, dnsdomainname (lien vers hostname), domainname (lien vers hostname), hostname, ifconfig, nameif, netstat, nisdomainname (lien vers hostname), plipconfig, rarp, route, slattach et ypdomainname (lien vers hostname)

### Descriptions

#### arp

arp manipule le cache ARP du noyau, habituellement pour ajouter ou supprimer une entrée, ou affiche le contenu du cache ARP.

#### dnsdomainname

dnsdomainname affiche le nom de domaine DNS du système.

#### domainname

domainname affiche ou enregistre le nom de domaine NIS/YP du système.

#### hostname

hostname est utilisé pour enregistrer ou afficher le nom d'hôte du système.

#### ifconfig

ifconfig est la commande principale pour configurer les interfaces réseau.

#### nameif

nameif nomme les interfaces réseau, basées sur les adresses MAC.

#### netstat

netstat est un utilitaire ayant de nombreux usages; il affiche les connections réseau, les tables de routage, les statistiques des interfaces, les connections masquées et les membres multicast.

### nisdomainname

nisdomainname affiche ou enregistre le nom de domaine NIS/YP du système.

### plipconfig

plipconfig permet de règler finement les paramètres d'un périphérique PLIP. En étant optimiste, il le rend plus rapide.

### rarp

Apparenté au programme arp, rarp manipule la table RARP du système.

### route

route est l'utilitaire utilisé pour manipuler les tables de routage IP.

### slattach

slattach attache une interface réseau à une ligne série, c'est−à−dire qu'il positionne une ligne normale pour un terminal dans un des nombreux modes "réseau".

### ypdomainname

updomainname affiche ou enregistre le nom de domaine NIS/YP du système.

## Dépendances d'installation de Net−tools

Dernière vérification effectuée auprès de la version 1.60.

Bash: bash, sh Binutils: ar, as, ld Fileutils: install, ln, ls, mv, rm Gcc: cc, cc1, collect2, cpp0
Make: make Sh−utils: echo

# Patch

## Site officiel de téléchargement

Patch (2.5.4):  ftp://ftp.gnu.org/gnu/patch/

## Contenu de Patch

Dernière vérification effectuée auprès de la version 2.5.4.

### Programmes

patch

### Descriptions

#### patch

Le programme patch modifie un fichier en fonction d'un fichier de patch, aussi appelé correctif. Ce correctif est habituellement une liste créée par le programme diff et contient les instructions pour modifier le fichier original. Patch est très souvent utilisé pour la modification de code source car il fait économiser du temps et de la taille mémoire. Imaginez un package faisant 1 Mo. La version suivante de ce package n'a modifié que deux fichiers de la précédente version. On peut alors le considérer entièrement comme un nouveau package d'1 Mo ou juste sous la forme d'un correctif d'1 Ko qui viendra mettre à jour la précédente version pour la rendre identique à la dernière version réalisée. Ainsi, si ma première version avait déjà été téléchargée, un correctif évite un deuxième téléchargement conséquent.

## Dépendances d'installation de Patch

Dernière vérification effectuée auprès de la version 2.5.4.

Bash: sh Binutils: as, ld Diffutils: cmp Fileutils: chmod, install, mv, rm
Gcc: cc, cc1, collect2, cpp0, gcc Glibc: getconf Grep: egrep, grep Make: make Sed: sed
Sh−utils: echo, expr, hostname, uname Textutils: cat, tr

# Perl

## Site officiel de téléchargement

Perl (5.8.0): http://www.perl.com/

## Contenu de Perl

Dernière vérification effectuée auprès de la version 5.6.1.

### Programmes

a2p, c2ph, dprofpp, find2perl, h2ph, h2xs, perl, perl5.6.1, perlbug, perlcc, perldoc, pl2pm, pod2html, pod2latex, pod2man, pod2text, pod2usage, podchecker, podselect, pstruct, s2p et splain

### Descriptions

#### a2p

a2p est un traducteur perl vers awk.

#### c2ph

c2ph affiche les strutures C telles que générées avec "cc −g −S".

#### dprofpp

dprofpp affiche les données profile de perl.

#### find2perl

find2perl traduit les lignes de commande find en code Perl.

#### h2ph

h2ph convertit les fichiers d'entêtes .h C en fichiers d'entêtes .ph Perl.

#### h2xs

h2xs convertit les fichiers d'entêtes .h C en extensions Perl.

#### perl, perl5.6.1

perl est l'acronyme de Practical Extraction and Report Language (Langage pratique d'extraction et de rapport). Il combine certaines des meilleurs qualités de C, sed, awk et sh dans un puissant langage.

### perlbug

perlbug aide à générer des rapports de bugs sur perl ou sur un des modules qui sont inclus, et les envoie par mail.

### perlcc

perlcc génère des exécutables à partir de programmes Perl.

### perldoc

perldoc recherche une documentation au format .pod, intégré dans l'arbre d'installation de Perl ou dans un script perl, et l'affiche via "pod2man | nroff −man | $PAGER".

### pl2pm

pl2pm est un outil aidant à la conversion des bibliothèques style Perl4 en des modules de bibliothèques style Perl5.

### pod2html

pod2html convertit en HTML des fichiers au format pod.

### pod2latex

pod2latex convertit en LaTeX des fichiers au format pod.

### pod2man

pod2man convertit en *roff des fichiers au format pod.

### pod2text

pod2text convertit en ASCII des fichiers au format pod.

### pod2usage

pod2usage affiche les messages d'usage à partir des documentations intégrées.

### podchecker

podchecker vérifie la syntaxe des fichiers de documentation pod.

### podselect

podselect envoie sur la sortie standard les sections sélectionnées d'une documentation pod.

### pstruct

pstruct affiche les structures C telles que générées à partir de la commande "cc −g −S".

### s2p

s2p est un traducteur sed vers perl.

### splain

splain est un programme qui force de manière importante les messages de diagnostique avec Perl.

## Bibliothèques

attrs.so, B.so, ByteLoader.so, DProf.so, Dumper.so, DynaLoader.a, Fcntl.so, Glob.so, Hostname.so, IO.so, libperl.a, Opcode.so, Peek.so, POSIX.so, re.so, SDBM_File.so, Socket.so, Syslog.so et SysV.so

## Descriptions

### attrs

Aucune description disponible.

### B

Aucune description disponible.

### ByteLoader

Aucune description disponible.

### DProf

Aucune description disponible.

### Dumper

Aucune description disponible.

### DynaLoader

Aucune description disponible.

### Fcntl

Aucune description disponible.

### Glob

Aucune description disponible.

### Hostname

Aucune description disponible.

### IO

Aucune description disponible.

### libperl

Aucune description disponible.

### Opcode

Aucune description disponible.

### Peek

Aucune description disponible.

### POSIX

Aucune description disponible.

### re

Aucune description disponible.

### SDBM_File

Aucune description disponible.

### Socket

Aucune description disponible.

### Syslog

Aucune description disponible.

### SysV

Aucune description disponible.

## Dépendances d'installation de Perl

Dernière vérification effectuée auprès de la version 5.6.1.

Bash: sh Binutils: ar, as, ld, nm Diffutils: cmp Fileutils: chmod, cp, ln, ls, mkdir, mv, rm, touch Gcc: cc, cc1, collect2, cpp0, gcc Grep: egrep, grep Make: make Gawk: awk Sed: sed

Sh−utils: basename, date, echo, expr, hostname, pwd, uname, whoami
Textutils: cat, comm, sort, split, tr, uniq, wc

# Procinfo

## Site officiel de téléchargement

Procinfo (18): ftp://ftp.cistron.nl/pub/people/svm/

## Contenu de Procinfo

Dernière vérification effectuée auprès de la version 18.

### Programmes

lsdev, procinfo et socklist

### Descriptions

#### lsdev

lsdev récupère des informations sur le matériel installé dans votre ordinateur à partir des fichiers sur les interruptions, ports d'entrées/sorties et ports dma dans le répertoire /proc, vous donnant un rapide aperçu de qui utilise quoi (quel matériel utilise tel adresse d'entrée/sortie et tel IRQ et tels canaux DMA).

#### procinfo

procinfo récupère des informations à partir du répertoire /proc et les affiche proprement formatté sur la sortie standard.

#### socklist

Il s'agit d'un script Perl vous donnant une liste de tous les sockets ouverts, énumérant leurs types, port, inode, identifiant utilisateur, numéro de processus, identifiant du fichier ainsi que le programme à qui ils appartiennent.

## Dépendances d'installation de Procinfo

Dernière vérification effectuée auprès de la version 18.

Binutils: as, ld Fileutils: install, mkdir Gcc: cc1, collect2, cpp0, gcc Make: make

# Procps

## Site officiel de téléchargement

Procps (2.0.7): ftp://people.redhat.com/johnsonm/procps/ Correctif Procps (2.0.7): ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/ http://ftp.linuxfromscratch.org/lfs−packages/4.0/

## Contenu de Procps

Dernière vérification effectuée auprès de la version 2.0.7.

### Programmes

free, kill, oldps, pgrep, pkill, ps, skill, snice, sysctl, tload, top, vmstat, w et watch

### Descriptions

#### free

free affiche le nombre total de mémoire, physique ou swap, disponible ou utilisée dans le système. Il affiche également la mémoire partagée et les tampons utilisés par le noyau.

#### kill

kill envoie des signaux aux processus.

#### oldps et ps

ps donne un état des processus en cours.

#### pgrep

pgrep cherche les processus à partir de leur nom et d'autres attributs.

#### pkill

pkill envoie des signaux en se basant sur leur nom et d'autres attributs.

#### skill

skill envoie des signaux aux processus remplissant certains critères.

#### snice

snice change la priorité d'ordonnancement des processus remplissant certains critères.

### sysctl

sysctl modifie les paramètres noyau au lancement.

### tload

tload affiche un graphique de la charge moyenne du système en cours au terminal spécifié (ou au tty du processus tload si aucun n'est spécifié).

### top

top affiche l'activité du processeur en temps réel.

### vmstat

vmstat fournit des informations sur les processus, la mémoire, la pagination, les entrées−sorties, les interruptions et l'activité du processeur.

### w

w affiche des informations sur les utilisateurs actuellement connectés, et leurs processus.

### watch

watch exécute une commande répétitivement, en affichant ses sorties (au premier écran plein).

## Bibliothèques

libproc.so

## Descriptions

### libproc

libproc est une bibliothèque avec laquelle la plupart des programmes de ce package ont été lié pour sauver de l'espace disque, en implémentant des fonctions communes une seule fois.

# Dépendances d'installation de Procps

Dernière vérification effectuée auprès de la version 2.0.7.

Bash: sh Binutils: as, ld, strip Fileutils: install, ln, mv, rm Gcc: cc1, collect2, cpp0, gcc Grep: grep
Make: make Gawk: awk Sed: sed Sh−utils: basename, pwd Textutils: sort, tr

# Psmisc

## Site officiel de téléchargement

Psmisc (21): http://download.sourceforge.net/psmisc/
ftp://download.sourceforge.net/pub/sourceforge/psmisc/

## Contenu de Psmisc

Dernière vérification effectuée auprès de la version 21.

### Programmes

fuser, killall et pstree

A noter que dans LFS nous n'installons pas le lien pidof par défaut car nous utilisons pidof de sysvinit à la place.

### Descriptions

#### fuser

fuser affiche les PIDs des processus qui utilisent les fichiers ou les systèmes de fichiers spécifiés.

#### killall

killall envoie un signal à tous les processus faisant tourner l'une des commandes spécifiées.

#### pstree

pstree présente les processus tournant sous forme d'arborescence.

## Dépendances d'installation de Psmisc

Dernière vérification effectuée auprès de la version 20.2.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh Bison: bison Binutils: as, ld
Diffutils: cmp Fileutils: chmod, install, ls, mkdir, mv, rm Gettext: msgfmt, xgettext
Gcc: cc, cc1, collect2, cpp0, gcc Grep: egrep, grep M4: m4 Make: make Gawk: gawk Sed: sed
Sh−utils: basename, echo, expr, hostname, sleep, uname Texinfo: makeinfo Textutils: cat, tr

# Sed

## Site officiel de téléchargement

Sed (3.02): ftp://ftp.gnu.org/gnu/sed/

## Contenu de Sed

Dernière vérification effectuée auprès de la version 3.02.

### Programmes

sed

### Descriptions

#### sed

sed est un éditeur de flux. Un éditeur de flux est utilisé pour exécuter des transformations de texte basiques sur un flux d'entrée (un fichier ou l'entrée reçue d'un tuyau).

## Dépendances d'installation de Sed

Dernière vérification effectuée auprès de la version 3.02.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh Binutils: ar, as, ld, ranlib
Diffutils: cmp Fileutils: chmod, install, ls, mv, rm Gcc: cc1, collect2, cpp0, gcc Glibc: getconf
Grep: egrep, fgrep, grep M4: m4 Make: make Gawk: gawk Sed: sed
Sh−utils: echo, expr, hostname, sleep Texinfo: install−info, makeinfo Textutils: cat, tr

# Shadow

## Site officiel de téléchargement

Shadow (4.0.3): ftp://ftp.pld.org.pl/software/shadow/

## Contenu de Shadow

Dernière vérification effectuée auprès de la version 4.0.3.

### Programmes

chage, chfn, chpasswd, chsh, dpasswd, expiry, faillog, gpasswd, groupadd, groupdel, groupmod, grpck, grpconv, grpunconv, lastlog, login, logoutd, mkpasswd, newgrp, newusers, passwd, pwck, pwconv, pwunconv, sg (lien vers newgrp), su, useradd, userdel, usermod, vigr (lien vers vipw) et vipw

### Descriptions

#### chage

chage change le nombre de jours entre chaque changement de mot de passe et la date du dernier changement de mot de passe.

#### chfn

chfn change le nom complet de l'utilisateur ainsi que d'autres informations (les coordonnées professionnelles, le numéro de téléphone au bureau et le numéro de téléphone personnel).

#### chpasswd

chpasswd lit des paires nom d'utilisateur/mot de passe depuis l'entrée standard pour mettre à jour un groupe d'utilisateurs déjà existants.

#### chsh

chsh change le shell de login d'un utilisateur.

#### dpasswd

dpasswd ajoute, efface ou met à jour les mots de passe téléphoniques pour les shells de login d'un utilisateur.

#### expiry

expiry vérifie et applique les règles d'expiration des mots de passe.

### faillog

faillog met en forme le contenu du journal d'erreur, /var/log/faillog, et tient à jour le compte et les limites de ces erreurs.

### gpasswd

gpasswd administre le fichier /etc/group.

### groupadd

La commande groupadd crée un nouveau groupe en utilisant les valeurs entrées en ligne de commande et les valeurs par défaut du système.

### groupdel

La commande groupdel modifie les fichiers de gestion des comptes du système en effaçant toutes les entrées se référant au groupe spécifié.

### groupmod

La commande groupmod modifie les fichiers de gestion des comptes du système en fonction des entrées en ligne de commande.

### groups

groups affiche le groupe d'appartenance de l'utilisateur.

### grpck

grpck vérifie l'intégrité des informations du système d'authentification.

### grpconv

grpconv convertit des fichiers de groupes normaux en fichiers de groupe de type shadow.

### grpunconv

grpunconv convertit des fichiers de groupes de type shadow en fichiers de groupes normaux.

### lastlog

lastlog met en forme et affiche le contenu du journal des dernières connections, /var/log/lastlog. Le nom de login, le port et l'heure de dernière connection seront affichés.

### login

login permet d'établir une nouvelle session sur le système.

### logoutd

logoutd applique les restrictions de temps de connection et de port de connection définis dans le fichier /etc/porttime.

### mkpasswd

mkpasswd lit un fichier au format spécifié en option et le convertit dans un format de fichier de base de données correspondant.

### newgrp

newgrp permet de changer l'identifiant du groupe en cours d'utilisation.

### newusers

newusers lit un fichier de noms d'utilisateur, efface leur mot de passe et utilise ces informations pour mettre à jour ou pour créer de nouveaux utilisateurs.

### passwd

passwd change les mots de passe des comptes d'utilisateurs et de groupes.

### pwck

pwck vérifie l'intégrité des fichiers de mots de passe.

### pwconv

pwconv convertit un fichier de mots de passe normal en fichier de mots de passe de type shadow.

### pwunconv

pwunconv convertit un fichier de mots de passe de type shadow en fichier de mots de passe normal.

### sg

sg positionne l'identifiant de groupe de l'utilisateur à celui du groupe donné, ou exécute une commande donnée en tant que membre du groupe spécifié.

### useradd

useradd crée un nouvel utilisateur ou met à jour les informations par défaut d'un utilisateur.

### userdel

userdel modifie les fichiers du système de gestion de comptes en effaçant toutes les entrées relative au nom de login spécifié.

### usermod

usermod modifie les fichiers du système de gestion de comptes en fonction des informations données en ligne de commande.

### vipw and vigr

vipw et vigr éditeront les fichiers /etc/passwd et /etc/group, respectivement. Avec l'option −s, ces deux commandes pourront éditer les versions de type shadow de ces mêmes fichiers, /etc/shadow et /etc/gshadow, respectivement.

## Bibliothèques

libmisc.[a,so], libshadow.[a,so]

## Descriptions

### libmisc

Aucune description encore disponible.

### libshadow

libshadow apporte des fonctions communes aux programmes shadow.

# Dépendances d'installation de Shadow

Dernière vérification effectuée auprès de la version 20001016.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh Binutils: ar, as, ld, nm, ranlib Diffutils: cmp Fileutils: chmod, cp, install, ln, ls, mkdir, mv, rm, rmdir Gettext: msgfmt, xgettext Gcc: cc1, collect2, cpp0, gcc Glibc: ldconfig Grep: egrep, grep M4: m4 Make: make Gawk: gawk Net−tools: hostname Sed: sed Sh−utils: basename, echo, expr, sleep, uname Texinfo: makeinfo Textutils: cat, sort, tr, uniq

# Sh−utils

## Site officiel de téléchargement

Sh−utils (2.0): ftp://ftp.gnu.org/gnu/sh−utils/ Correctif Sh−utils (2.0): ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/ http://ftp.linuxfromscratch.org/lfs−packages/4.0/

## Contenu de Sh−utils

Dernière vérification effectuée auprès de la version 2.0.

### Programmes

basename, chroot, date, dirname, echo, env, expr, factor, false, groups, hostid, id, logname, nice, nohup, pathchk, pinky, printenv, printf, pwd, seq, sleep, stty, su, tee, test, true, tty, uname, uptime, users, who, whoami et yes

### Descriptions

#### basename

basename enlève le répertoire et les suffixes d'un nom de fichier.

#### chroot

chroot exécute une commande ou un shell interactif depuis un répertoire racine particulier.

#### date

date affiche le temps courant dans le format spécifié, ou met à jour la date système.

#### dirname

dirname enlève les suffixes ne faisant pas partie du répertoire d'un nom de fichier.

#### echo

echo affiche une ligne de texte.

#### env

env exécute un programme dans un environnement modifié.

#### expr

expr évalue des expressions.

### factor

factor affiche les facteurs premiers des nombres entiers spécifiés.

### false

false sort toujours avec un code de statut indiquant une erreur.

### groups

groups affiche les groupes auquels appartient l'utilisateur.

### hostid

hostid affiche l'identifiant numérique (en hexadécimal) pour l'hôte courant.

### id

id affiche les identifiants effectifs d'utilisateur et de groupe de l'utilisateur courant ou d'un utilisateur donné.

### logname

logname affiche le nom de login de l'utilisateur.

### nice

nice exécute un programme avec une priorité de scheduling modifiée.

### nohup

nohup lance une commande qui ne tiendra pas compte du signal hangups, avec une sortie vers un fichier de trace.

### pathchk

pathchk vérifie si les noms de fichiers sont valides et portables.

### pinky

pinky est une commande finger allégée qui permet de retrouver des informations sur certains utilisateurs.

### printenv

printenv affiche tout ou partie des variables d'environnement.

### printf

printf formate et affiche des données (identique à la fonction printf en C).

**pwd**

pwd affiche le nom du répertoire de travail courant.

**seq**

seq affiche des nombres dans un certain ordre avec un incrément à spécifier.

**sleep**

sleep stoppe une tâche pendant un certain laps de temps.

**stty**

stty change et affiche les caractéristiques d'une ligne de terminal.

**su**

su exécute un shell en substituant l'identifiant d'utilisateur et de groupe.

**tee**

tee lit l'entrée standard et écrit sur la sortie standard et un fichier.

**test**

test vérifie les types de fichiers et compare les valeurs.

**true**

true sort toujours avec un code de statut indiquant le succès.

**tty**

tty affiche le nom de fichier du terminal connecté à l'entrée standard.

**uname**

uname affiche des informations système.

**uptime**

uptime donne le temps depuis lequel le système fonctionne.

**users**

users affiche les noms des utilisateurs actuellement connectés sur la machine hôte.

### who

who affiche qui est connecté.

### whoami

whoami affiche le nom de l'utilisateur associé à l'identifiant utilisateur effectif courant.

### yes

yes affiche 'y' ou une chaîne de caractères donnée répétitivement jusqu'à ce que la tâche soit tuée.

## Dépendances d'installation de Sh−utils

Dernière vérification effectuée auprès de la version 2.0.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh Binutils: ar, as, ld, ranlib
Diffutils: cmp Fileutils: chmod, chown, install, ls, mv, rm Gettext: msgfmt, xgettext
Gcc: cc, cc1, collect2, cpp0, gcc Glibc: getconf Grep: egrep, fgrep, grep M4: m4 Make: make
Gawk: gawk Perl: perl Sed: sed Sh−utils: basename, echo, expr, hostname, sleep, uname Tar: tar
Texinfo: install−info, makeinfo Textutils: cat, tr

# Sysklogd

## Site officiel de téléchargement

Sysklogd (1.4.1): http://www.infodrom.org/projects/sysklogd/

## Contenu de Sysklogd

Dernière vérification effectuée auprès de la version 1.4.1.

### Programmes

klogd et syslogd

### Descriptions

#### klogd

klogd est un démon système qui intercepte et journalise les messages du noyau linux.

#### syslogd

syslogd fournit un système de journalisation que la plupart des programmes récents utilise. Chaque message journalisé contient au minimum les champs heure et nom de machine, normalement un champ réservé au nom du programme concerné mais cela dépend des options de journalisation de ce programme.

## Dépendances d'installation de Sysklogd

Dernière vérification effectuée auprès de la version 1.4.1.

Binutils: as, ld, strip Fileutils: install Gcc: cc1, collect2, cpp0, gcc Make: make

# Sysvinit

## Site officiel de téléchargement

Sysvinit (2.84): ftp://ftp.cistron.nl/pub/people/miquels/sysvinit/

## Contenu de Sysvinit

Dernière vérification effectuée auprès de la version 2.84.

### Programmes

halt, init, killall5, last, lastb (lien vers last), mesg, pidof (lien vers killall5), poweroff (lien vers halt), reboot (lien vers halt), runlevel, shutdown, sulogin, telinit (lien vers init), utmpdump et wall

### Descriptions

#### halt

halt enregistre la demande d'arrêt du système dans le fichier /var/log/wtmp et envoie ensuite au noyau les commandes halt, reboot ou poweroff. Si halt ou reboot sont appelés alors que le système ne se trouve pas dans un état de fonctionnement 0 ou 6, la commande shutdown sera exécutée (avec l'option −h ou −r).

#### init

init est le père de tous les processus. Son rôle premier est de créer les processus décrits dans le fichier /etc/inittab. Ce fichier contient en général les entrées pour lancer gettys sur toutes les lignes pouvant réceptionner les demandes de connection des utilisateurs. Il contrôle également les processus requis par un système particulier.

#### killall5

killall5 est l'équivalent SystemV de la commande killall. Elle envoie un signal à tous les processus excepté ceux de sa propre session, ainsi elle ne tuera pas le shell qui a lancé le script appelant cette commande.

#### last

last recherche dans le fichier /var/log/wtmp (ou dans le fichier désigné après l'option −f) et affiche une liste des utilisateurs qui se sont connectés (et déconnectés) depuis la création de ce fichier.

#### lastb

lastb est identique à last, sauf que par défaut il affiche le journal du fichier /var/log/btmp, lequel contient tous les essais infructueux de connection.

### mesg

mesg contrôle l'accès au terminal des utilisateurs par les autres utilisateurs. C'est typiquement utilisé pour permettre ou interdire aux autres utilisateurs d'écrire sur son propre terminal.

### pidof

pidof recherche les identifiants de processus (PIDs) des programmes données.

### poweroff

poweroff est l'équivalent de shutdown −h −p now. Il arrête l'ordinateur et l'éteint (avec un BIOS compatible APM et APM validé dans le noyau).

### reboot

reboot est l'équivalent de shutdown −r now. Il redémarre l'ordinateur.

### runlevel

runlevel lit le fichier système utmp (typiquement /var/run/utmp) pour localiser l'enregistrement de l'état de fonctionnement, puis affiche l'état de fonctionnement précédent et celui en cours sur la sortie standard, séparés par un espace.

### shutdown

shutdown arrête le système en toute sécurité. Tous les utilisateurs connectés sont notifiés de l'arrêt, et les nouvelles connections sont bloquées.

### sulogin

sulogin est employé par init lorsque le système démarre en mode single−user (ceci est déterminé dans le fichier /etc/inittab). init essaie aussi d'exécuter sulogin lorsque le chargeur de boot l'appelle avec l'option −b (par exemple LILO).

### telinit

telinit envoie les signaux appropriés à init pour lui spécifier l'état de fonctionnement dans lequel basculer.

### utmpdump

utmpdump affiche le contenu d'un fichier (habituellement /var/run/utmp) sur la sortie standard dans un format explicite pour l'utilisateur.

### wall

wall envoie un message à tous les utilisateurs connectés qui ont positionné leur permission mesg à oui.

# Dépendances d'installation de Sysvinit

Dernière vérification effectuée auprès de la version 2.84.

Bash: sh Binutils: as, ld Fileutils: chown, cp, install, ln, mknod, rm Gcc: cc, cc1, collect2, cpp0
Make: make Sed: sed

# Tar

## Site officiel de téléchargement

Tar (1.13): ftp://ftp.gnu.org/gnu/tar/ Correctif Tar (1.13): ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/ http://ftp.linuxfromscratch.org/lfs−packages/4.0/

## Contenu de Tar

Dernière vérification effectuée auprès de la version 1.13.

### Programmes

rmt et tar

### Descriptions

#### rmt

rmt est un programme utilisé par les programmes distants dump et restore pour manipuler un lecteur de bandes magnétiques via une connection de communication entre processus.

#### tar

tar est un programme de sauvegarde destiné à stocker et à extraire des fichiers depuis un format d'archive connu sous le nom de fichier tar.

## Dépendances d'installation de Tar

Dernière vérification effectuée auprès de la version 1.13.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh Binutils: ar, as, ld, ranlib
Diffutils: cmp Fileutils: chmod, install, ls, mv, rm Gettext: msgfmt, xgettext
Gcc: cc, cc1, collect2, cpp0, gcc Glibc: getconf Grep: egrep, fgrep, grep M4: m4 Make: make
Gawk: gawk Net−tools: hostname Patch: patch Sed: sed Sh−utils: basename, echo, expr, sleep, uname
Texinfo: install−info, makeinfo Textutils: cat, tr

# Texinfo

## Site officiel de téléchargement

Texinfo (4.2): ftp://ftp.gnu.org/gnu/texinfo/

## Contenu de Texinfo

Dernière vérification effectuée auprès de la version 4.2.

### Programmes

info, infokey, install−info, makeinfo, texi2dvi et texindex

### Descriptions

#### info

Le programme info lit les documents de type Info, habituellement stockés dans le répertoire /usr/doc/info. Les documents info sont identiques aux pages de manuel, mais ils tendent à être plus détaillés et n'expliquent pas seulement les différentes options d'un programme.

#### infokey

infokey compile un fichier source contenant des personnalisations Info en un format binaire.

#### install−info

Le programme install−info met à jour les documents info. Lorsque le programme info est invoqué, il présente une liste des sujets disponibles (c'est−à−dire disponibles dans les documents info). Le programme install−info est utilisé pour tenir à jour cette liste. Si des fichiers info sont effacés manuellement, vous devez aussi supprimer les entrées correspondantes dans le fichier d'index. Ce programme le fait et fait aussi l'inverse lorsque des documents info sont ajoutés.

#### makeinfo

Le programme makeinfo traduit des documents sources Texinfo en d'autres formats. Les formats disponibles sont : fichiers info, texte et HTML.

#### texi2dvi

Le programme texi2dvi affiche les documents Texinfo.

#### texindex

Le programme texindex est utilisé pour trier les fichiers d'index Texinfo.

# Dépendances d'installation de Texinfo

Dernière vérification effectuée auprès de la version 4.0.

Bash: sh Binutils: ar, as, ld, ranlib Diffutils: cmp Fileutils: chmod, install, ln, ls, mkdir, mv, rm
Gcc: cc1, collect2, cpp0, gcc Grep: egrep, fgrep, grep Make: make Sed: sed
Sh−utils: basename, echo, expr, hostname, sleep Texinfo: makeinfo Textutils: cat, tr

# Textutils

## Site officiel de téléchargement

Textutils (2.1): ftp://ftp.gnu.org/gnu/textutils/

## Contenu de Textutils

Dernière vérification effectuée auprès de la version 2.0.

### Programmes

cat, cksum, comm, csplit, cut, expand, fmt, fold, head, join, md5sum, nl, od, paste, pr, ptx, sort, split, sum, tac, tail, tr, tsort, unexpand, uniq et wc

### Descriptions

#### cat

cat concatène un ou plusieurs fichiers ou entrée standard vers la sortie standard.

#### cksum

cksum affiche la somme de contrôle cyclique et le compte d'octets des fichiers spécifiés.

#### comm

comm compare deux fichiers triés ligne par ligne.

#### csplit

csplit découpe un fichier en plusieurs sections nommées xx01, xx02, ..., et affiche le compte d'octets de chaque section sur la sortie standard.

#### cut

cut sélectionne certaines colonnes d'un ou plusieurs fichiers et les affiche sur la sortie standard.

#### expand

expand convertit les tabulations d'un fichier en espaces, il affiche le résultat sur la sortie standard.

#### fmt

fmt reformate chaque paragraphe d'un ou plusieurs fichiers et affiche le résultat sur la sortie standard.

### fold

fold enregistre les lignes entrées dans des fichiers spécifiés (sur l'entrée standard par défaut), et écrit sur la sortie standard.

### head

head affiche les xx premières lignes (10 par défaut) des fichiers spécifiés sur la sortie standard.

### join

join fusionne les lignes de deux fichiers ayant un champ commun.

### md5sum

md5sum affiche ou vérifie la somme de contrôle MD5.

### nl

nl affiche les fichiers spécifiés sur la sortie standard en ajoutant le numéro de ligne.

### od

od affiche le contenu d'un fichier en octal par défaut sur la sortie standard.

### paste

paste affiche des lignes regroupant les lignes correspondantes de chaque fichier, séparées par des tabulations, sur la sortie standard.

### pr

pr met en page des fichiers pour l'impression.

### ptx

ptx produit un index permuté du contenu d'un fichier.

### sort

sort écrit la concaténation triée de fichiers sur la sortie standard.

### split

split découpe un fichier en sections de taille préfixée nommées PREFIXaa, PREFIXab, ...

### sum

sum affiche la somme de contrôle et le compte de blocs pour chaque fichier spécifié.

### tac

tac écrit les fichiers spécifiés sur la sortie standard, dans l'ordre inverse.

### tail

tail affiche les xx dernières lignes (10 par défaut) des fichiers spécifiés sur la sortie standard.

### tr

tr transpose ou élimine des caractères de l'entrée standard et affiche le résultat sur la sortie standard.

### tsort

tsort affiche une liste ordonnée cohérente à partir de fichiers partiellement ordonnés.

### unexpand

unexpand convertit les espaces en tabulations pour les fichiers spécifiés, et affiche le résultat sur la sortie standard.

### uniq

uniq efface les lignes dupliquées d'un fichier trié.

### wc

wc affiche le nombre de lignes, de mots et d'octets pour tous les fichiers spécifiés, et un total du nombre de lignes si plusieurs fichiers ont été spécifiés.

## Dépendances d'installation de Textutils

Dernière vérification effectuée auprès de la version 2.0.

Autoconf: autoconf, autoheader Automake: aclocal, automake Bash: sh Binutils: ar, as, ld, ranlib
Diffutils: cmp Fileutils: chmod, install, ls, mv, rm Gettext: msgfmt, xgettext
Gcc: cc, cc1, collect2, cpp0, gcc Glibc: getconf Grep: egrep, fgrep, grep M4: m4 Make: make
Gawk: gawk Net−tools: hostname Perl: perl Sed: sed Sh−utils: basename, echo, expr, sleep, uname
Tar: tar Texinfo: install−info, makeinfo Textutils: cat, tr

# Util–linux

## Site officiel de téléchargement

Util–Linux (2.11u): ftp://ftp.win.tue.nl/pub/linux–local/utils/util–linux/

## Contenu de Util–linux

Dernière vérification effectuée auprès de la version 2.11t.

### Programmes

agetty, arch, blockdev, cal, cfdisk, chkdupexe, col, colcrt, colrm, column, ctrlaltdel, cytune, ddate, dmesg, elvtune, fdformat, fdisk, fsck.cramfs, fsck.minix, getopt, hexdump, hwclock, ipcrm, ipcs, isosize, line, logger, look, losetup, mcookie, mkfs, mkfs.bfs, mkfs.cramfs, mkfs.minix, mkswap, more, mount, namei, parse.bash, parse.tcsh, pg, pivot_root, ramsize (lien vers rdev), raw, rdev, readprofile, rename, renice, rev, rootflags (lien vers rdev), script, setfdprm, setsid, setterm, sfdisk, swapoff (lien vers swapon), swapon, test.bash, test.tcsh, tunelp, ul, umount, vidmode (lien vers rdev), whereis et write

### Descriptions

#### agetty

agetty ouvre un port tty, affiche une invite pour un nom de login et fait appel à la commande /bin/login.

#### arch

arch affiche l'architecture de la machine.

#### blockdev

blockdev permet les appels ioctl pour les périphériques de type bloc depuis la ligne de commande.

#### cal

cal affiche un simple calendrier.

#### cfdisk

cfdisk est un gestionnaire de table des partitions de disque, basé sur la bibliothèque libncurses.

#### chkdupexe

chkdupexe cherche les exécutables dupliqués.

### col

col filtre les sauts de ligne inverses sur l'entrée standard.

### colcrt

colcrt filtre la sortie de nroff pour CRT.

### colrm

colrm enlève des colonnes d'un fichier.

### column

column met une liste sous forme de colonnes.

### ctrlaltdel

ctrlaltdel initialise l'action de la combinaison des touches CTRL+ALT+DEL (reset matériel ou logiciel).

### cytune

cytune demande et modifie les interruptions associées au périphériques Cyclades.

### ddate

ddate convertit les dates Grégoriennes en dates Discordiennes.

### dmesg

dmesg examine ou contrôle le buffer circulaire du noyau (messages de démarrage du noyau).

### elvtune

elvtune permet d'affiner les entrées/sorties des périphériques de type bloc.

### fdformat

fdformat lance le formattage bas−niveau d'une disquette.

### fdisk

fdisk est un manipulateur de tables de partition disque.

### fsck.cramfs

Aucune description encore disponible.

### fsck.minix

fsck.minix exécute une vérification logique d'une système de fichier Linux MINIX.

### getopt

getops gère les options de commande de la même façon que la fonction C getopt.

### hexdump

hexdump affiche le fichier spécifié, ou l'entrée standard, dans le format spécifié par l'utilisateur (ascii, décimal, hexadécimal, octal).

### hwclock

hwclock affiche et met à jour l'horloge matérielle (aussi appelée horloge RTC ou BIOS).

### ipcrm

ipcrm supprime la ressource spécifiée.

### ipcs

ipcs fournit des informations sur les ressources IPC.

### isosize

isosize indique la taille d'un système de fichiers iso9660

### line

line copie une ligne (jusqu'à une nouvelle ligne) de l'entrée standard et l'écrit sur la sortie standard.

### logger

logger crée une entrée dans le système de journalisation.

### look

look affiche les lignes commençant par une chaîne de caractères donnée.

### losetup

losetup active et contrôle les périphériques de type loop.

### mcookie

mcookie génère un magic−cookie pour xauth.

### mkfs

mkfs construit un système de fichiers Linux sur un périphérique, habituellement une partition d'un disque dur.

### mkfs.bfs

mkfs.bfs crée un système de fichiers SCO bfs sur un périphérique, habituellement une partition d'un disque dur.

### mkfs.cramfs

Aucune description encore disponible.

### mkfs.minix

mkfs.minix crée un système de fichiers Linux MINIX sur un périphérique, habituellement une partition d'un disque dur.

### mkswap

mkswap crée une zone de swap Linux swap sur un périphérique ou dans un fichier.

### more

more affiche du texte page par page.

### mount

mount monte le système de fichiers d'un périphérique dans un répertoire (point de montage).

### namei

namei parcourt un nom de chemin jusqu'à ce qu'un point terminal soit trouvé.

### parse.bash, parse.tcsh, test.bash, test.tcsh

Ces exemples de scripts, BASH et TCSH, utilisent le programme getopt.

### pg

Aucune description encore disponible.

### pivot_root

pivot_root déplace le système de fichiers root pour le processus courant.

### ramsize

ramsize affiche et initialise une taille de disque mémoire.

### raw

raw est utilisé pour faire correspondre un périphérique caractère Linux vers un périphérique bloc.

### rdev

rdev affiche et positionne l'image du périphérique racine, le périphérique de swap, la taille de disque mémoire, ou le mode vidéo.

### readprofile

readprofile lit les informations de fonctionnement du noyau.

### rename

rename renomme les fichiers.

### renice

renice modifie les priorités des processus en cours.

### rev

rev inverse les lignes d'un fichier.

### rootflags

rootflags affiche et met à jour les informations supplémentaires utilisées lorsque la partition racine est montée.

### script

script crée un fichier de tout ce qui est affiché dans un terminal.

### setfdprm

setfdprm initialise les paramètres des disquettes.

### setsid

setsid exécute les programmes dans une nouvelle session.

### setterm

setterm positionne les attributs d'un terminal.

### sfdisk

sfdisk est un manipulateur de tables de partition disque.

#### swapoff

swapoff désactive les périphériques et les fichiers de pagination et de swap.

#### swapon

swapon active les périphériques et les fichiers de pagination et de swap.

#### tunelp

tunelp initialise de nombreux paramètres pour le périphérique LP.

#### ul

ul lit un fichier et transforme les occurrences soulignées en séquences donnant l'aspect souligné pour le terminal utilisé.

#### umount

umount démonte un système de fichiers monté.

#### vidmode

vidmode affiche et positionne le mode vidéo.

#### whereis

whereis localise un exécutable, un fichier source ou une page de manuel de commande.

#### write

write envoie un message à un autre utilisateur.

## Dépendances d'installation de Util–linux

Dernière vérification effectuée auprès de la version 2.11n.

Bash: sh Binutils: as, ld Diffutils: cmp Fileutils: chgrp, chmod, cp, install, ln, mv, rm
Gettext: msgfmt, xgettext Gcc: cc, cc1, collect2, cpp, cpp0 Glibc: rpcgen Grep: grep Make: make
Sed: sed Sh–utils: uname, whoami textutils: cat

# Vim

## Site officiel de téléchargement

Vim (6.1): ftp://ftp.vim.org/pub/editors/vim/unix/ Correctif Vim (6.1): ftp://ftp.linuxfromscratch.org/lfs−packages/4.0/ http://ftp.linuxfromscratch.org/lfs−packages/4.0/

## Contenu de Vim

Dernière vérification effectuée auprès de la version 6.1.

### Programmes

efm_filter.pl, efm_perl.pl, ex (lien vers vim), less.sh, mve.awk, pltags.pl, ref, rview (lien vers vim), rvim (lien vers vim), shtags.pl, tcltags, vi (lien vers vim), view (lien vers vim), vim, vim132, vim2html.pl, vimdiff (lien vers vim), vimm, vimspell.sh, vimtutor and xxd

### Descriptions

#### efm_filter.pl

efm_filter.pl est un filtre qui lit à partir de l'entrée standard, copie sur la sortie standard et crée un fichier d'erreurs qui peut être lu par vim.

#### efm_perl.pl

efm_perl.pl reformatte les messages d'erreur de l'intepréteur Perl pour l'utiliser avec le mode quickfix de vim.

#### ex

ex démarre vim dans le mode Ex.

#### less.sh

less.sh est un script qui lance vim avec less.vim.

#### mve.awk

mve.awk traite les erreurs de vim.

#### pltags.pl

pltags.pl crée un fichier de tags pour du code Perl, en utilisation avec vim.

#### ref

ref vérifie l'exactitude des arguments.

### rview

rview est une version restreinte de view. Aucune commande shell ne peut être lancé et Vim ne peut être suspendu.

### rvim

rvim est une version restreinte de vim. Aucune commande shell ne peut être lancé et Vim ne peut être suspendu.

### shtags.pl

shtags.pl génère un ficheir tag pour les scripts perl.

### tcltags

tcltags génère un fichier tag pour le code TCL.

### vi

vi démarre vim en mode de compatibilité avec vi.

### view

view démarre vim en mode lecture seule.

### vim

vim démarre vim de la façon normal, par défaut.

### vim132

vim132 lance vim pour un terminal avec 132 colonnes.

### vim2html.pl

vim2html.pl convertit la documentation vim en HTML.

### vimdiff

vimdiff édite deux ou trois versions d'un fichier avec Vim et affiche les différences.

### vimm

vimm active le 'DEC locator input model' sur un terminal distant.

### vimspell.sh

vimspell.sh est un script qui vérifie un fichier et génère les actions nécessaires pour l'utilisation du surlignage de syntaxe dans vim.

#### vimtutor

vimtutor démarre le tutoriel de Vim.

#### xxd

xxd crée un dump hexadécimal ou fait l'inverse.

## Dépendances d'installation de Vim

Dernière vérification effectuée auprès de la version 6.0.

Bash: sh Binutils: as, ld, strip Diffutils: cmp, diff Fileutils: chmod, cp, ln, mkdir, mv, rm, touch
Find: find Gcc: cc1, collect2, cpp0, gcc Grep: egrep, grep Make: make Net−tools: hostname Sed: sed
Sh−utils: echo, expr, uname, whoami Textutils: cat, tr, wc

# Zlib

## Site officiel de téléchargement

Zlib (1.1.4): ftp://ftp.info−zip.org/pub/infozip/zlib/

## Contenu de Zlib

Dernière vérification effectuée auprès de la version 1.1.4.

### Bibliothèques

libz[a,so]

### Descriptions

#### libz

Il s'agit de la bibliothèque zlib, utilisée par de nombreux programmes pour ses fonctions de compression/décompression.

## Dépendances d'installation de Zlib

Aucune dépendance encore vérifiée.

# Annexe B. Ressources

# Introduction

Une liste de livres, HOWTOs et d'autres documents, qui peuvent être utile à télécharger ou à acheter, suit. Cette liste est toute petite pour commencer. Nous espérons être capable d'agrandir cette liste au fur et à mesure de la découverte de nouveaux documents ou livres utiles.

# Livres

- Linux Network Administrator's Guide publié par O'Reilly. ISBN: 1−56502−087−2
- Running Linux publié par O'Reilly. ISBN: 1−56592−151−8

*Note de l'équipe de traduction:* Ces deux livres sont disponibles en version française:

- Administration réseau sous Linux publié par O'Reilly. ISBN: 2−84177−125−3
- Le système Linux publié par O'Reilly. ISBN: 2−84177−086−9

# HOWTOs Et Guides

Tous les HOWTOs indiqués ci−dessous peuvent être téléchargés à partir du site du Linux Documentation Project à l'adresse http://www.tldp.org

- Linux Network Administrator's Guide
- From−PowerUp−To−Bash−Prompt−HOWTO

*Note de l'équipe de traduction:* Ces deux HOWTOs sont téléchargeables en version française à l'adresse suivante http://traduc.org/HOWTO/:

- Administration Réseau sous Linux
- From−PowerUp−To−Bash−Prompt−HOWTO

# Autres

- Les différentes pages man et info distribuées avec les packages.