Linux From Scratch

Table of Contents

Linux From Scratch....1
Version 3.3....1
Gerard Beekmans....1

Dédicace....2

Préface....6

Avant−propos....7

Qui voudrait lire ce livre....8

Qui ne voudrait pas lire ce livre....9

Organisation....10
Partie I − Introduction....10
Partie II − Installation du système LFS....10
Partie III − Annexes....10

I. Première Partie − Introduction....11

Chapitre 1. Introduction....12

Remerciements....13

Comment les choses se dérouleront−elles....15

Conventions utilisées dans ce livre....16

Version du livre....17

Sites miroirs....18
Miroirs HTTP....18
Amérique du Nord....18
Europe....18
Australie....18
Miroirs FTP....18
Amérique du Nord....19
Europe....19
Australie....19

Journal des modifications (Changelog)....20

Listes de diffusion et archives....34
lfs−support....34
lfs−dev....34
lfs−announce....35
lfs−security....35

Table of Contents

lfs−book....35
lfs−chat....35
alfs−discuss....35
blfs−dev....35
blfs−book....35
blfs−support....36
Archives des listes de diffusion....36
Comment envoyer un message sur une liste....36
Comment s'abonner ?....36
Comment se désabonner....36
Autres modes de listes....37
Résumé....37
En vacances....37

Serveur News....38

FAQ....39

Contacts et informations....40

Chapitre 2. Informations importantes....41

A propos de $LFS....42

Comment télécharger les logiciels....43

Comment installer les logiciels....44

Pour quelle plateforme?....46

Comment demander de l'aide....47
Information de base....47
Problèmes de configuration....47
Problèmes de compilation....47

II. Deuxième partie − Installation du système LFS....49

Chapitre 3. Packages qui doivent être téléchargés....50

Introduction....51

Packages nécessitant d'être téléchargés....52

Chapitre 4. Préparer une nouvelle partition....55

Introduction....56

Table of Contents

Créer une nouvelle partition....57

Créer un système de fichiers sur la nouvelle partition....58

Monter une nouvelle partition....59

Chapitre 5. Préparer le système LFS....60

Introduction....61

Pourquoi utiliser une édition de lien statique ?....62

Installer tous les logiciels en tant qu'utilisateur normal....63

Création des répertoires....64
Notes de conformité FHS....64

Installer Bash−2.05a....65
Installation de Bash....65
Détail de la commande....66
Contenu de bash−2.05a....66
Programmes....66
Descriptions....66
Dépendances....67

Installer Binutils−2.12....68
Installation de Binutils....68
Explication de la commande....68
Contenu de binutils−2.11.2....68
Programmes....68
Descriptions....68
Fichiers de Librairies....70
Descriptions....70
Dépendances....71

Installer Bzip2−1.0.2....72
Installation de Bzip2....72
Explication des commandes....72
Contenu de bzip2−1.0.1....72
Programmes....72
Descriptions....72
Bibliothèques....73
Dépendances....73

Installer Diffutils−2.8....74
Installation de Diffutils....74
Explication des commandes....74
Contenu de diffutils−2.7....74

Table of Contents

Programmes....74
Descriptions....74
Dépendances....75

Installer Fileutils−4.1....76
Installation de Fileutils....76
Explication de la commande....76
Contenu de fileutils−4.1....77
Programmes....77
Descriptions....77
Dépendances....79

Installer Gawk−3.1.0....80
Installation de Gawk....80
Contenu de gawk−3.1.0....80
Dépendances....80

Installer GCC−2.95.3....81
Installation de GCC....81
Explication des commandes....81
Contenu de gcc−2.95.3....82
Programmes....82
Descriptions....82
Bibliothèques....83
Dépendances....83

Installer Grep−2.5....84
Installation of Grep....84
Contenu de grep−2.4.2....84
Programmes....84
Descriptions....84
Dépendances....84

Installer Gzip−1.2.4a....86
Installation de Gzip....86
Explication des commandes....86
Contenu de gzip−1.2.4a....86
Programmes....86
Description....86
Dépendances....87

Installer Linux Kernel−2.4.18....88
Installation du noyau Linux....88
Explication des commandes....88
Pourquoi nous copions les en−têtes du noyau et pourquoi nous ne créons pas de liens....88
Contenu de kernel−2.4.17....89
Programmes....89
Descriptions....89

Table of Contents

Dépendances....89

Installer Make−3.79.1....90
Installation de Make....90
Contenu de make−3.79.1....90
Programmes....90
Descriptions....90
Dépendances....90

Installer Patch−2.5.4....91
Installation de Patch....91
Explications des commandes....91
Contenu de patch−2.5.4....91
Programmes....91
Descriptions....91
Dépendances....91

Installer Sed−3.02....92
Installation de Sed....92
Contenu de sed−3.02....92
Programmes....92
Descriptions....92
Dépendances....92

Installer Sh−utils−2.0....93
Installation de Sh−utils....93
Contenu de sh−utils−2.0....93
Programmes....93
Descriptions....93
Dépendances....96

Installer Tar−1.13....97
Installation de Tar....97
Contenu de tar−1.13....97
Programmes....97
Descriptions....97
Dépendances....97

Installer Texinfo−4.1....98
Installation de Texinfo....98
Contenu de texinfo−4.0....98
Programmes....98
Descriptions....98
Dépendances....99

Installer Textutils−2.0....100
Installation df Textutils....100
Contenu de textutils−2.0....100

Table of Contents

Programmes....100
Descriptions....100
Dépendances....102

Créer les fichiers passwd et group....103

Copie des anciennes bibliothèques NSS....104

Monter le système de fichiers $LFS/proc....105

Chapitre 6. Installation des logiciels de base du système....106

Introduction....107

A propos des symboles de débogage....108

Créer $LFS/root/.bash_profile....109

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

Changer le propriétaire de la partition LFS....111

Créer le lien symbolique /etc/mtab....112

Installer Glibc−2.2.5....113
Installation de Glibc....113
Explication des commandes....114
Contenu de glibc−2.2.5....114
Programmes....114
Descriptions....114
Bibliothèques....117
Descriptions....117
Dépendances....119

Créer les périphériques (Makedev−1.4)....120
Créer les périphériques....120
Explication des commandes....120
Contenu de MAKEDEV−1.4....120
Programmes....120
Descriptions....120
Dépendances....121

Installer Man−pages−1.48....122
Installation de Man−pages....122
Contenu de manpages−1.47....122
Fichiers support....122
Descriptions....122
Dépendances....122

Table of Contents

Installer Findutils−4.1....123
Installer Findutils....123
Note de conformité FHS....123
Explication des commandes....123
Contenu du package findutils−4.1....123
Programmes....123
Descriptions....123
Dépendances....124

Installer Gawk−3.1.0....125
Installation de Gawk....125
Contenu de gawk−3.1.0....125
Dépendances....125

Installer Ncurses−5.2....126
Installation de Ncurses....126
Explication de la commande....126
Contenu....126
Programmes....126
Descriptions....127
Bibliothèques....128
Dépendances....128

Installer Vim−6.1....129
Installation de Vim....129
Notes sur la conformité FHS....129
Explication des commandes....129
Contenu....129
Programmes....129
Descriptions....129
Dépendances....130

Installer GCC−2.95.3....131
Installation de GCC....131
Contenu de gcc−2.95.3....131
Programmes....131
Descriptions....131
Bibliothèques....132
Dépendances....133

Installation de Bison−1.34....134
Installation de Bison....134
Contenu de bison−1.31....134
Programmes....134
Descriptions....134
Dépendances....135

Table of Contents

Installer Less−374.....136
Installation de Less.....136
Contenu de less−358.....136
Programmes.....136
Description.....136
Dépendances.....136

Installer Groff−1.17.2.....137
Installation de Groff.....137
Explications des commandes.....137
Contenu de groff−1.17.2.....137
Programmes.....137
Descriptions.....137
Dépendances.....140

Installer Man−1.5j.....141
Installation de Man.....141
Contenu de man−1.5j.....141
Programmes.....141
Descriptions.....141
Dépendances.....142

Installer Perl−5.6.1.....143
Installation de Perl.....143
Contenu de perl−5.6.1.....143
Programmes.....143
Descriptions.....143
Dépendances.....145

Installer M4−1.4.....146
Installation de M4.....146
Contenu de m4−1.4.....146
Programmes.....146
Descriptions.....146
Dépendances.....146

Installer Texinfo−4.1.....147
Installation de Texinfo.....147
Explication des commandes.....147
Contenu de texinfo−4.0.....147
Programmes.....147
Descriptions.....147
Dépendances.....148

Installation de Autoconf−2.53.....149
Installation de Autoconf.....149
Contenu de autoconf−2.52.....149
Programmes.....149

Table of Contents

Descriptions....149
Dépendances....150

Installation de Automake−1.6....151
Installation de Automake....151
Contenu de automake−1.5....151
Programmes....151
Descriptions....151
Dépendances....151

Installation de Bash−2.05a....152
Installation de Bash....152
Contenu de bash−2.05a....152
Programmes....152
Descriptions....152
Dépendances....152

Installer Flex−2.5.4a....153
Installation de Flex....153
Contenu de flex−2.5.4a....153
Programmes....153
Descriptions....153
Bibliothèques....154
Descriptions....154
Dépendances....154

Installer File−3.37....155
Installation de File....155
Explications des commandes....155
Contenu de file−3.37....155
Programmes....155
Description....155
Dépendances....155

Installer Libtool−1.4.2....156
Installation de Libtool....156
Contenu de libtool−1.4.2....156
Programmes....156
Descriptions....156
Bibliothèques....156
Descriptions....156
Dépendances....156

Installation de Bin86−0.16.2....157
Installation de Bin86....157
Contenu de bin86−0.16.0....157
Programmes....157
Descriptions....157

Table of Contents

Dépendances....158

Installation de Binutils−2.12....159
Installation de Binutils....159
Explications des commandes....159
Contenu de binutils−2.11.2....159
Programmes....159
Descriptions....159
Fichiers de Librairies....161
Descriptions....161
Dépendances....161

Installation de Bzip2−1.0.2....163
Installation de Bzip2....163
Explication de la commande....163
Contenu de bzip2−1.0.1....163
Programmes....163
Descriptions....163
Bibliothèques....164
Dépendances....164

Installer Ed−0.2....165
Installation de Ed....165
Explication des commandes....165
Contenu de ed−0.2....165
Programmes....165
Description....165
Dépendances....165

Installer Gettext−0.11.1....167
Installation de Gettext....167
Contenu de gettext−0.10.40 ;....167
Programmes....167
Descriptions....167
Dépendances....168

Installer Kbd−1.06....169
Installation de Kbd....169
Explications des commandes....169
Contenu de kbd−1.06....169
Programmes....169
Descriptions....169
Dépendances....171

Installation de Diffutils−2.8....172
Installation de Diffutils....172
Contenu de diffutils−2.7....172
Programmes....172

Table of Contents

Descriptions....172
Dépendances....172

Installation de E2fsprogs−1.27....173
Installation de E2fsprogs....173
Explication des commandes....173
Contenu du package e2fsprogs−1.25....173
Programmes....173
Descriptions....173
Bibliothèques....175
Descriptions....175
Dépendances....175

Installer Fileutils−4.1....177
Installation de Fileutils....177
Contenu de fileutils−4.1....177
Programmes....177
Descriptions....177
Dépendances....179

Installer Grep−2.5....180
Installation de Grep....180
Contenu de grep−2.4.2....180
Programmes....180
Descriptions....180
Dépendances....180

Installer Gzip−1.2.4a....181
Installation de Gzip....181
Contenu de gzip−1.2.4a....181
Programmes....181
Description....181
Dépendances....182

Installer Lilo−22.2....183
Installation de Lilo....183
Contenu de lilo−22.1....183
Programmes....183
Descriptions....183
Dépendances....183

Installer Make−3.79.1....184
Installation de Make....184
Explication des commandes....184
Contenu de make−3.79.1....184
Programmes....184
Descriptions....184
Dépendances....184

Table of Contents

Installer Modutils−2.4.15.....185
Installation de Modutils.....185
Contenu de modutils−2.4.12.....185
Programmes.....185
Descriptions.....185
Dépendances.....186

Installer Netkit−base−0.17.....187
Installation de Netkit−base.....187
Contenu de netkit−base−0.17.....187
Programmes.....187
Descriptions.....187
Dépendances.....187

Installer Patch−2.5.4.....188
Installation de Patch.....188
Contenu de patch−2.5.4.....188
Programmes.....188
Descriptions.....188
Dépendances.....188

Installer Procinfo−18.....189
Installation de Procinfo.....189
Explication des Commandes.....189
Contenu de procinfo−18.....189
Programmes.....189
Descriptions.....189
Dépendances.....189

Installer Procps−2.0.7.....190
Installation de Procps.....190
Explication des Commandes.....190
Contenu de procps−2.0.7.....190
Programmes.....190
Descriptions.....190
Bibliothèques.....191
Descriptions.....191
Dépendances.....192

Installer Psmisc−20.2.....193
Installation de Psmisc.....193
Explication des Commandes.....193
Contenu de psmisc−20.2.....193
Programmes.....193
Descriptions.....193
Dépendances.....194

Table of Contents

Installer Reiserfsprogs−3.x.1b....195
Installation de Reiserfsprogs....195
Explication des Commandes....195
Contenu de reiserfsprogs−3.x.0j....195
Programmes....195
Descriptions....195
Dépendances....196

Installer Sed−3.02....197
Installation de Sed....197
Contenu de sed−3.02....197
Programmes....197
Descriptions....197
Dépendances....197

Installer Sh−utils−2.0....198
Installation de Sh−utils....198
Notes sur la conformité FHS....198
Contenu de sh−utils−2.0....198
Programmes....198
Descriptions....198
Dépendances....201

Installer Net−tools−1.60....202
Installation de Net−tools....202
Explication des commandes....202
Contenu de net−tools−1.60....202
Programmes....202
Descriptions....202
Dépendances....204

Installer Shadow−4.0.3....205
Installation de Shadow Password Suite....205
Explication des commandes....205
Contenu de shadow−20001016....205
Programmes....205
Descriptions....205
Bibliothèques....208
Descriptions....208
Dépendances....208

Installer Sysklogd−1.4.1....210
Installation de Sysklogd....210
Contenu de sysklogd−1.4.1....210
Programmes....210
Descriptions....210
Dépendances....210

Table of Contents

Installer Sysvinit−2.84 211
Installation de Sysvinit 211
Contenu de sysvinit−2.84 211
Programmes 211
Descriptions 211
Dépendances 213

Installer Tar−1.13 214
Installation de Tar 214
Contenu de tar−1.13 214
Programmes 214
Descriptions 214
Dépendances 214

Installer Textutils−2.0 215
Installation de Textutils 215
Contenu de textutils−2.0 215
Programmes 215
Descriptions 215
Dépendances 217

Installer Util−linux−2.11o 218
Notes sur la conformité FHS 218
Installation de Util−Linux 218
Explication des commandes 218
Contenu de util−linux−2.11n 218
Programmes 218
Descriptions 218
Dépendances 223

Installation de LFS−Bootscripts−1.9 224
Installation de LFS−Bootscripts 224
Contenu de LFS−bootscripts−1.9 224
Scripts 224
Descriptions 224
Dépendances 226

Désinstaller les anciennes librairies NSS 227

Configurer les logiciels essentiels 228
Configurer Vim 228
Configurer Glibc 228
Configurer le chargeur dynamique 229
Configurer Sysklogd 229
Configurer Shadow Password Suite 229
Configurer Sysvinit 230
Configuration du clavier 230
Créer les fichiers /var/run/utmp, /var/log/wtmp et /var/log/btmp 231

Table of Contents

Créer un mot de passe pour root....231

Chapitre 7. Mise en place des scripts de démarrage....232

Introduction....233

Comment fonctionne le processus de démarrage utilisant ces scripts?....234

Configuration du script setclock....236

Configuration du script loadkeys....237

Configuration du script sysklogd....238

Configurer le script localnet....239

Créer le fichier /etc/hosts....240

Configuration du script network....241
Configuration de la passerelle par défaut....241
Création des fichiers de configuration d'interfaces....241

Chapitre 8. Rendre le système LFS démarrable....242

Introduction....243

Créer le fichier /etc/fstab....244

Installer linux−2.4.18....245
Dépendances....245

Rendre son système LFS démarrable....246

Chapitre 9. La fin....247

La fin....248

Enregistrez−vous....249

Redémarrer le système....250

III. Troisième Partie − Annexes....251

Annexe A. Descriptions des packages et dépendances....252

Introduction....253

Table of Contents

Autoconf....254
Site officiel de téléchargement....254
Contenu de autoconf−2.52....254
Programmes....254
Descriptions....254
Dépendances....255

Automake....256
Site Officiel de Téléchargement....256
Contenu de automake−1.5....256
Programmes....256
Descriptions....256
Dépendances....256

Bash....257
Site Officiel de Téléchargements....257
Contenu de bash−2.05a....257
Programmes....257
Descriptions....257
Dépendances....257

Bin86....258
Site Officiel de Téléchargement....258
Contenu de bin86−0.16.0....258
Programmes....258
Descriptions....258
Dépendances....258

Binutils....259
Site Officiel de Téléchargement....259
Contenu de binutils−2.11.2....259
Programmes....259
Descriptions....259
Fichiers de Librairies....261
Descriptions....261
Dépendances....261

Bison....262
Site officiel de téléchargement....262
Contenu de bison−1.31....262
Programmes....262
Descriptions....262
Dépendances....263

Bzip2....264
Site officiel de....264
Contenu de bzip2−1.0.1....264
Programmes....264

Table of Contents

Descriptions....264
Bibliothèques....264
Dépendances....264

Diffutils....265
Site officiel de....265
Contenu de diffutils−2.7....265
Programmes....265
Descriptions....265
Dépendances....265

E2fsprogs....266
Site officiel de téléchargement....266
Contenu du package e2fsprogs−1.25....266
Programmes....266
Descriptions....266
Bibliothèques....267
Descriptions....268
Dépendances....268

Ed....269
Site officiel de téléchargement....269
Contenu de ed−0.2....269
Programmes....269
Description....269
Dépendances....269

File....270
Site officiel de téléchargement....270
Contenu de file−3.37....270
Programmes....270
Description....270
Dépendances....270

Fileutils....271
Site officiel de téléchargement....271
Contenu de fileutils−4.1....271
Programmes....271
Descriptions....271
Dépendances....273

Findutils....274
Site officiel de téléchargement....274
Contenu du package findutils−4.1....274
Programmes....274
Descriptions....274
Dépendances....275

Table of Contents

Flex....276
Site officiel de téléchargement....276
Contenu de flex−2.5.4a....276
Programmes....276
Descriptions....276
Bibliothèques....276
Descriptions....276
Dépendances....276

Gawk....277
Site officiel de téléchargement....277
Contenu de gawk−3.1.0....277
Dépendances....277

GCC....278
Site officiel de téléchargement....278
Contenu de gcc−2.95.3....278
Programmes....278
Descriptions....278
Bibliothèques....279
Dépendances....279

Gettext....280
Site officiel de téléchargement....280
Contenu de gettext−0.10.40 ;....280
Programmes....280
Descriptions....280
Dépendances....281

Glibc....282
Site officiel de téléchargement....282
Contenu de glibc−2.2.5....282
Programmes....282
Descriptions....282
Bibliothèques....284
Descriptions....284
Dépendances....286

Grep....287
Site officiel de téléchargement....287
Contenu de grep−2.4.2....287
Programmes....287
Descriptions....287
Dépendances....287

Groff....288
Site officiel de téléchargement....288
Contenu de groff−1.17.2....288

Table of Contents

Programmes....288
Descriptions....288
Dépendances....290

Gzip....291
Site officiel de téléchargement....291
Contenu de gzip−1.2.4a....291
Programmes....291
Description....291
Dépendances....292

Kbd....293
Site officiel de téléchargement....293
Contenu de kbd−1.06....293
Programmes....293
Descriptions....293
Dépendances....295

Linux kernel....296
Site officiel de téléchargement....296
Contenu de kernel−2.4.17....296
Programmes....296
Descriptions....296
Dépendances....296

Less....297
Site officiel de téléchargement....297
Contenu de less−358....297
Programmes....297
Description....297
Dépendances....297

LFS−Bootscripts....298
Site officiel de....298
Contenu de LFS−bootscripts−1.9....298
Scripts....298
Descriptions....298
Dépendances....299

Libtool....300
Site officiel de téléchargement....300
Contenu de libtool−1.4.2....300
Programmes....300
Descriptions....300
Bibliothèques....300
Descriptions....300
Dépendances....300

Table of Contents

Lilo....301
Site officiel de téléchargement....301
Contenu de lilo−22.1....301
Programmes....301
Descriptions....301
Dépendances....301

M4....302
Site officiel de téléchargement....302
Contenu de m4−1.4....302
Programmes....302
Descriptions....302
Dépendances....302

Make....303
Site officiel de téléchargement....303
Contenu de make−3.79.1....303
Programmes....303
Descriptions....303
Dépendances....303

MAKEDEV....304
Site officiel de téléchargement....304
Contenu de MAKEDEV−1.4....304
Programmes....304
Descriptions....304
Dépendances....304

Man....305
Site officiel de téléchargement....305
Contenu de man−1.5j....305
Programmes....305
Descriptions....305
Dépendances....305

Man−pages....307
Site officiel de téléchargement....307
Contenu de manpages−1.47....307
Fichiers support....307
Descriptions....307
Dépendances....307

Modutils....308
Site officiel de téléchargement....308
Contenu de modutils−2.4.12....308
Programmes....308
Descriptions....308
Dépendances....309

Table of Contents

Ncurses....310
Site officiel de téléchargement....310
Contenu....310
Programmes....310
Descriptions....310
Bibliothèques....311
Dépendances....311

Netkit−base....313
Site officiel de téléchargement....313
Contenu de netkit−base−0.17....313
Programmes....313
Descriptions....313
Dépendances....313

Net−tools....314
Site officiel de téléchargement....314
Contenu de net−tools−1.60....314
Programmes....314
Descriptions....314
Dépendances....315

Patch....316
Site officiel de téléchargement....316
Contenu de patch−2.5.4....316
Programmes....316
Descriptions....316
Dépendances....316

Perl....317
Site officiel de téléchargement....317
Contenu de perl−5.6.1....317
Programmes....317
Descriptions....317
Dépendances....319

Procinfo....320
Site officiel de téléchargement....320
Contenu de procinfo−18....320
Programmes....320
Descriptions....320
Dépendances....320

Procps....321
Site officiel de téléchargement....321
Contenu de procps−2.0.7....321
Programmes....321
Descriptions....321

Table of Contents

Bibliothèques....322
Descriptions....322
Dépendances....322

Psmisc....323
Site officiel de téléchargement....323
Contenu de psmisc−20.2....323
Programmes....323
Descriptions....323
Dépendances....323

Reiserfsprogs....324
Site officiel de téléchargement....324
Contenu de reiserfsprogs−3.x.0j....324
Programmes....324
Descriptions....324
Dépendances....324

Sed....325
Site officiel de téléchargement....325
Contenu de sed−3.02....325
Programmes....325
Descriptions....325
Dépendances....325

Shadow Password Suite....326
Site officiel de téléchargement....326
Contenu de shadow−20001016....326
Programmes....326
Descriptions....326
Bibliothèques....329
Descriptions....329
Dépendances....329

Sh−utils....330
Site officiel de téléchargement....330
Contenu de sh−utils−2.0....330
Programmes....330
Descriptions....330
Dépendances....333

Sysklogd....334
Site officiel de téléchargement....334
Contenu de sysklogd−1.4.1....334
Programmes....334
Descriptions....334
Dépendances....334

Table of Contents

Sysvinit....335
Site officiel de téléchargement....335
Contenu de sysvinit−2.84....335
Programmes....335
Descriptions....335
Dépendances....336

Tar....337
Site officiel de téléchargement....337
Contenu de tar−1.13....337
Programmes....337
Descriptions....337
Dépendances....337

Texinfo....338
Site officiel de téléchargement....338
Contenu de texinfo−4.0....338
Programmes....338
Descriptions....338
Dépendances....338

Textutils....339
Site officiel de téléchargement....339
Contenu de textutils−2.0....339
Programmes....339
Descriptions....339
Dépendances....341

Util Linux....342
Site officiel de téléchargement....342
Contenu de util−linux−2.11n....342
Programmes....342
Descriptions....342
Dépendances....347

Vim....348
Site officiel de téléchargement....348
Contenu....348
Programmes....348
Descriptions....348
Dépendances....349

Annexe B. Ressources....350

Introduction....351

Table of Contents

Livres.....352

HOWTOs Et Guides.....353

Autres.....354

Linux From Scratch

Version 3.3

Gerard Beekmans

Copyright © 1999−2002 par Gerard Beekmans

Ce livre décrit le processus de création d'un système Linux depuis rien à partir d'une distribution Linux déjà installée, 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*
  - *Comment télécharger les logiciels*
  - *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 une nouvelle partition*
- *5. Préparer le système LFS*
  - *Introduction*
  - *Pourquoi utiliser une édition de lien statique ?*
  - *Installer tous les logiciels en tant qu'utilisateur normal*
  - *Création des répertoires*
  - *Installer Bash−2.05a*
  - *Installer Binutils−2.12*
  - *Installer Bzip2−1.0.2*
  - *Installer Diffutils−2.8*
  - *Installer Fileutils−4.1*

Installer Gawk−3.1.0
Installer GCC−2.95.3
Installer Grep−2.5
Installer Gzip−1.2.4a
Installer Linux Kernel−2.4.18
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−4.1
Installer Textutils−2.0
Créer les fichiers passwd et group
Copie des anciennes bibliothèques NSS
Monter le système de fichiers $LFS/proc
6. Installation des logiciels de base du système
Introduction
A propos des symboles de débogage
Créer $LFS/root/.bash_profile
Entrée dans l'environnement "chrooté"
Changer le propriétaire de la partition LFS
Créer le lien symbolique /etc/mtab
Installer Glibc−2.2.5
Créer les périphériques (Makedev−1.4)
Installer Man−pages−1.48
Installer Findutils−4.1
Installer Gawk−3.1.0
Installer Ncurses−5.2
Installer Vim−6.1
Installer GCC−2.95.3
Installation de Bison−1.34
Installer Less−374
Installer Groff−1.17.2
Installer Man−1.5j
Installer Perl−5.6.1
Installer M4−1.4
Installer Texinfo−4.1
Installation de Autoconf−2.53
Installation de Automake−1.6
Installation de Bash−2.05a
Installer Flex−2.5.4a
Installer File−3.37
Installer Libtool−1.4.2
Installation de Bin86−0.16.2
Installation de Binutils−2.12
Installation de Bzip2−1.0.2
Installer Ed−0.2
Installer Gettext−0.11.1
Installer Kbd−1.06
Installation de Diffutils−2.8
Installation de E2fsprogs−1.27

Installer Fileutils−4.1
Installer Grep−2.5
Installer Gzip−1.2.4a
Installer Lilo−22.2
Installer Make−3.79.1
Installer Modutils−2.4.15
Installer Netkit−base−0.17
Installer Patch−2.5.4
Installer Procinfo−18
Installer Procps−2.0.7
Installer Psmisc−20.2
Installer Reiserfsprogs−3.x.1b
Installer Sed−3.02
Installer Sh−utils−2.0
Installer Net−tools−1.60
Installer Shadow−4.0.3
Installer Sysklogd−1.4.1
Installer Sysvinit−2.84
Installer Tar−1.13
Installer Textutils−2.0
Installer Util−linux−2.11o
Installation de LFS−Bootscripts−1.9
Désinstaller les anciennes librairies NSS
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
Configuration du script loadkeys
Configuration du script sysklogd
Configurer le script localnet
Créer le fichier /etc/hosts
Configuration du script network
8. Rendre le système LFS démarrable
Introduction
Créer le fichier /etc/fstab
Installer linux−2.4.18
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
Linux kernel
Less
LFS−Bootscripts
Libtool
Lilo
M4
Make
MAKEDEV
Man
Man−pages
Modutils
Ncurses
Netkit−base
Net−tools
Patch
Perl
Procinfo
Procps
Psmisc
Reiserfsprogs
Sed
Shadow Password Suite
Sh−utils
Sysklogd
Sysvinit
Tar
Texinfo
Textutils
Util Linux
Vim

B. Ressources

Introduction
Livres
HOWTOs Et Guides
Autres

Préface

Avant–propos

Ayant utilisé certaines distributions de Linux, je n'ai jamais été satisfait par aucune d'entre elles. Je n'aimais pas la façon dont les scripts de boot étaient arrangés, je n'aimais pas la manière dont certains programmes étaient configurés par défaut, et d'autres choses de ce genre. J'en suis arrivé à la conclusion que si je voulais être entièrement satisfait d'un système linux, il me fallait créer mon propre système depuis le début, idéalement en n'utilisant que le code source. Sans utiliser de package pré–compilé. Sans l'aide d'aucun CD–ROM ni d'aucune disquette d'amorce qui installeraient quelques utilitaires de base. J'utiliserais mon système Linux actuel et m'en servirais pour construire le mien.

Ceci, au premier abord, peut sembler très difficile et parfois impossible. Après avoir réglé les problèmes de dépendance, problèmes de compilation, etc..., un système Linux personnalisé a été créé avec succès. J'ai nommé ce système le système LFS, c'est à dire "Linux From Scratch" (Linux Par Le Début).

J'espère que vous prendrez plaisir à travailler sur 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 afin 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 question censée à laquelle j'espère pouvoir vous répondre.

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

Un autre avantage de LFS est que vous pouvez créer un système Linux compact. Quand vous installez une distribution courante, vous finirez 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 d'avoir un système LFS installé pour moins de 100 Mo. Celà vous semble−t−il 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 supermarché ou au restaurant fast−food, vous le mangeriez sans précisément savoir ce que vous mangez, alors que LFS vous donne tous les ingrédients pour faire un hamburger vous−même. 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 cet 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 souhaite : 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 d'installer la plomberie, le système électrique, la cuisine, la baignoire, le papier−peint, etc.

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 patchs 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

Les personnes ne voulant pas construire un système linux en entier depuis le début ne voudront probablement pas lire ce livre. Si cependant vous voulez en apprendre plus au sujet de ce qui se produit dans les coulisses, en particulier sur ce qui se passe entre le moment où vous allumez un ordinateur et le moment où la ligne de commande apparaît, vous devriez lire le << From−PowerUp−To−Bash−Prompt−HOWTO>> . Ce HOWTO décrit l'assemblage d'un système de base, d'une manière similaire à celle utilisée dans ce livre−ci, mais il se concentre plus sur l'installation d'un système de démarrage que sur celle d'un système complet.

Afin de décider de lire ce livre ou le From−PowerUp−To−Bash−Prompt−HOWTO, posez−vous cette question : "Mon objectif principal est−il d'avoir un système Linux opérationnel que je vais construire moi−même, et ainsi apprendre le rôle de chacun de ses composants, ou alors est−il juste d'apprendre ?". Si vous voulez construire et apprendre, lisez ce livre. Si vous voulez simplement apprendre les bases, alors le From−PowerUp−To−Bash−Prompt−HOWTO est probablement un meilleur choix.

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

Organisation

Ce livre est divisé en plusieurs parties. Bien que toutes les annexes soient regroupées dans la deuxième partie (ce qui alourdit quelque peu le livre), nous pensons qu'il est plus facile de le lire ainsi. Cela vous évite d'avoir à cliquer sur une annexe puis de retourner d'où vous veniez dans la deuxième partie, ce qui est une vrai corvée, particulièrement si vous lisez la version TXT de ce livre.

Partie I – Introduction

La première partie donne les informations générales à propos de ce livre (les versions, où se le procurer, le journal des modifications, les listes de diffusions, et comment nous contacter). Elle explique également certains aspects importants qu'il faut lire avant de commencer à créer son système LFS.

Partie II – Installation du système LFS

La seconde partie va vous guider à travers l'installation du système LFS, qui deviendra la base du reste du système. 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 voudrions remercier les personnes et les organisations suivantes pour leurs contributions au 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.
- Hagen Herrschaft <hrx@hrxnet.de> pour la maintenance du miroir de.linuxfromscratch.org.
- 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 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 l'installation 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. Nul besoin de disquette de démarrage. Nous utiliserons un système Linux existant comme base de travail (puisque nous aurons besoin d'un compilateur, d'un éditeur de liens, d'un éditeur de texte et d'autres outils).

Après avoir téléchargé les packages nécessaires à la constitution d'un système LFS, vous devrez créer une nouvelle partition native Linux, sur laquelle sera installé le système LFS.

L'étape suivante, abordée dans le Chapitre 5, est l'installation d'un nombre de packages liés statiquement et installés sur la partition LFS. Ces packages forment une suite de développement de base utilisée pour installer le système réel et sont aussi nécessaires pour résoudre des dépendances circulaires. Les exemples de dépendances circulaires sont: vous avez besoin d'un compilateur pour installer un compilateur, vous avez besoin d'un shell pour installer un shell. Et ainsi de suite.

Le Chapitre 6 permet l'installation du système de base en question. Nous utiliserons le programme chroot pour commencer un nouveau shell, dont le répertoire root sera situé dans la partition LFS. En un sens, cela équivaut à redémarrer et avoir le noyau monté sur la partition LFS de la même manière qu'une partition root. Pourquoi utiliser chroot plutôt que redémarrer? Parce qu'ainsi, vous conservez la possibilité d'utiliser votre système hôte. Tandis que le logiciel est en cours d'installation, vous pouvez passer sur une autre CV (Console Virtuelle) ou bureau X et continuer d'utiliser votre ordinateur.

Lorsque le logiciel sera entièrement installé, on passera à la configuration des scripts de démarrage dans le Chapitre 7. Le chapitre 8 configurera le chargeur Linux et dans le chapitre 9, vous trouverez des 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 mais principalement pour attirer l'attention sur des points importants ou pour donner des exemples sur 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 de tout ce qui peut être tapé dans les 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 3.3 datée du 7 avril 2002. Si cette version a plus d'un mois, une version plus récente est probablement disponible en téléchargement. Allez voir l'un de nos sites miroirs ci–dessous pour des versions remises à jour.

Sites miroirs

Ci−dessous se trouve une liste de sites miroirs HTTP et FTP en date du 7 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 [100 Mbit] − http://www.linuxfromscratch.org/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
- Freising, Allemagne [4 Mbit] − http://www.de.linuxfromscratch.org/lfs/intro.shtml
- Université Technique de Vienne, Autriche [64 Mbit] − http://www.at.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
- Teeside, Royaume−Uni [256 Kbit] − http://www.linuxfromscratch.co.uk/lfs/intro.shtml
- Amsterdam, Hollande [100 Mbit] − http://www.nl.linuxfromscratch.org/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] [100 Mbit] – ftp://ftp.linuxfromscratch.org
- Fremont, Californie, USA [HTTP] [100 Mbit] – http://ftp.linuxfromscratch.org

Europe

- 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
- 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 de Twente, Hollande [HTTP] [100 Mbit] – http://ftp.nl.linuxfromscratch.org/linux/lfs
- Université de Twente, Hollande [FTP] [100 Mbit] – ftp:/ftp.nl.linuxfromscratch.org/pub/linux/lfs
- Freising, Allemagne [FTP] [4 Mbit] – ftp://ftp.de.[..]/mirrors/ftp.linuxfromscratch.org

Australie

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

Journal des modifications (Changelog)

3.3 – 7 avril 2002

- Mise à jour vers:
  - autoconf–2.53
  - automake–1.6
  - bin86–0.16.2
  - binutils–2.12
  - bison–1.34
  - bzip2–1.0.2
  - diffutils–2.8
  - e2fsprogs–1.27
  - gawk–3.1.0
  - gettext–0.11.1
  - grep–2.5
  - less–374
  - lfs–bootscripts–1.9
  - lilo–22.2
  -

linux−2.4.18

- man−pages−1.48
- modutils−2.4.15
- reiserfsprogs−3.x.1b
- shadow−4.0.3
- texinfo−4.1
- util−linux−2.11o
- vim−6.1

- 7 avril 2002 [gerard]: Ajout d'un nouveau site miroir localisé à Freising, Allemagne.
- 5 avril 2002 [gerard]: Chapitre 07 − Loadkeys: Ajout de de cette page expliquant que vous pouvez supprimer le lien symbolique du script loadkeys `/etc/rc.d/rcsysinit.d` si vous avez compilé le plan de codage directement dans le noyau.
- 5 avril 2002 [gerard]: Chapitre 06 − Configuration du clavier: Explique que vous pouvez aussi compiler le plan de codage directement dans le noyau, ce qui apporte quelques avantages.
- 5 avril 2002 [gerard]: Mise à jour vers lfs−bootscripts−1.9
- 5 avril 2002 [gerard]: Chapitre 05+06 − GCC: Ajout des commandes pour supprimer les répertoires `/usr/*-gnu`.
- 4 avril 2002 [gerard]: Chapitre 05 − Diffutils: Ajout de −−disable−nls
- 3 avril 2002 [gerard]: Annexe A − Gettext: Ajout des descriptions manquantes de certains programmes.
-

3 avril 2002 [gerard]: Chapitre 05 – Monter le système de fichiers $LFS/proc: Ajout de la commande **`chown root.root $LFS/proc`**. Cette commande récursive de chown dans le chapitre 6 ne touche pas à proc, donc celle–ci continuera à appartenir à l'utilisateur *lfs*. Ce n'est pas très important, juste une manière propre de faire les choses.

- 3 avril 2002 [gerard]: Chapitre 06 – Groff: Ajout de quelques liens symboliques utilisés par des programmes comme **`xman`**.

- 3 avril 2002 [gerard]: Chapitre 04 – Montage d'une nouvelle partition: Ajout de quelques notes sur la façon de gérer de multiples partitions ($LFS, $LFS/usr et ainsi de suite).

- 3 avril 2002 [gerard]: Chapitre 06 – E2fsprogs: Ajout de la commande **`install–info`** pour finir l'installation des pages info.

- 3 avril 2002 [gerard]: Chapitre 06 – Bzip2: Inversion des commandes **`make`** et **`make –f Makefile–libbz2_so`**. Ceci est nécessaire pour que tous les fichiers objets soient compilés avec l'option PIC (Position Independant Code).

- 3 avril 2002 [gerard]: Chapitre 05 – Linux: Les instructions d'installation ont été raccourcies en supprimant les étapes **`make config`** et **`make dep`**.

- 1er avril 2002 [gerard]: Ce n'est pas une blague: Chapitres 5+6 – Gawk: Ajout d'un avertissement pour ne jamais lancer **`make uninstall`** sur ce package. Cela serait pratiquement identique à un **`rm –rf /usr/bin/*`** parce que nous modifions la définition du répertoire `libexec` par `/usr/bin`

- 29 mars 2002 [markh]: Chapitres 05 et 06 – Mise à jour vers diffutils–2.8, modutils–2.4.15 et vim–6.1. Suppression du paramètre PR_PROGRAM pour diffutils car /usr/bin/pr est maintenant détecté par le script configure. Suppression de sed pour corriger le problème de coloration syntaxique du shell dans vim car cela est corrigé avec la nouvelle version.

- 26 mars 2002 [markh]: Chapitre 02 – Comment demander de l'aide: Ajout d'une référence au document d'ESR sur la façon pour demander de l'aide.

- 25 mars 2002 [markh]: Binutils – Ajout d'une description de la bibliothèque libopcodes.

- 21 mars 2002 [gerard]: Chapitre 06 – Bzip2: Avant de déplacer les fichiers `/usr/bin/bzless` et `/usr/bin/bzmore` dans le répertoire `/bin`, nous devons d'abord supprimer les fichiers `/bin/bzless` et `/bin/bzmore`. Sur certains systèmes, écraser les fichiers existants ne fonctionne pas à cause des liens symboliques utilisés.

-

21 mars 2002 [gerard]: Annexe A – Sysklogd: Mise à jour du site de téléchargement vers http://www.infodrom.org/projects/sysklogd/

- 20 mars 2002 [gerard]: Chapitre 06 – Configuration du chargeur dynamique: Suppression des répertoires `/lib` et `/usr/lib` du fichier `ld.so.conf`. Ils n'étaient pas nécessaires.

- 16 mars 2002 [gerard]: Chapitre 06 + Annexe A: Suppression des dépendances pour chroot. Ce n'est pas un package, donc cela n'a pas sa place ici.

- 16 mars 2002 [gerard]: Chapitres 05+06 – Gawk: Ajout des commandes pour modifier avec sed le fichier `awklib/Makefile.in` en changeant les définitions de *datadir* et *libexecdir*.

- 5 mars 2002 [gerard]: Chapitre 01 – Listes de diffusions: Ajout de la description de lfs–chat.

- 15 mars 2002 [gerard]: Chapitre 06–Shadow: Déplacement de `libmisc.*a` vers `/usr/lib`.

- 14 mars 2002 [gerard]: Mise à jour vers bison–1.34, gettext–0.11.1, grep–2.5, lfs–bootscripts–1.8, shadow–4.0.3

- 11 mars 2002 [gerard]: Mise à jour vers binutils–2.12

- 11 mars 2002 [gerard]: Chapitre 07 – Setclock: Le texte indiquait le fait que l'on pouvait passer la configuration, ce qui n'est pas vrai si le script entier n'est pas effacé. Donc le texte a été un peu remanié pour les laisser créer le fichier quelle que soit la façon dont l'horloge matérielle est configurée.

- 11 mars 2002 [gerard]: Chapitre 07 – Loadkeys: Suppression du besoin de configuration du fichier `/etc/sysconfig/keyboard`. Le patch kbd le rend obsolète (loadkeys –d est maintenant utilisé).

- 11 mars 2002 [gerard]: Chapitre 05 – Gawk: Ajout de la correction de bug avec –Dre_max_failures=re_max_failures2 pour les systèmes glibc–2.1.x.

- 11 mars 2002 [gerard]: Chapitre 06 – Bzip2: Avant l'installation, suppression de `/usr/bin/bz*`. L'installation de bzip2 ne gère pas proprement les fichiers déjà existants lors de la configuration des liens, d'où la suppression des fichiers.

- 10 mars 2002 [gerard]: Chapitre 06 – Configuration de keyboard: Ajout d'une section pour configurer le fichier de correspondance clavier en créant le lien symbolique du fichier `/usr/share/kbd/keymaps/defkeymap.map.gz`.

9 mars 2002 [gerard]: Chapitre 08 – Rendre le système LFS démarrable: Ajout d'une commande **cp** qui récupère toutes les images noyau à partir du fichier `/etc/lilo.conf` et les copie dans `$LFS/boot`.

- 9 mars 2002 [gerard]: Chapitre 06 – Man: Déplacement de `man.conf` du répertoire `/usr/share/misc` vers `/etc`.

- 9 mars 2002 [gerard]: Chapitre 07: Ajout d'une page sur le script sysklogd et explication sur l'inclusion de l'option *–m 0* pour **syslogd**.

- 8 mars 2002 [gerard]: Suppression du package Mawk et remplacement avec Gawk. Ceci a été fait car mawk n'est plus développé alors que gawk l'est. Mawk a des bugs qui sont corrrigés dans Gawk.

- 8 mars 2002 [gerard]: Mise à jour vers: autoconf–2.53, automake–1.6, bin86–0.16.2, bison–1.33, bzip2–1.0.2, e2fsprogs–1.27, gawk–3.1.0, gettext–0.11, less–374, lilo–22.2, linux–2.4.18, man–pages–1.48, modutils–2.4.14, reiserfsprogs–3.x.1b, shadow–4.0.2, texinfo–4.1, util–linux–2.11o

3.2 – 7 mars 2002

- Mise à jour:

  - lfs–bootscripts–1.6

- 1er mars 2002 [gerard]: Chapitre 05 – Création des répertoires: Suppression des répertoires `/usr/var` et `/usr/local/var`. Ils ne sont pas recommandés par le *FHS*.

- 27 février 2002 [gerard]: Chapitre 06 – Make: Ajout des commandes pour supprimer le bit setgid kmem de `/usr/bin/make`. Ceci n'est pas nécessaire sur les systèmes Linux pour s'occuper de la charge système et il cause des problèmes supplémentaires résolus en supprimant ce bit.

- 26 février 2002 [gerard]: Mise à jour vers lfs–bootscripts–1.6

- 17 février 2002 [gerard]: Chaptire 05 – Sh–utils: Ajout de nouveau de la commande qui déplace $LFS/usr/bin/chroot vers $LFS/usr/bin.

- 17 février 2002 [gerard]: Mise à jour des dépendances pour tous les paquetages.

-

15 février 2002 [gerard]: Chapitre 01: Ajout dans la liste d'un nouveau miroir, situé en Hollande (www.nl et ftp.nl).

- 11 février 2002 [markh]: Chapitre 05: Sh−utils: Suppression des && supplémentaires en fin des instructions d'installation.

- 10 février 2002 [gerard]: Chapter 05 − Sh−utils: Suppression de *su* de la commande *mv* comme il n'est pas installé dans le chapitre 5.

3.2−RC1 − 10 février 2002

- Mise à jour vers:
  - bison−1.31
  - file−3.37
  - glibc−2.2.5
  - kbd−1.06−2.patch
  - lfs−bootscripts−1.5
  - linux−2.4.17
  - man−pages−1.47
  - psmisc−20.2
  - sysvinit−2.84
  - util−linux−2.11n

- 10 février 2002 [gerard]: Chapitre 6: Ajout d'une commande sed pour modifier le chemin /usr/bin/gzip enregistré en dur dans gzexe en /bin/gzip.

-

10 février 2002 [gerard]: Chapitres 5+6: Déplacement des programmes additionnels, utilisé par les bootscripts, vers le répertoire ($LFS)/bin. Aucun programme utilisé par bootscripts (à l'exception des démons eux−mêmes) ne doit être dans le répertoire /usr au cas où /usr ne soit disponible que bien après le processus de démarrage (par exemple lorsqu'il s'agit d'un partage NFS).

- 6 février 2002 [markh]: Annexe A − Toutes les descriptions sont maintenant synchronisées et mises à jour.

- 2 février 2002 [gerard]: Chapitre 6 − Changer le propriétaire de la partition LFS: Ajout de << cd />> de façon à ce que le premier caractère slash puisse être supprimé de tous les répertoires dans les commandes chown. C'est plus plaisant de les taper ainsi.

- 2 février 2002 [gerard]: Mise à jour vers lfs−bootscripts−1.5

- 2 février 2002 [gerard]: Chapitre 6 − Gzip: Suppression du lien symbolique compress. Gzip peut décompresser les fichier .Z mais ne peut pas compresser dans ce format.

- 1er février 2002 [gerard]: Mise à jour vers lfs−bootscripts−1.3

- 1er février 2002 [gerard]: Chapitre 6 − Glibc: Au lieu de modifier avec sed le fichier `config.make`, on crée le fichier `glibc-build/configparms` contenant << cross−compiling = no>> .

- 30 janvier 2002 [marcheerdink]: Chapitre 5: Modification des commandes pour copier les fichiers d'entête pour supporter les versions de cp plus anciennes que la 4.1.

- 30 janvier 2002 [markh]: Chapitres 5+6: Ajout de CPPFLAGS="$CPPFLAGS −D_GNU_SOURCE" à la commande configure pour patch. Ceci permet la compilation sur les plateformes PPC et m68k et ne gêne pas sur x86.

- 30 janvier 2002 [gerard]: Chapitre 5 − Monter le système de fichiers $LFS/proc: ré écriture du texte (il laisse entendre que vous pouvez seulement monter le système de fichiers proc plus d'une fois, ce qui n'est plus vrai de nos jours).

- 30 janvier 2002 [markh]: Chapitre 5: Amélioration de l'explication sur make mrproper.

- 30 janvier 2002 [marcheerdink]: Chapitres 5+6: Suppression de l'option −−libexecdir des options de configure pour fileutils.

-

30 janvier 2002 [marcheerdink]: Chapitre 6: Ajout d'un lien symbolique de vipw vers vigr après l'installation de shadow.

- 30 janvier 2002 [markh]: Chapitres 5+6: Modification des instructions d'installation de binutils et e2fsprogs pour utiliser des répertoires séparés, de la même manière que gcc et glibc.

- 30 janvier 2002 [gerard]: Chapitre 6 – Bootscripts: Ajout d'un chown root.root après le cp.

- 30 janvier 2002 [gerard]: Annexe A – Texinfo: Le programme info fonctionne sur le répertoire /usr/share/info et non /usr/doc/info.

- 30 janvier 2002 [gerard]: Chapitre 6 – Procps: Correction d'une erreur de frappe dans le chemin du répertoire app–defaults (il s'agit de /usr/X11R6/lib/X11/app–defaults au lieu de /usr/X11R6/lib/app–defaults)

- 30 janvier 2002 [gerard]: Chapitre 6 – Configuration des logiciels: simplification des commandes pour créer les fichiers utmp, btmp, lastlog et wtmp.

- 30 janvier 2002 [gerard]: Chapitre 1: Déplacement de Remerciements pour qu'il soit affiché sur la première page du chapitre 1.

- 30 janvier 2002 [gerard]: Chapitre 1: Création d'une page séparée listant les miroirs HTTP et FTP.

- 30 janvier 2002 [gerard]: Chapitre 4 – Créer une partition: La taille de la partition suggérée a été agrandie (passée de 750Mo à 1Go).

- 29 janvier 2002 [gerard]: Chapitre 6 – Shadow: Combinaison des commandes << mv libshadow.a /usr/lib>> et << mv libshadow.la /usr/lib>> pour obtenir la commande << mv libshadow.*a /usr/lib>> .

- 26 janvier 2002 [gerard]: Mise à jour vers lfs–bootscripts–1.2

- 26 janvier 2002 [marcheerdink]: Chapitre 6: Suppression de l'option datadir du configure pour bison, car les versions récentes de bison utilisent par défaut le bon répertoire.

- 23 janvier 2002 [markh]: Chapitre 6: Ajout de la section Création du lien /etc/mtab.

-

23 janvier 2002 [gerard]: Suppression de la commande file −C lors de l'installation de file. Ce package lance cette commande à la fin de l'installation donc nous n'avons plus besoin de le faire.

- 23 janvier 2002 [marcheerdink]: Chapitres 4+5+6: L'environnement statique est maintenant construit en tant qu'utilisateur non privilégié, enlevant le risque d'écrire sur des fichiers de la distribution hôte.
- 22 janvier 2002 [markh]: Suppression des instructions d'installation des pages de manuel de linuxthreads, car elles ne fonctionnent pas (elles ont besoin de perl, ce dont nous ne disposons pas à ce moment).
- 21 janvier 2002 [markh]: Mise à jour vers glibc−2.2.5. De plus, résolution de l'installation de glibc de manière à ce que les pages de manuel de linuxthreads soient installées.
- 21 janvier 2002 [markh]: Mise à jour vers bison−1.31, file−3.37, kernel−2.4.17, psmisc−20.2 et sysvinit−2.84.
- 21 janvier 2002 [markh]: Mise à jour vers util−linux−2.11n et suppression de ADD_RAW=yes car cette option n'est plus nécessaire.
- 21 janvier 2002 [markh]: Mise à jour vers man−pages−1.47 et suppression du patch pour les pages de manuel.
- 15 janvier 2002 [gerard]: Annexe A: Ajout des scripts de démarrage (dépendances, site de téléchargement, descriptions).
- 15 janvier 2002 [gerard]: Chapitre 6: Ajout de l'installation des scripts de démarrage.
- 15 janvier 2002 [gerard]: Chapitre 7: Suppression de la plupart des scripts, en laissant simplement ceux pour lesquels nous configurons des fichiers dans /etc/sysconfig.
- 15 janvier 2002 [gerard]: Chapitre 6 − Configuring Sysvinit: Modification du contenu de inittab pour correspondre aux nouveaux scripts de démarrage.
- 15 janvier 2002 [marcheerdink]: Chapitre 6 − File: Modification des instructions d'installation car la commande sed n'est plus nécessaire.
- 14 janvier 2002 [marcheerdink]: Modification de l'installation des entêtes du noyau dans le chapitre 5 pour que cela soit un peu plus portable.
-

6 janvier 2002 [gerard]: Reformattage des listes de dépendances.

- 1er janvier 2002 [gerard]: Bonne année LFS!

- 1er janvier 2002 [markh]: Premier changelog de la nouvelle année: Mise à jour de la notice de copyright pour couvrir 2002 ;–) OK – Je suis triste...

- 16 décembre 2001 [gerard]: Chapitre 6 – Ed: Modification du texte expliquant que ed est optionnel pour éviter toute confusion.

- 16 décembre 2001 [gerard]: Chapitre 6 – Texinfo: Modification de l'explication sur TEXMF pour éviter toute confusion.

- 15 décembre 2001 [gerard]: Chapitre 4: Remplacement de la référence à l'astuce "One partition hint" par la référence à l'astuce lfs_next_to_existing_systems.txt.

- 15 décembre 2001 [markh]: Fin de la jointure de l'annexe. Toutes les anciennes annexes A, B et D sont maintenant une seule (grande) annexe A.

- 14 décembre 2001 [markh]: Jointure des annexes A et B.

- 13 décembre 2001 [markh]: Annexe B: Modification du tag dbhtml de manière à ce que la page flex soit maintenant créée en tant que flex.html au lieu de flex.

- 13 décembre 2001 [markh]: Annexe D: Modification des références à metalab.unc.edu et ftp.ibiblio.org vers l'URL correcte ibiblio.org.

- 12 décembre 2001 [marcheerdink]: Chapitre 6: Déplacement du patch de kbd vers les instructions d'installation par défaut; mise à jour vers kbd–1.06–2.patch pour corriger l'installation de certaines programmes; ajout des descriptions de ces programmes; suppression du message pour loadkeys –d, maintenant que loadkeys –d est corrigé.

- 11 décembre 2001 [markh]: Chapitre 6: Ajout de l'explication "pourquoi faire cd $LFS avant chroot".

- 10 décembre 2001 [markh]: Chapitre 6: Ajout du patch pour kbd suite au comportement de loadkeys –d (patch de Matthias Benkmann; posté originellement sur la liste lfs–dev).

-

10 décembre 2001 [markh]: Chapitre 6: Re−création des liens symboliques dans les instructions pour bash, fileutils et gcc pour que le chapitre 6 soit indépendant au niveau des instructions par rapport au chapitre 5.

- 10 décembre 2001 [marcheerdink]: Chapitres 5+6: Nettoyage des instructions sed pour utiliser le fichier de sauvegarde qui nous avons créé précédemment au lieu d'écrire dans un fichier intermédiaire 'tmp~'.

- 10 décembre 2001 [marcheerdink]: Chapitres 5+6: la commande 'make' pour l'installation de diffutils a changé en 'make PR_PROGRAM=/usr/bin/pr.' Ce bug a été rapporté par Greg Schafer.

- 7 décembre 2001 [gerard]: Chapitre 6: Modification de la commande configure de *./Configure −Dprefix=/usr* en *./configure.gnu −−prefix=/usr*. Ceci est plus consistant avec les instructions d'installation pour les autres paquetages, et le résultat est identique à celui obtenue avec l'ancienne version.

- 3 décembre 2001 [markh]: Chapitre 2: Ajout de la section "Quelle plateforme?".

3.1 − 3 décembre 2001

- Ajouté:
  - reiserfsprogs−3.x.0j
- Mise à jour:
  - MAKEDEV−1.4
  - bash−2.05a
  - e2fsprogs−1.25
  - gettext−0.10.40
  - libtool−1.4.2
  - lilo−22.1
  -

linux−2.4.16

- man−1.5j
- man−pages−1.43
- modutils−2.4.12
- sysvinit−2.83
- util−linux−2.11m
- vim−6.0

- 30 novembre 2001 [markh]: Chapitre 6: Mise à jour vers man−1.5j. Suppression de sed. Nous utilisions une ancienne version alors que la nouvelle détecte parfaitement awk.
- 30 novembre 2001 [markh]: Chapitre 5: Ajout de l'explication d'une librairie statique qui avait été envoyée au départ sur lfs−apps (quand elle existait encore) par Plasmatic.
- 26 novembre 2001 [markh]: Chapitres 5+6: Mise à jour vers kernel−2.4.16 et modutils−2.4.12.
- 26 november 2001 [markh]: Chapitre 6: Ajout de notes sur la compatibilité FHS à l'installation du package findutils.
- 19 novembre 2001 [markh]: Chapitres 5+6: Mise à jour vers bash−2.05a, lilo−22.1, MAKEDEV−1.4, man−pages−1.43 et util−linux−2.11m.
- 5 novembre 2001 [markh]: Chapitre 6: Création d'un nouveau script lex au lieu d'un lien sur flex et ceci suite au commentaire sur la liste lfs−dev (exactement comme nous avons fait pour bison et yacc).
- 7 ctobre 2001 [markh]: général: Fait de l'ordre au niveau XML. Cela ne doit pas affecter le contenu ou la présentation du livre. Si cela n'est pas le cas, quelque chose s'est mal passée!
- 7 octobre 2001 [markh]: Chapitre 6: Ajout de reiserfsprogs−3.x.0j et mise à jour vers lilo−22.0.2.
-

24 octobre 2001 [markh]: général: Correction d'un bon nombre de fautes d'orthographe qui avaient été signalées.

- 12 octobre 2001 [markh]: Chapitre 5 – Noyau: Ajout d'explications sur les raisons pour lesquelles nous copions les entêtes du noyau plutôt que de faire un lien symbolique sur eux.

- 12 octobre 2001 [markh]: Annexe A – Gzip: Ajout de la descritption manquante de gunzip.

- 12 octobre 2001 [markh]: Chapitre 6 – Util−linux: Suppression de USRGAMES_DIR=/usr/bin qui est devenu inutile avec util−linux−2.11l.

- 9 octobre 2001 [gerard]: Chapitre 6 – Kbd: Suppression de l'option −−datadir, par défaut kbd est déjà initialisé correctement.

- 7 octobre 2001 [gerard]: Chapitre 6 – Shadow: Ajout du lien http://hints.linuxfromscratch.org/hints/shadowpasswd_plus.txt vers lfs−hint

- 7 octobre 2001 [gerard]: Chapitre 6 – Vim: Modification des instructions d'installation pour corriger un bug dans le fichier vim−6.0 `syntax/sh.vim`, et ajout de la variable CPPFLAGS pour spécifier le fichier vimrc global en tant que `/etc/vimrc`

- 7 octobre 2001 [gerard]: Chapitre 6: Mise à jour vers libtool−1.4.2, lilo−22.0, man−pages−1.40, modutils−2.4.10, sysvinit−2.83, util−linux−2.11l et vim−6.0

- 2 octobre 2001 [gerard]: Chapitre 9 – La fin: Ajout d'un lien sur le compteur LFS sur http://linuxfromscratch.org/cgi−bin/lfscounter.cgi

- 26 septembre 2001 [gerard]: Chapitre 1 – Serveur de news: Ajout d'un lien sur le serveur de news

- 26 septembre 2001 [markh]: Chapitre 6 – E2fsprogs: Changement de −−with−root−prefix=/ vers −−with−root−prefix="" dans les instructions d'installation de e2fsprogs. La raison de ce changement est que si nous mettons / cela donnera un truc du style //lib au lieu de /lib lors de la mise en place de liens symboliques ou de chemins. Cela n'est pas mauvais mais ce n'est pas très propre.

- 26 septembre 2001 [markh]: Chapitres 5+6: Mise à jour vers e2fsprogs−1.25, gettext−0.10.40, linux−2.4.10, modutils−2.4.9 et util−linux−2.11i.

- 22 septembre 2001 [markh]: Annexe A: Réorganise les descriptions dans l'ordre alphabétique.

3.0 – 21 septembre 2001

- mise à jour vers:
  - e2fsprogs−1.24
- 21 septembre 2001 [markh]: Chapitres 1+7: Changement de l'information sur la liste de diffusion pour qu'elle reflète la nouvelle organisation de la ml. Les scripts rc et rcS dans le chapitre 7 demandent maintenant aux gens de remonter les problèmes sur la liste lfs−dev plutôt que sur lfs−discuss.
- 18 septembre 2001 [gerard]: Chapitres 5+6 – GCC: Ajout de −−enable−threads=posix au chapitre 5, et changement de −−enable−threads vers −−enable−threads=posix au chapitre 6 bien que par défaut les threads posix soient utilisés quand rien n'est spécifié. De cette façon, il est plus clair de voir les choses qui sont validées.
- 17 septembre 2001 [gerard]: Chapitre 6 – Psmisc: Ajout de quelques notes sur comment faire avec les liens symboliques de psmisc (au cas où sysvinit ne soit pas installé) et ajout des pages de manuel. De plus, ajout de −−exec−prefix=/ au script de configuration de psmisc de façon à ce que les programmes soit installés dans /bin plutôt que /usr/bin (le script de démarrage peut les utiliser, c'est pourquoi ils doivent être dans /bin).
- 16 Septembre 2001 [markh]: Chapitre 6 – Util−linux: Ajout de USRGAMES_DIR=/usr/bin à la routine d'installation du make. Comme cela /usr/games n'est pas créé en tant que bannière et il est installé dans /usr/bin.
- 14 septembre 2001 [markh]: Chapitre 6 – E2fsprogs: Mise à jour vers la version 1.24.
- 11 septembre 2001 [gerard]: Chapitre 6 – Man: Ajout de && manquant à 'done' et modification des droits du script de configuration à l'aide de la commande chmod pour passer du mode 755 au lieu de 700 (de façon à ce que les gens ne soit pas obligés d'être le propriétaire de ce fichier pour l'exécuter).

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" offre une assistance aux utilisateurs construisant un système LFS, mais seulement pour le matériel contenu dans le 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 quant à 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 du 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 questionnements 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 l'éditeur du livre pour coordonner sa mise à jour ainsi que les publications XML et les préférences. Les discussions du moment à propos des choses 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 inclus) 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 signifie 'Automated Linux From Scratch'. Le but de ce projet est de développer un outil d'installation qui permettrait d'installer de façon automatique un système LFS. Le principal objectif est d'accélérer la compilation en se dégageant de l'obligation d'entrer manuellement les commandes de configuration, de compilation et d'installation des packages.

blfs−dev

La liste de diffusion "blfs−dev" s'occupe des problèmes liés au livre BLFS (Beyond LFS, c'est−à−dire après LFS). Si des problèmes surviennent avec ce livre, que ce soient des bugs ou des suggestions pour améliorer ce livre, cette liste de diffusion est là pour ça.

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 compris dans LFS. Cette liste ne se limite pas à fournir de l'aide pour les seuls logiciels renseignés dans BLFS, mais peut être utilisée pour tout logiciel.

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 *listname@linuxfromscratch.org*, où *listname* 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 a été envoyé, le programme Listar répondra avec un email de demande de confirmation d'abonnement. Après que cette confirmation a 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 a été envoyé, le programme Listar répondra avec un email de demande de confirmation de désabonnement. Après que cette confirmation a été renvoyée, Listar retournera un autre email avec un message indiquant le désabonnement de l'utilisateur de(s) liste(s).

Autres modes de listes

Les modes pouvant être activés par l'utilisateur requiert l'envoi d'un email à listar@linuxfromscratch.org. Les modes en eux−même sont activés en tapant les commandes appropriées dans le champ Objet 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 email, ceci peut être réalisé en laissant le champ Objet vide et en tapant les commandes dans le corps de l'email.

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 auxquelles 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 recevoir un unique email 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 ce mail quotidien ainsi que chacun des emails 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 temps et qu'il ne souhaite plus recevoir d'email 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 un problème lors de la construction de votre système LFS, vous devez visiter ce lien http://www.linuxfromscratch.org/faq/ pour voir si votre problème n'a pas déjà une réponse dans la FAQ.

Contacts et informations

Envoyez de préférence tous vos emails vers l'une des listes de diffusion. Voir Chapitre 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 plait, 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 qui contient votre système LFS. La méthode pour créer cette partition et le choix de l'endroit où la monter seront expliqués en détails au chapitre 4. A titre d'exemple, supposons que la partition LFS soit montée dans le répertoire /mnt/lfs.

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

Il est très important que cela soit fait quel que soit l'endroit où vous le lisez, que ce soit pour une commande à saisir dans 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 accompli 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`** 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 re–créer le fichier /etc/passwd de votre système courant. 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 connections (ou sessions par 'su') vers le compte root, la variable $LFS soit définie.

Comment télécharger les logiciels

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

Vous pourriez par exemple choisir comme convention d'avoir un répertoire $LFS/usr/src/sources, dans lequel vous pourriez créer des répertoires 0−9 et a à z. Un package tel que sysvinit−2.84.tar.bz2 serait alors placé dans $LFS/usr/src/sources/s/. Le package bash−2.05a.tar.bz2 serait quant à lui dans $LFS/usr/src/sources/b/, et ainsi de suite.

Le chapitre suivant contient la liste de tous les packages que vous devez télécharger, mais la partition devant héberger votre système LFS n'a pas encore été créée. Les fichiers seront donc stockés temporairement ailleurs, libre à vous de décider où, et ce ne sera qu'à la fin du chapitre décrivant la préparation de cette partition que vous les déplacerez dans le répertoire $LFS/usr/src/.

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
```

Quelques versions de tar (la plupart de nos jours, mais pas encore toutes) ont été modifiées pour utiliser directement bzip2 grâce à l'option I, y ou j, qui fonctionne de manière identique à l'option z pour la gestion des archives compressées avec gzip. La commande ci−dessus fonctionne quel que soit la manière dont votre système a été modifié pour gérer le bzip2 dans la commande tar.

Si une archive est simplement en 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. A 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 simples fichiers tels que des fichiers de patch. Ces fichiers sont généralement 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 de ré−utiliser 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 utilisés au chapitre 5 auront des paramètres de configuration propres à la distribution hôte que vous avez utilisé, 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 garantie 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.

Il y a une exception à cette règle : les sources du noyau. Gardez les dans un coin, car vous en aurez besoin plus tard pour compiler un noyau. Rien n'utilisera l'arborescence des sources du noyau ; ainsi l'arboresence des sources ne vous gênera pas. Si, toutefois, vous manquez de place disque, vous pouvez supprimer les sources du noyau et les re−décompresser plus tard, lorsque cela sera nécessaire.

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, vous vous rendrez compte que les personnes sur IRC et sur les listes de discussion seront prêtes à vous aider. Vous trouverez la liste des listes de discussion concernant LFS dans Chapitre 1 – Listes de diffusion et archives. Pour nous permettre de vous aider, vous devrez vous assurer que vous aurez relevé autant d'informations pertinentes que vous le pouvez. Cela nous aidera à diagnostiquer et résoudre votre problème. Cette partie du livre va vous indiquer le type d'informations utiles.

Information de base

Tout d'abord, nous aurons besoin d'une explication rapide du problème. Les choses essentielles à inclure sont :

- la version du livre que vous utilisez, c'est–à–dire 3.3
- avec quel package ou section vous avez un problème
- quel est le message d'erreur exact, ou quel symptôme, que vous recevez
- 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, car après tout la 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 de tous les packages à télécharger pour construire le système de base. Les numéros de version affichés correspondent à des versions de logiciel qui fonctionnent, et qui sont la base du livre. Si vous rencontrez des problèmes que vous ne pouvez résoudre vous−même, veuillez télécharger dans ce cas la version indiquée par le livre (au cas où vous auriez téléchargé une version plus récente).

Toutes les URLs ci−dessous appartiennent au serveur ftp.linuxfromscratch.org. Nous avons quelques miroirs FTP disponibles, à partir desquels vous pouvez aussi télécharger les fichiers. Les adresses des sites miroirs peuvent être trouvées sur Chapitre 1 − Version du livre.

Nous avons fourni une liste des sites officiels de téléchargement des packages dans Annexe A. L'archive LFS FTP ne contient que les versions des packages recommandées par ce livre. Vous avez toujours la possibilité de consulter les sites officiels dans l'Annexe A afin de vérifier s'il existe une nouvelle version. Pour ceux qui téléchargeront une version plus récente, nous apprécierions énormément de savoir si vous êtes parvenus ou non à installer cette version, grâce aux instructions de ce livre.

Prenez note que tous les fichiers téléchargés de l'archive LFS FTP sont compressés avec bzip2 au lieu de gzip. Si vous ne savez pas comment utiliser les fichiers bz2, veuillez vous reporter à Chapitre 2 − Comment installer les logiciels.

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 – 87,260 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/3.3/lfs–packages–3.3.tar
http://ftp.linuxfromscratch.org/lfs–packages/3.3/lfs–packages–3.3.tar
Soit télécharger individuellement les packages suivants: Bash (2.05a) – 1,400 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/3.3/bash–2.05a.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/3.3/bash–2.05a.tar.bz2 Binutils (2.12) – 9,312 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/3.3/binutils–2.12.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/3.3/binutils–2.12.tar.bz2 Bzip2 (1.0.2) – 610 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/3.3/bzip2–1.0.2.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/3.3/bzip2–1.0.2.tar.bz2 Diff Utils (2.8) – 640 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/3.3/diffutils–2.8.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/3.3/diffutils–2.8.tar.bz2 File Utils (4.1) – 1217 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/3.3/fileutils–4.1.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/3.3/fileutils–4.1.tar.bz2 GCC (2.95.3) – 9,618 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/3.3/gcc–2.95.3.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/3.3/gcc–2.95.3.tar.bz2 Patch de GCC (2.95.3–2) – 8 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/3.3/gcc–2.95.3–2.patch.bz2
http://ftp.linuxfromscratch.org/lfs–packages/3.3/gcc–2.95.3–2.patch.bz2
Noyau Linux (2.4.18) – 23,595 Ko: ftp://ftp.linuxfromscratch.org/lfs–packages/3.3/linux–2.4.18.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/3.3/linux–2.4.18.tar.bz2 Grep (2.5) – 545 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/3.3/grep–2.5.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/3.3/grep–2.5.tar.bz2 Gzip (1.2.4a) – 178 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/3.3/gzip–1.2.4a.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/3.3/gzip–1.2.4a.tar.bz2 Patch de Gzip (1.2.4a) – 1 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/3.3/gzip–1.2.4a.patch.bz2
http://ftp.linuxfromscratch.org/lfs–packages/3.3/gzip–1.2.4a.patch.bz2 Make (3.79.1) – 794 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/3.3/make–3.79.1.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/3.3/make–3.79.1.tar.bz2 Sed (3.02) – 221 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/3.3/sed–3.02.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/3.3/sed–3.02.tar.bz2 Sh–utils (2.0) – 824 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/3.3/sh–utils–2.0.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/3.3/sh–utils–2.0.tar.bz2 Patch de Sh–utils (2.0) – 1 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/3.3/sh–utils–2.0.patch.bz2
http://ftp.linuxfromscratch.org/lfs–packages/3.3/sh–utils–2.0.patch.bz2 Tar (1.13) – 730 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/3.3/tar–1.13.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/3.3/tar–1.13.tar.bz2 Patch de Tar (1.13) – 1 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/3.3/tar–1.13.patch.bz2
http://ftp.linuxfromscratch.org/lfs–packages/3.3/tar–1.13.patch.bz2 Text Utils (2.0) – 1,040 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/3.3/textutils–2.0.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/3.3/textutils–2.0.tar.bz2 Gawk (3.1.0) – 1,286 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/3.3/gawk–3.1.0.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/3.3/gawk–3.1.0.tar.bz2 Texinfo (4.1) – 1,161 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/3.3/texinfo–4.1.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/3.3/texinfo–4.1.tar.bz2 Patch (2.5.4) – 149 Ko:
ftp://ftp.linuxfromscratch.org/lfs–packages/3.3/patch–2.5.4.tar.bz2
http://ftp.linuxfromscratch.org/lfs–packages/3.3/patch–2.5.4.tar.bz2 MAKEDEV (1.4) – 7 Ko:

ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/MAKEDEV−1.4.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/MAKEDEV−1.4.bz2 Glibc (2.2.5) − 12,114 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/glibc−2.2.5.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/glibc−2.2.5.tar.bz2 Glibc−linuxthreads (2.2.5) − 164 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/glibc−linuxthreads−2.2.5.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/glibc−linuxthreads−2.2.5.tar.bz2
Man−pages (1.48) − 537 Ko: ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/man−pages−1.48.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/man−pages−1.48.tar.bz2 Ed (0.2) − 158 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/ed−0.2.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/ed−0.2.tar.bz2 Find Utils (4.1) − 226 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/findutils−4.1.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/findutils−4.1.tar.bz2 Patch de Find Utils (4.1) − 1 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/findutils−4.1.patch.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/findutils−4.1.patch.bz2 Ncurses (5.2) − 1,308 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/ncurses−5.2.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/ncurses−5.2.tar.bz2 Vim (6.1) − 2,890 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/vim−6.1.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/vim−6.1.tar.bz2 Bison (1.34) − 585 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/bison−1.34.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/bison−1.34.tar.bz2 Less (374) − 189 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/less−374.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/less−374.tar.bz2 Groff (1.17.2) − 1,214 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/groff−1.17.2.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/groff−1.17.2.tar.bz2 Man (1.5j) − 167 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/man−1.5j.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/man−1.5j.tar.bz2 Perl (5.6.1) − 4,750 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/perl−5.6.1.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/perl−5.6.1.tar.bz2 M4 (1.4) − 249 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/m4−1.4.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/m4−1.4.tar.bz2 Autoconf (2.53) − 739 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/autoconf−2.53.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/autoconf−2.53.tar.bz2 Automake (1.6) − 451 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/automake−1.6.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/automake−1.6.tar.bz2 Flex (2.5.4a) − 278 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/flex−2.5.4a.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/flex−2.5.4a.tar.bz2 File (3.37) − 140 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/file−3.37.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/file−3.37.tar.bz2 Libtool (1.4.2) − 653 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/libtool−1.4.2.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/libtool−1.4.2.tar.bz2 Bin86 (0.16.2) − 112 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/bin86−0.16.2.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/bin86−0.16.2.tar.bz2 Gettext (0.11.1) − 2,039 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/gettext−0.11.1.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/gettext−0.11.1.tar.bz2 Kbd (1.06) − 559 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/kbd−1.06.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/kbd−1.06.tar.bz2 Patch Kbd (1.06−2) − 3 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/kbd−1.06−2.patch.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/kbd−1.06−2.patch.bz2 E2fsprogs (1.27) − 1,176 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/e2fsprogs−1.27.tar.bz2
http://ftp.linuxfromscratch.org/lfs−packages/3.3/e2fsprogs−1.27.tar.bz2 Lilo (22.2) − 292 Ko:
ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/lilo−22.2.tar.bz2

http://ftp.linuxfromscratch.org/lfs-packages/3.3/lilo-22.2.tar.bz2 Modutils (2.4.15) - 211 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/modutils-2.4.15.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/modutils-2.4.15.tar.bz2 Procinfo (18) - 22 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/procinfo-18.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/procinfo-18.tar.bz2 Procps (2.0.7) - 153 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/procps-2.0.7.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/procps-2.0.7.tar.bz2 Psmisc (20.2) - 123 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/psmisc-20.2.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/psmisc-20.2.tar.bz2 Reiserfsprogs (3.x.1b) - 243 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/reiserfsprogs-3.x.1b.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/reiserfsprogs-3.x.1b.tar.bz2
Kit Shadow Password (4.0.3) - 760 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/shadow-4.0.3.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/shadow-4.0.3.tar.bz2 Sysklogd (1.4.1) - 67 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/sysklogd-1.4.1.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/sysklogd-1.4.1.tar.bz2 Sysvinit (2.84) - 76 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/sysvinit-2.84.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/sysvinit-2.84.tar.bz2 Util Linux (2.11o) - 1,020 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/util-linux-2.11o.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/util-linux-2.11o.tar.bz2 Netkit-base (0.17) - 49 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/netkit-base-0.17.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/netkit-base-0.17.tar.bz2 Net-tools (1.60) - 194 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/net-tools-1.60.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/net-tools-1.60.tar.bz2
Scripts de démarrage LFS (1.9) - 26 Ko:
ftp://ftp.linuxfromscratch.org/lfs-packages/3.3/lfs-bootscripts-1.9.tar.bz2
http://ftp.linuxfromscratch.org/lfs-packages/3.3/lfs-bootscripts-1.9.tar.bz2
Taille totale de tous les packages: 87,260 Ko (85.21 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, puis un système de fichiers et enfin la structure des répertoires. Après cela, nous pourrons passer au chapitre suivant et entamer réellement le processus de construction.

Créer une nouvelle partition

Tout d'abord, sachez qu'il est possible de construire un système LFS sur une seule partition, où 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. Au cas où cela vous tente, jetez 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. Gardez à l'esprit que ceci est un vrai hint et que par conséquent, il ne s'agit pas encore d'un document fini.

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'environ 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 envie de plus d'espace que cela, si vous décidez d'utiliser le système LFS comme système principal. Si c'est le cas, vous aurez besoin de plus d'espace pour installer d'autres logiciels. Si vous disposez déjà d'une partition native Linux, vous pouvez passer cette sous−section.

Le programme cfdisk (ou tout autre programme de style fdisk de votre choix) est 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). Cela permet de créer une partition native Linux, écrire la table de partition et sortir du programme cfdisk. Veuillez vous reporter à la documentation concernant le programme de style 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. Ce peut être quelque chose comme hda11. Cette partition nouvellement créée sera appelée la partition LFS dans ce livre.

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. C'est, bien sûr, à vous de décider quel système de fichiers vous voulez créer, mais comme nous devons travailler avec un d'entre eux, nous considèrons que vous avez choisi le système de fichier ext2.

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 une 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 au répertoire /mnt/lfs. Nous considérerons dans ce livre que vous l'avez monté 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 l'identificateur de votre partition (tel que hda11).

Ce répertoire (/mnt/lfs) est la variable $LFS dont nous avons déjà parlé au 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 apropriées des partitions.

Chapitre 5. Préparer le système LFS

Introduction

Dans les chapitres suivants, nous allons installer tous les logiciels qui font partie d'un système Linux de base. Après en avoir terminé avec ce chapitre et le suivant, vous aurez un système Linux pleinement fonctionnel. Les chapitres restants traitent de la création des scripts de démarrage, de la manière de rendre le système LFS amorçable, et de la mise en place du réseau de base.

Les logiciels dans ce chapitre seront liés statiquement et seront réinstallés au chapitre suivant avec une édition de liens dynamiques. Nous commencons par une version statique parce qu'il est possible que notre système Linux normal et notre système LFS n'utilisent pas la même version de bibliothèque C. Si les programmes de la première partie étaient liés à une version ancienne de la bibliothèque C, ces programmes pourraient ne pas bien fonctionner dans le système LFS. Une autre raison est de résoudre les dépendances circulaires. Un exemple d'une telle dépendance est que vous avez besoin d'un compilateur pour installer un compilateur, et que vous allez avoir besoin d'un shell pour installer un shell et le compilateur.

La clef pour apprendre ce qui fait fonctionner Linux est de savoir exactement quelle est l'utilité de chaque package et en quoi un utilisateur ou le système en a besoin. Les descriptions du contenu des packages sont fournies après la sous−section Installation de chaque package, et aussi dans l'annexe A.

Pendant l'installation de certains packages, vous verrez très probablement des messages d'alerte du compilateur défilant 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++. C'est simplement que, souvent, les standards du langage C ont changé, et les packages utilisent encore le vieux standard, ce qui n'est pas un problème).

Avant de commencer, assurez−vous que vous avez positionné correctement la variable LFS, 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).

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.

Lorsque l'on crée (compile) un programme, plutôt que d'avoir à réécrire l'ensemble des fonctions nécessaires à l'accès au noyau, au matériel, aux fichiers, etc. on récupère toutes ces fonctions de base 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'accès aux fichiers, l'affichage d'informations à l'écran, et les comptes–rendus aux 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 régulièrement 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.

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'à ce que l'on utilise la bibliothèque). L'avantage REEL de l'édition de liens dynamiques est qu'il nous suffit d'une seule copie de cette bibliothèque. Si `ls` et `rm` utilisent tous deux la même bibliothèque, ils n'ont pas besoin de deux copies de cette bibliothèque, alors qu'ils peuvent tous les deux accéder au code d'un seul et 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 l'édition de liens dynamiques a tous ces avantages, pourquoi utiliser alors exclusivement l'édition de liens statiques ? Et bien, c'est parce que lorsque vous exécuterez chroot dans votre flambant neuf (mais plutôt incomplet) environnement LFS, ces bibliothèques dynamiques ne seront pas disponibles car elles se situeront dans 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 être sûr 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 la création de la principale bibliothèque du système, glibc. Cela fait, nous commencerons à recréer 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 l'édition de liens dynamiques, consultez un ouvrage ou un site web sur la programmation, plus spécialement consacré à Linux.

Installer tous les logiciels en tant qu'utilisateur normal

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

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

Maintenant il est temps de changer les permissions de votre partition LFS de manière à ce que votre utilisateur "lfs" ait les droits en écriture dessus. Lancez la commande suivante en tant que root pour changer le propriétaire de la partition LFS:

```
chown -R lfs $LFS
```

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
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 et $LC_ALL sont positionnées. $LFS a déjà été expliqué dans les chapitres précédents. $LC_ALL est une variable utilisée pour l'internationalisation.

Quand votre distribution de base utilise glibc en version antérieure à la 2.2.4, en ayant LC_ALL valant autre chose que "C" ou "POSIX", lors du travail du chapitre 5, peut poser des problèmes entre le moment où vous sortez de l'environnement chroot du chapitre 6 et que vous essayez d'y retourner. En indiquant "POSIX" ("C" est un alias pour "POSIX"), vous vous assurez que tout fonctionnera comme prévu dans l'environnement chroot.

Création des répertoires

Créons maintenant la hiérarchie de répertoires sur la partition de LFS, basé sur le standard FHS qui peut être trouvé sur http://www.pathname.com/fhs/. Lancer les commandes suivantes pour créer la hiérarchie par défaut des répertoires:

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

Normalement, les répertoires sont créés avec le mode de permission 755, ce qui n'est pas souhaité pour tous les répertoires. La première modification concerne le répertoire $LFS/root avec une permission 750. Ainsi, personne ne pourra entrer dans le répertoire /root (de façon identique à un utilisateur pour son répertoire /home/nomutilisateur). Le deuxième changement concerne les répertoires tmp avec un mode 1777. De cette façon, tout utilisateur peut écrire des données dans /tmp et /var/tmp mais ne peut pas supprimer des fichiers d'autres utilisateurs (ceci grâce au "sticky bit" – bit 1 du masque 1777).

```
cd $LFS &&
chmod 0750 root &&
chmod 1777 tmp var/tmp
```

Maintenant que les répertoires sont créés, copiez les fichiers sources qui ont été téléchargés dans le chapitre 3 sur un sous–répertoire de $LFS/usr/src (vous devrez créer ce répertoire vous–même).

Notes de conformité FHS

Le FHS stipule que le répertoire /usr/local doit contenir les sous–répertoires bin, games, include, lib, man, sbin et share. Vous pouvez modifier votre répertoire /usr/local vous–même si vous souhaitez que votre système soit en conformité avec FHS.

De la même manière, le standard indique qu'il doit exister un répertoire /usr/share/games, ce dont nous ne voulons pas pour un système de base. Mais n'hésitez pas à rendre votre système conforme à FHS si vous le souhaitez. Le FHS n'est pas très précis sur la structure des sous–répertoires de /usr/local/share, donc nous avons pris la liberté de créer les répertoires qui nous semblaient nécessaires.

Installer Bash–2.05a

```
Estimation du temps de construction:           3 minutes
Estimation de l'espace disque requis:  20 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 suivez les instructions du livre que vous êtes en train de lire précisement.

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, alors créez `libcurses.a` comme un lien symbolique en exécutant les commandes suivantes en tant qu'utilisateur root:

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

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

```
./configure --enable-static-link --prefix=$LFS/usr \
   --bindir=$LFS/bin --with-curses &&
make &&
make install &&
cd $LFS/bin &&
ln -sf bash sh
```

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)
```

cela veut dire probablement que vous utilisez une distribution Debian et que vous avez une ancienne version du package texinfo. Cette erreur n'est pas gênante outre mesure : les pages d'info seront installées quand vous recompilerez bash dynamiquement au chapitre 6, vous pouvez donc l'ignorer.

Lorsque nous l'avons testé avec la dernière distribution de Debian, les deux dernières commandes ont été exécutées car le processus d'installation n'a pas retourné de valeurs supérieures à 0. Mais vous feriez bien de vérifier si vous avez le lien symbolique `$LFS/bin/sh` dans votre partition LFS. Si vous ne l'avez pas, lancez les deux dernières commandes manuellement.

Détail de la commande

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

--prefix=$LFS/usr: Cette option installe tous les fichiers Bash dans le répertoire $LFS/usr, qui devient alors le répertoire /usr lors d'un chroot ou d'un reboot dans LFS.

--bindir=$LFS/bin: Ceci installe les fichiers exécutables dans $LFS/bin. Nous faisons ceci pour que bash soit dans /bin, et non pas dans /usr/bin. Une des raisons d'être: /usr devrait être dans une partition différente et, avant d'être montée vous aurez besoin de bash (il serait difficile d'exécuter les scripts au boot sans un shell de disponible).

--with-curses: Ceci permet à Bash d'être lié à la bibliothèque curses plutôt qu'à la bibliothèque termcap par défaut, laquelle devient 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 là 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 requiert ncurses et termcap ne peut plus alors être utilisé.

ln -sf bash sh: Cette commande crée le lien symbolique sh qui pointe sur bash. La plupart des scripts se lance eux-même via 'sh' (par #!/bin/sh en première ligne du script) qui invoque un mode spécial de bash. Bash se comportera alors (aussi fidèlement que possible) comme le Bourne shell original.

Les caractères **&&** à la fin de chaque ligne permettent de n'exécuter la prochaine commande que si la commande précédente existe et retourne 0 indiquant qu'elle s'est exécutée sans erreur. Dans ce cas, toutes les commandes sont copiées et collées dans le shell, il est important de s'assurer que si ./configure échoue, make ne sera pas exécuté et, de ce fait, si make échoue, alors make install ne doit pas être exécuté, et ainsi de suite.

Contenu de bash-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

Bash−2.05a nécessite que les programmes suivants soient installés:

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–2.12

```
Estimation du temps de construction:           6 minutes
Estimation de l'espace disque requis:  96 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 –march et –mcpu). Binutils fonctionne bien mieux tel quel, nous vous recommandons donc de ne pas modifier CFLAGS, CXXFLAGS et autres variables ou options qui pourraient modifier l'optimisation par défaut.

Installer Binutils en exécutant les commandes suivantes :

```
mkdir ../binutils-build &&
cd ../binutils-build &&
../binutils-2.12/configure --prefix=$LFS/usr --disable-nls &&
make LDFLAGS=-all-static tooldir=$LFS/usr &&
make tooldir=$LFS/usr install
```

Explication de la commande

`mkdir ../binutils-build`: Les instructions d'installation pour Binutils recommande 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 dans 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.

`tooldir=$LFS/usr`: Normalement, tooldir (le répertoire où les exécutables de binutils vont se situer) est $(exec_prefix)/$(target_alias) ce qui donne, par exemple, /usr/i686–pc–linux–gnu. Depuis que nous avons construit notre propre système, nous n'avons pas eu besoin de positionner cette option au répertoire $LFS/usr. Cette option ne devrait être utilisée que pour une plateforme de compilation croisée (par exemple compiler un package sur une machine Intel qui génère un code exécutable sur des PowerPC).

Contenu de binutils–2.11.2

Programmes

addr2line, ar, as, c++filt, 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.

c++filt

Le langage C++ fournit un mécanisme de surcharge, ce qui signifie qu'il est possible d'écrire plusieurs fonctions ayant le même nom (pourvu que chacune demande des paramètres de types différents). Tous les noms de fonction C++ sont encodés dans un nom d'assemblage de bas niveau (ce processus est connu sous le nom de broyage). Le programme c++filt fait l'association inverse : il décode (dé−broie) les noms de bas niveau en noms de niveau utilisateur, pour que l'éditeur de liens puisse prévenir un conflit entre ces fonctions surchargées.

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 objfile 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.

Fichiers de Librairies

libbfd.a, libiberty.a et libopcodes.a

Descriptions

libbfd

libbfd est une librairie de description des fichiers binaires.

libiberty

libiberty est un ensemble de sous−routines utilisé dans de divers programmes GNU dont getopt, obstack, strerror, strtol et strtoul.

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

Binutils−2.11.2 nécessite que les programmes suivants soient installés:

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:           1 minute
Estimation de l'espace disque requis:  3 Mo
```

Installation de Bzip2

Installez Bzip2 en lançant les commandes suivantes:

```
make CC="gcc -static" &&
make PREFIX=$LFS/usr install &&
cd $LFS/usr/bin &&
mv bzcat bunzip2 bzip2 bzip2recover bzless bzmore $LFS/bin
```

Bien qu'il ne fasse pas partie d'un système de base LFS, il est utile de mentionner qu'un patch de compression/décompression utilisant les programmes bzip2/bunzip2 est disponible au téléchargement. Sans ajout de ce patch, vous devrez utiliser des commandes comme bzcat file.tar.bz|tar xv ou tar −−use−compress−prog=bunzip2 −xvf file.tar.bz2 pour utiliser bzip2/bunzip2 avec tar. Ce patch apporte l'option –j pour décompresser une archive bzip2 avec la commande tar xvfj file.tar.bz2. Appliquer ce patch sera décrit plus tard quand le package tar sera installé.

Explication des commandes

`make CC="gcc -static"`: Cette méthode nous permet de dire à gcc que nous voulons lier bzip2 statiquement.

Contenu de bzip2–1.0.1

Programmes

bunzip2 (lien vers bzip2), bzcat (lien vers bzip2), bzip2 et bzip2recover

Descriptions

bunzip2

Bunzip2 décompresse les fichiers compressés avec bzip2.

bzcat

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

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.

Bibliothèques

libbz2.[a,so]

libbz2

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

Dépendances

Bzip2−1.0.1 nécessite que les programmes suivants soient installés:

bash: sh binutils: ar, as, ld, ranlib fileutils: cp, ln, rm gcc: cc1, collect2, cpp0, gcc make: make

Installer Diffutils–2.8

```
Estimation du temps de construction:           1 minute
Estimation de l'espace disque requis:  4 Mo
```

Installation de Diffutils

Si vous installez Diffutils en utilisant glibc–2.1.x à partir de votre système de base, il pourra être nécessaire d'utiliser un patch pour empêcher un conflit de nom de variable. Les commandes suivantes peuvent être utilisées dans ce cas. Notez que ces commandes peuvent aussi être utilisées pour d'autres versions de glibc, donc en cas de doute, utilisez la première version.

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

Si vous utilisez une version récente de glibc (2.2.x), vous pouvez utiliser les commandes suivantes pour installer Diffutils:

```
./configure --prefix=$LFS/usr --disable-nls &&
make LDFLAGS=-static &&
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 certaines plate–formes (cela dépend de la version de Glibc utilisée par le système) et cette construction résout ce problème.

Contenu de diffutils–2.7

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

Diffutils−2.7 nécessite que les programmes suivants soient installés:

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:           3 minutes
Estimation de l'espace disque requis:  25 Mo
```

Installation de Fileutils

A partir d'un fileutils lié statiquement, les programmes peuvent causer des erreurs de segmentation sur certains systèmes, si votre distribution possède Glibc–2.2.3 ou supérieur installée. Cela semble apparaître souvent sur des machines disposant d'un processeur AMD; cependant quelques systèmes à base de processeurs Intel présentent les mêmes disfonctionnements. Si votre système appartient à cette catégorie, essayez la correction suivante.

Noter que, dans certains cas, utiliser les commandes sed aboutira à des problèmes de compilation du package, même si votre système dispose d'un processeur AMD et de Glibc–2.2.3 (ou supérieur) installée. Si c'est le cas, vous aurez besoin de supprimer le répertoire fileutils–4.1 et de le déballer de nouveau de 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, exécuter les commandes suivantes. *N'essayez pas* cette correction si vous n'avez pas Glibc–2.2.3 installé. Cela créerait toutes sortes de problèmes de compilation.

```
cp lib/Makefile.in lib/Makefile.in.backup &&
sed -e 's/\(.*\)\(fopen-safer\.c \)\\/\1\2atexit.c \\/' \
   -e 's/\(.*\)\(idcache\$U\.\$.*\)\\/\1\2atexit$U.$(OBJEXT) \\/' \
   lib/Makefile.in.backup > lib/Makefile.in
```

Installez Fileutils en lançant les commandes suivantes:

```
./configure --disable-nls \
   --prefix=$LFS/usr --bindir=$LFS/bin &&
make LDFLAGS=-static &&
make install &&
cd $LFS/usr/bin &&
ln -sf ../../bin/install
```

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

Explication de la commande

`cp lib/Makefile.in lib/Makefile.in.backup :` Nous exécutons cette commande pour garder une copie du fichier que nous allons modifier.

```
cp lib/Makefile.in lib/Makefile.in.backup &&
sed -e 's/\(.*\)\(fopen-safer\.c \)\\/\1\2atexit.c \\/' \
   -e 's/\(.*\)\(idcache\$U\.\$.*\)\\/\1\2atexit$U.$(OBJEXT) \\/' \
   lib/Makefile.in.backup > lib/Makefile.in:
```

Cela corrige un problème lors de la construction statique de fileutils sur un système comprenant glibc 2.2.3. Si cela n'est pas fait, les programmes du package fileutils pourraient causer des erreurs de segmentation une fois chroot lancé à partir du chapitre 6.

Contenu de fileutils–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 argument et chacun de ces 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

Fileutils−4.1 nécessite que les programmes suivants soient installés:

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 Gawk–3.1.0

```
Estimation du temps de construction:           2 minutes
Estimation de l'espace disque requis:  12 Mo
```

Installation de Gawk

Avertissement: NE LANCEZ PAS **`make uninstall`** pour ce package si vous avez utilisé le patch *sed* pour changer 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/*`**.

Si vous installez Gawk en utilisant glibc–2.1.x sur votre système de base, il peut être nécessaire d'utiliser une correction pour prévenir un conflit de nom de variable. Les commandes suivantes peuvent être utilisées dans ce cas. Notez que ces commandes peuvent être aussi utilisées pour d'autres versions anciennes de glibc, donc, si vous n'êtes pas sûr, utilisez la première version.

```
cp awklib/Makefile.in awklib/Makefile.in.backup &&
sed -e '/^datadir/s/awk/gawk/' \
-e '/^libexecdir/s%/awk%%' awklib/Makefile.in.backup \
   > awklib/Makefile.in &&
export CPPFLAGS=-Dre_max_failures=re_max_failures2 &&
./configure --prefix=$LFS/usr --disable-nls \
   --libexecdir=$LFS/usr/bin &&
unset CPPFLAGS &&
make LDFLAGS=-static &&
make install
```

Si vous utilisez une version récente de glibc (2.2.x), vous pouvez utiliser les commandes suivantes pour installer Gawk:

```
cp awklib/Makefile.in awklib/Makefile.in.backup &&
sed -e '/^datadir/s/awk/gawk/' \
-e '/^libexecdir/s%/awk%%' awklib/Makefile.in.backup \
   > awklib/Makefile.in &&
./configure --prefix=$LFS/usr --disable-nls \
   --libexecdir=$LFS/usr/bin &&
make LDFLAGS=-static &&
make install
```

Contenu de gawk–3.1.0

Pas encore vérifié.

Dépendances

Gawk–3.1.0 nécessite que les programmes suivants soient installés :

Aucune dépendance encore vérifiée.

Installer GCC–2.95.3

```
Estimation du temps de construction:           22 minutes
Estimation de l'espace disque requis:  168 Mo
```

Installation de GCC

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 nous vous recommendons de supprimer CFLAGS, CXXFLAGS et toute autre variable/option qui pourrait changer l'optimisation par défaut du package.

Installez GCC en lançant les commandes suivantes:

```
patch -Np1 -i ../gcc-2.95.3-2.patch &&
mkdir ../gcc-build &&
cd ../gcc-build &&
../gcc-2.95.3/configure --prefix=/usr --enable-languages=c,c++ \
   --disable-nls --disable-shared --enable-threads=posix &&
make BOOT_LDFLAGS=-static bootstrap &&
make prefix=$LFS/usr install &&
cd $LFS/lib &&
ln -sf ../usr/bin/cpp &&
cd $LFS/usr/lib &&
ln -sf ../bin/cpp &&
cd $LFS/usr/bin &&
ln -sf gcc cc
```

Explication des commandes

`patch -Np1 -i ../gcc-2.95.3-2.patch`: Ce nouveau patch traite la gestion incorrecte des symboles faibles, la sur–optimisation des appels vers ces symboles, un problème sur atexit et le symbole __dso_handle requis pour la fonction propre de atexit.

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

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

`--enable-languages=c,c++`: Ceci ne crée que les compilateurs C et C++, et pas les autres compilateurs disponibles, parce qu'ils sont en moyenne très peu utilisés. Si certains de ces autres compilateurs sont nécessaires, le paramètre ––enable–languages peut être omis.

`--enable-threads=posix`: Ceci permet la gestion des exceptions C++ pour le code multithread.

`ln -sf ../usr/bin/cpp`: Ceci crée le lien symbolique $LFS/lib/cpp. Certains packages essaient explicitement de trouver cpp dans /lib.

`ln -sf ../bin/cpp`: Ceci crée le lien symbolique $LFS/usr/lib/cpp, parce qu'il y a des packages qui s'attendent à ce que cpp soit dans /usr/lib.

Contenu de gcc–2.95.3

Programmes

c++, c++filt, cc (lien vers gcc), cc1, cc1plus, collect2, cpp, cpp0, g++, gcc, gcov, protoize et unprotoize

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

c++filt est utilisé pour filtrer les symboles C++.

collect2

Aucune description actuellement disponible.

cpp, cpp0

cpp réalise un travail préliminaire sur un fichier source, comme inclure le contenu des fichiers d'entêtes dans ce code source. C'est une bonne idée de ne pas faire ceci manuellement pour gagner beaucoup de temps. Il suffit d'insérer une ligne comme #include <filename>. Ce préprocesseur insère le contenu de ce fichier dans le fichier source. C'est une des choses que réalise le préprocesseur.

gcov

Aucune description actuellement disponible.

protoize

Programme optionnel qui transforme des fonctions ou des définitions pré–ANSI dans le nouveau style des prototypes C ANSI (le fichier par défaut à rechercher est `/usr/lib/gcc-lib/<arch>/<version>/SYSCALLS.c.X`)

unprotoize

Programme optionnel qui transforme des prototypes réalisés avec protoize vers l'ancien style pré–ANSI (job correct seulement quand conversion avant protoize)

Bibliothèques

libgcc.a, libiberty.a, libstdc++.[a,so]

libgcc

libgcc.a est une bibliothèque utilisée par gcc lors de son exécution. La plupart du temps, sur la plupart des machines, libgcc.a n'est pas réellement nécessaire.

libiberty

libiberty est une collection de sous−routines utilisées par différents programmmes 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 utilisé 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.

Dépendances

GCC−2.95.3 nécessite que les programmes suivants soient installés:

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:           1 minute
Estimation de l'espace disque requis:  4 Mo
```

Installation of Grep

Si vous installez Grep en utilisant glibc–2.1.x sur votre système de base, il peut être nécessaire d'utiliser une correction pour prévenir un conflit de nom de variable. Les commandes suivantes peuvent être utilisées dans ce cas. Notez que ces commandes peuvent aussi être utilisées pour d'autres versions anciennes de glibc, donc, si vous n'êtes pas sûr, utilisez la première version.

```
export CPPFLAGS=-Dre_max_failures=re_max_failures2 &&
./configure --prefix=$LFS/usr --disable-nls --bindir=$LFS/bin &&
unset CPPFLAGS &&
make LDFLAGS=-static &&
make install
```

Si vous utilisez une version récente de glibc (2.2.x), vous pouvez utiliser les commandes suivantes pour installer Gawk:

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

Contenu de grep–2.4.2

Programmes

egrep, fgrep 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

Grep–2.4.2 nécessite que les programmes suivants soient installés :

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:           1 minute
Estimation de l'espace disque requis:  2 Mo
```

Installation de Gzip

Avant que Gzip soit installé, le patch doit être appliqué. Ce patch est nécessaire pour éviter un conflit de noms de variables avec les systèmes Glibc–2.0 lors de la compilation et de l'édition de liens statiques, seulement si votre système de base tourne avec Glibc–2.0. Toutefois, il est préférable d'utiliser ce patch même si vous utilisez une version différente de Glibc. Si vous n'êtes pas sûr, appliquez le patch.

Appliquer le patch en lançant la commande suivante:

```
patch -Np1 -i ../gzip-1.2.4a.patch
```

Installer Gzip en lançant les commandes suivantes:

```
./configure --prefix=$LFS/usr &&
make LDFLAGS=-static &&
make install &&
cp $LFS/usr/bin/gunzip $LFS/usr/bin/gzip $LFS/bin &&
rm $LFS/usr/bin/gunzip $LFS/usr/bin/gzip
```

Explication des commandes

`cp $LFS/usr/bin/gunzip $LFS/usr/bin/gzip $LFS/bin && rm $LFS/usr/bin/gunzip $LFS/usr/bin/gzip`: Gzip étant lié à gunzip, nous ne pouvons pas utiliser simplement << mv>> pour déplacer les fichiers. Sur certaines anciennes distributions, vous ne pouvez pas déplacer un lien entre différentes partitions (et il est tout à fait possible que $LFS et $LFS/usr soient des partitions séparées). Avec des distributions plus modernes, ceci n'est pas un problème. Si vous lancez mv pour déplacer des liens entre partitions, un simple << cp>> sera utilisé et annulera le lien. Mais, nous ne pouvons pas supposer que chaque hôte dispose d'un noyau et d'un package fileutils assez récents fonctionnant de cette manière.

Contenu de gzip–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 le codage Lempel−Ziv (LZ77).

zcat

zcat décompresse une liste de fichiers données en ligne de commande ou utilise l'entrée standard en lecture et écrit les données décompressées sur la sortie standard.

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

Gzip−1.2.4a nécessite que les programmes suivants soient installés:

bash: shbinutils: as, ld, nmfileutils: chmod, cp, install, ln, mv, rmgcc: cc1, collect2, cpp, cpp0, gccgrep: egrep, grepmake: r

Installer Linux Kernel–2.4.18

```
Estimation du temps de construction:           1 minute
Estimation de l'espace disque requis:  132 Mo
```

Installation du noyau Linux

Nous n'allons pas encore compiler une nouvelle image du noyau. Nous le ferons après avoir fini l'installation des logiciels du système de base dans ce chapitre. Mais, parce que certains logiciels ont besoin des fichiers d'en–tête du noyau, nous allons désarchiver le noyau maintenant et le configurer pour pouvoir compiler les packages ayant besoin du noyau.

Le fichier de configuration du noyau est créé en exécutant la commande suivante:

```
make mrproper &&
yes "" | make config &&
make dep &&
mkdir $LFS/usr/include/asm &&
cp include/asm/* $LFS/usr/include/asm &&
cp -R include/linux $LFS/usr/include
```

Explication des commandes

`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.

`yes "" | make config:` Ceci exécute make config et donne la réponse par défaut à toutes les questions que le script de configuration pose à l'utilisateur (il fait ceci en se contentant de transmettre l'équivalent de l'appui sur la touche Enter, qui accepte les réponses par défaut Y et N aux questions). Nous ne configurons pas ici le véritable noyau; nous n'avons besoin que d'un fichier de configuration quelconque, pour pouvoir ensuite exécuter make dep qui créera de nouveaux fichiers d'entête dans `include/linux`, comme version.h, entre autres, dont nous aurons besoin plus tard dans chroot pour compiler Glibc ainsi que d'autres packages.

`make dep:` make dep vérifie les dépendances et crée le fichier des dépendances. Nous n'avons pas vraiment besoin de la vérification des dépendances, mais ce qui nous importe est que make dep crée les fichiers susmentionnés dans `include/linux`, dont nous aurons besoin plus tard.

`mkdir $LFS/usr/include/asm` and **`cp include/asm/* $LFS/usr/include/asm`**: Ceci commande copie les fichiers d'entête assembleur du kernel dans `$LFS/usr/include/asm`.

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

Pourquoi nous copions les en–têtes 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 et asm vers respectivement /usr/src/linux/include/linux et asm. Ceci est une *mauvaise* idée d'après cet extrait d'un envoi 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
en-têtes du kernel 2.2.13, même si je n'ai pas lancé cette version du kernel 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 en–têtes doivent être *ceux avec lesquels glibc a été compilé*. Ces en–têtes 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`.

Contenu de kernel–2.4.17

Programmes

le noyau linux et les en–têtes du noyau linux

Descriptions

Noyau linux

Le noyau de 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 remplacés lors d'une mise à jour du kernel. Ils sont essentiels pour compiler un grand nombre de logiciels.

Dépendances

Linux–2.4.17 nécessite que les programmes suivants soient installés :

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 Make–3.79.1

```
Estimation du temps de construction:           1 minute
Estimation de l'espace disque requis:  6 Mo
```

Installation de Make

Installez Make en lançant les commandes suivantes:

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

Lors de la phase du `make install', vous allez voir ce message:

```
chgrp: changing group of `/mnt/lfs/usr/bin/make': Operation not permitted
/mnt/lfs/usr/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/usr/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 (que vous utilisez pour indiquer à make de ne pas démarrer certains travaux quand une certaine charge du système est atteinte).

Contenu de make–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

Make–3.79.1 nécessite que les programmes suivants soient installés :

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:           1 minute
Estimation de l'espace disque requis:  2 Mo
```

Installation de Patch

Installez Patch en lançant les commandes suivantes:

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

Explications des commandes

`CPPFLAGS=-D_GNU_SOURCE:` Ajoute **`-D_GNU_SOURCE`** à la commande CPPFLAGS avant de configurer l'installation des patchs pour les packages des plateformes PPC et m68k dont nous sommes au courant. Comme cela ne gêne pas la compilation sur les autres plateformes, telle que x86, nous le faisons pas par défaut.

Contenu de patch–2.5.4

Programmes

patch

Descriptions

patch

Le programme patch modifie un fichier en fonction d'un fichier de patch. Un fichier de patch 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 1Mo. 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'1Mo ou juste sous la forme d'un fichier de patch d'1Ko qui viendra mettre à jour la précédente version pour la rendre identique à la dernière version réalisée. Ainsi, si la première version avait déjà été téléchargée, un fichier de patch évite un deuxième téléchargement conséquent.

Dépendances

Patch–2.5.4 nécessite que les programmes suivants soient installés:

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:           1 minute
Estimation de l'espace disque requis:  2 Mo
```

Installation de Sed

Lors de l'installation de Sed avec glibc–2.1.x à partir de votre système de base, il peut être nécessaire d'utiliser une correction pour empêcher un conflit de noms de variables. Les commandes suivantes peuvent être utilisées dans ce cas. Notez que ces commandes peuvent aussi être utilisées avec d'autres versions de glibc, donc si vous n'êtes pas sûr, essayez la première version.

```
export CPPFLAGS=-Dre_max_failures=re_max_failures2 &&
./configure --prefix=$LFS/usr --bindir=$LFS/bin &&
unset CPPFLAGS &&
make LDFLAGS=-static &&
make install
```

Si vous utilisez une version plus récente de glibc (2.2.x), vous pouvez utiliser les commandes suivantes pour installer Sed:

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

Contenu de sed–3.02

Programmes

sed

Descriptions

sed

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

Dépendances

Sed–3.02 nécessite que les programmes suivants soient installés :

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:           2 minutes
Estimation de l'espace disque requis:  23 Mo
```

Installation de Sh–utils

Avant d'installer Sh–utils, il sera peut–être nécessaire d'appliquer un patch. Ce patch sert à éviter un conflit de noms de variables avec certaines versions de Glibc (habituellement glibc–2.1.x) lors de la compilation de sh–utils en statique. Néanmoins, il est préférable d'installer le patch même si vous utilisez une autre version de glibc, donc si vous n'êtes pas sûr, il est mieux de l'appliquer.

Installez le patch en lançant la commande suivante:

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

Installez Shellutils en lançant les commandes suivantes:

```
./configure --prefix=$LFS/usr --disable-nls &&
make LDFLAGS=-static &&
make install &&
cd $LFS/usr/bin &&
mv basename date echo false hostname $LFS/bin &&
mv pwd sleep stty test true uname $LFS/bin &&
mv chroot ../sbin
```

Durant l'étape "make install", vous allez voir les messages suivants:

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

Vous pouvez ignorer en toute sécurité ces messages. Vous avez 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 le réinstallerons lors du chapitre 6, vous pouvez prétendre ne pas avoir vu ce message.

Contenu de sh–utils–2.0

Programmes

basename, chroot, date, dirname, echo, env, expr, factor, false, groups, hostid, hostname, 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 l'heure et la date courantes 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 le groupe auquel appartient l'utilisateur.

hostid

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

hostname

hostname met à jour ou affiche le nom courant de l'hôte.

id

id affiche les UIDs et GIDs, réels et effectifs, d'un utilisateur ou de l'utilisateur courant.

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 standard qui ne sera pas de type tty.

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 dans 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 l'identifiant effectif de l'utilisateur.

yes

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

Dépendances

Sh−utils−2.0 nécessite que les programmes suivants soient installés :

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:           1 minute
Estimation de l'espace disque requis:  7 Mo
```

Installation de Tar

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

Installer le patch en lançant la commande suivante:

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

Installer Tar en lançant les commandes suivantes:

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

Contenu de tar−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

Tar−1.13 nécessite que les programmes suivants soient installés :

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–4.1

```
Estimation du temps de construction:           1 minute
Estimation de l'espace disque requis:  11 Mo
```

Installation de Texinfo

Installez Texinfo en lançant les commandes suivantes:

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

Contenu de texinfo–4.0

Programmes

info, 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.

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, il convient également de 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

Texinfo−4.0 nécessite que les programmes suivants soient installés:

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–2.0

```
Estimation du temps de construction:           2 minutes
Estimation de l'espace disque requis:  24 Mo
```

Installation df Textutils

Installer Textutils en lançant les commandes suivantes:

```
./configure --prefix=$LFS/usr --disable-nls &&
make LDFLAGS=-static &&
make install &&
mv $LFS/usr/bin/cat $LFS/usr/bin/head $LFS/bin
```

Contenu de textutils–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 l'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 extrait d'un fichier des fragments séparés par un ou plusieurs motifs, dans des fichiers xx01, xx02, ..., et affiche sur la sortie standard le compte d'octets de chaque segment.

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, et 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

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 chaque fichier spécifié, et une ligne de totaux si plusieurs fichiers ont été spécifiés.

Dépendances

Textutils−2.0 nécessite que les programmes suivants soient installés :

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

Créer les fichiers passwd et group

Pour que l'utilisateur et le groupe root soient reconnus et puissent se connecter, il faut une entrée correspondante dans les fichiers /etc/passwd et /etc/group. En outre, le groupe root ainsi que quelques autres sont recommandés et nécessaires pour certains packages. Les groupes et leur GID ci–dessous ne font partie d'aucun standard. Le LSB recommande uniquement un groupe bin avec un GID de 1 à côté d'un groupe root. Les noms et GID des autres groupes peuvent être choisis par l'utilisateur. Les packages bien écrits ne dépendent pas du numéro de GID, mais seulement du nom du groupe, car le GID d'un groupe importe peu. Comme il n'y a aucun standard pour les groupes, je ne suivrai pas les conventions utilisées par Debian, RedHat ou d'autres. Les groupes ajoutés ici sont ceux mentionnés dans le script MAKEDEV (le script qui crée les fichiers de périphériques dans le répertoire /dev).

Créez un nouveau fichier `$LFS/etc/passwd` en lançant la commande suivante:

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

Créez un nouveau fichier `$LFS/etc/group` en lançant ce qui suit:

```
cat > $LFS/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
```

Copie des anciennes bibliothèques NSS

Si votre système Linux de base utilise Glibc−2.0, vous avez besoin de copier les bibliothèques NSS dans la partition LFS. Certains programmes liés statiquement dépendent des bibliothèques NSS, spécialement les programmes qui ont besoin de récupérer les noms, les ID des utilisateurs et des groupes. Vous pouvez vérifier la version de la bibliothèque C que votre système Linux utilise en exécutant simplement la bibliothèque ainsi:

```
/lib/libc.so.6
```

La première ligne vous donne le numéro de version. Les lignes suivantes contiennent d'autres informations intéressantes. Si Glibc−2.0.x est installé sur votre système de base, copiez les bibliothèques NSS en lançant:

```
cp -av /lib/libnss* $LFS/lib
```

Monter le système de fichiers $LFS/proc

Pour que certains programmes fonctionnent correctement, le système de fichiers proc doit être monté et accessible depuis l'environnement chrooté. Il n'y a pas de problèmes à monter le système de fichiers proc (ou n'importe quel autre système de fichiers) deux fois ou plus.

Si vous êtes toujours connecté en tant qu'utilisateur "lfs", vous devez vous déconnecter et vous connecter en tant qu'utilisateur root. La raison pour ceci est simple : seul root est autorisé à monter des systèmes de fichiers et à utiliser chroot.

Le système de fichiers proc est monté sous $LFS/proc en lançant la commande suivante. Nous allons aussi modifier l'utilisateur et le groupe en root:root en même temps (le reste du système de fichier sera attribué à l'utilisateur root et au groupe root dans une minute lorsque nous commencerons le chapitre 6).

```
chown root.root $LFS/proc &&
mount proc $LFS/proc -t proc
```

Chapitre 6. Installation des logiciels de base du système

Introduction

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 chaque package et de seulement expliquer comment installer quelque chose si un certain package demande une méthode d'installation alternative. Même si je suis d'accord avec ceci, j'ai choisi de donner les instructions complètes pour chaque package. Il s'agit simplement d'éviter toute confusion ou erreur.

Ce serait le bon moment pour jeter un oeil aux astuces d'optimisation dans http://hints.linuxfromscratch.org/hints/optimization.txt si vous prévoyez d'utiliser des options d'optimisation lors la compilation des packages du chapitre qui suit. 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.

A propos des symboles de débogage

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

Un programme compilé avec des symboles de débogage permet à un utilisateur d'exécuter un programme ou une bibliothèque via un débogueur et d'utiliser facilement les résultats obtenus. Ces symboles de débogage augmentent également la taille du programme de façon significative.

Avant que vous ne commenciez à vous demander si ces symboles de débogage font réellement une grande différence, voici quelques statistiques. Utilisez–les pour en tirer vos propres conclusions.

- Un binaire Bash dynamique avec symboles de débogage: 1,2Mo.
- Un binaire Bash dynamique sans symboles de débogage: 478Ko.
- /lib et /usr/lib (fichiers glibc et gcc) avec symboles de débogage: 87Mo.
- /lib et /usr/lib (fichiers glibc et gcc) sans symboles de débogage: 16Mo.

Les tailles varient selon le compilateur utilisé et la version de la bibliothèque C utilisée pour les liens dynamiques entre les programmes, mais les résultats seront similaires si vous comparez des programmes avec ou sans symboles de débogage.

Pour enlever les symboles de débogage d'un binaire (qui doit être de type a.out ou ELF), exécutez **`strip --strip-debug filename`**. Les jokers peuvent être utilisés pour supprimer les symboles de débogage dans plusieurs fichiers (utilisez quelque chose comme **`strip --strip-debug $LFS/usr/bin/*`**). La plupart des gens n'utiliseront probablement jamais un débogueur sur leur logiciels, vous gagnerez donc beaucoup de place sur le disque en supprimant ces symboles.

Pour vous faciliter l'opération, le chapitre 9 comprend une commande unique pour supprimer tous les symboles de débogage de tous les programmes et bibliothèques de votre système.

Vous obtiendrez des informations supplémentaires dans le guide de l'optimisation qui peut être trouvé à http://hints.linuxfromscratch.org/hints/optimization.txt .

Créer $LFS/root/.bash_profile

Quand vous serez entré dans l'environnement chroot à la section suivante, nous exporterons quelques variables d'environnement vers cet interpréteur de commande telles que PS1, PATH et d'autres qu'il est bien d'avoir initialisées. Dans ce but, nous allons créer le fichier $LFS/root/.bash_profile qui sera lu par bash quand nous entrerons dans l'environnement chroot.

Créez un nouveau fichier `$LFS/root/.bash_profile` en exécutant ce qui suit.

```
cat > $LFS/root/.bash_profile << "EOF"
# Begin /root/.bash_profile

PS1='\u:\w\$ '
PATH=/bin:/usr/bin:/sbin:/usr/sbin

export PS1 PATH

# End /root/.bash_profile
EOF
```

La variable PS1 est une variable d'environnement qui contrôle l'apparence de l'invite de commande. Référez–vous aux pages de manuel de bash (man bash) pour les détails de sa construction. Les variables d'environnement supplémentaires, alias etc... nécessités ou voulus peuvent être rajoutés par vos soins.

Entrée dans l'environnement "chrooté"

C'est le moment d'entrer dans l'environnement chrooté afin d'installer le reste des programmes nécessaires.

Tapez les commandes suivantes pour entrer dans l'environnement chrooté. A partir de maintenant, il n'est plus besoin d'utiliser la variable $LFS puisque tout ce que l'utilisateur fait sera limité à la partition LFS (car / est en fait /mnt/lfs mais ça, l'interpréteur de commande ne le sait pas).

```
cd $LFS &&
chroot $LFS /usr/bin/env -i HOME=/root \
   TERM=$TERM /bin/bash --login
```

L'option –i efface toutes les variables d'environnement pour le temps que l'on sera en environnement chrooté et seules les variables HOME et TERM sont initialisées. La commande TERM=$TERM initialise la variable TERM à l'intérieur de chroot à la même valeur que celle à l'extérieur de chroot qui est utilisée par des programmes comme vim. Si vous avez besoin d'autres variables, telles que CFLAGS ou CXXFLAGS, initialisez–les de même.

La raison pour laquelle nous faisons un **`cd $LFS`** avant d'exécuter la commande **`chroot`** est que d'anciens packages sh–utils ont un programme chroot qui ne fait pas le cd de lui même, ce qui veut dire qu'il faut le faire manuellement. Bien que cela ne soit pas un problème avec les distributions modernes, cela ne fait pas de mal et puis cela assure que la commande fonctionne pour tout le monde.

Maintenant que nous sommes dans l'environnement chrooté, nous pouvons continuer à installer tous les programmes de base du système. Assurez–vous que les commandes suivantes dans ce chapitre et les suivants sont exécutées dans l'environnement chrooté. Si vous quittez l'environnement chrooté (en redémarrant l'ordinateur par exemple) rappellez–vous de monter $LFS/proc de nouveau et de retourner dans l'environnement chrooté 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 le propriétaire de la partition LFS

Maintenant que nous sommes en chroot, c'est le bon moment d'assigner l'utilisateur root comme le propriétaire de tous les fichiers et répertoires qui ont été installés au chapitre 5. Pour cela, lancez les commandes suivantes :

```
cd / &&
chown 0.0 . proc &&
chown -R 0.0 bin boot dev etc home lib mnt opt root sbin tmp usr var
```

Selon le système de fichiers que vous avez créé pour votre partition LFS, vous pouvez avoir un répertoire /lost+found. Si c'est le cas, alors lancez :

```
chown 0.0 lost+found
```

Ces commandes réassigneront l'utilisateur root comme le propriétaire de la partition root et du répertoire `/proc`, plus tout ce qui se trouve sous les répertoires mentionnés dans la 2e ligne. Dans ces commandes, 0.0 est utilisé plutôt que le root.root courant, car l'utilisateur root ne peut pas être résolu puisque glibc n'a pas été encore installé.

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 -s /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 sera 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.

Installer Glibc–2.2.5

```
Estimation du temps de construction:           46 minutes
Estimation de l'espace disque requis:  350 Mo
```

Installation de Glibc

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 est meilleur tel quel, nous vous recommandons donc de régler CFLAGS, CXXFLAGS et d'autres variable/réglages du même type qui changeraient la configuration par défaut. 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:

```
mknod –m 0666 /dev/null c 1 3 &&
touch /etc/ld.so.conf &&
cp malloc/Makefile malloc/Makefile.backup &&
sed 's%\$(PERL)%/usr/bin/perl%' malloc/Makefile.backup > malloc/Makefile &&
cp login/Makefile login/Makefile.backup &&
sed 's/root/0/' login/Makefile.backup > login/Makefile &&
mkdir ../glibc–build &&
cd ../glibc–build &&
../glibc–2.2.5/configure ––prefix=/usr \
   ––enable–add–ons ––libexecdir=/usr/bin &&
echo "cross–compiling = no" > configparms &&
make &&
make install &&
make localedata/install–locales &&
exec /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 localdef. Vous trouverez des informations à ce sujet dans le fichier INSTALL de l'arborescence de glibc–2.2.5.

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

`mknod -m 0666 /dev/null c 1 3:` Glibc a besoin d'un périphérique null pour compiler correctement. Tous les autres périphériques seront créés dans la prochaine section.

`touch /etc/ld.so.conf` Une des étapes finales de l'installation de Glibc est d'exécuter ldconfig afin de mettre à jour le cache du chargeur dynamique. Si le fichier n'existe pas, l'installation échouera retournant que le fichier ne peut être lu, donc nous avons simplement créé un fichier vide (avec ce fichier vide, Glibc utilisera par défaut /lib et /usr/lib, ce qui est satisfaisant).

`sed 's%\$(PERL)%/usr/bin/perl%' malloc/Makefile.backup > malloc/Makefile:` Cette commande sed remplace dans `malloc/Makefile.backup` toutes les occurences de `$(PERL)` en `/usr/bin/perl`. La sortie est alors écrite sur le fichier original `malloc/Makefile.in` qui est utilisé durant la configuration. Ceci est fait parce que Glibc ne peut détecter automatiquement perl puisqu'il n'a pas encore été installé.

`sed 's/root/0' login/Makefile.backup > login/Makefile:` Cette commande sed remplace toutes les occurences du mot `root` dans le fichier `login/Makefile.backup` par 0. Glibc n'étant pas encore installé sur le système LFS, les noms d'utilisateur ne peuvent pas encore être remplacés par leur numéros d'identifiant. Donc, nous remplaçons le nom d'utilisateur root par l'identifiant de root, qui est 0.

`--enable-add-ons:` Permet l'utilisation des linuxthreads que nous installons avec Glibc.

`--libexecdir=/usr/bin:` Cela force l'installation du programme pt_chown dans /usr/bin directory.

`echo "cross-compiling = no" > configparms:` Nous faisons ceci parce que nous créons seulement notre propre système. La cross–compilation est utilisée pour construire un package pour un Apple Power PC sur un système Intel. La raison pour laquelle Glibc pense que nous faisons de la cross–compilation est qu'il ne peut compiler un programme de test pour s'en apercevoir, donc il est par défaut en cross–compilation. Compiler le programme de test échoue car Glibc n'a pas encore été installé.

`exec /bin/bash:` 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 installé).

Contenu de glibc–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

Aucune description actuellement disponible.

gencat

gencat génère des catalogues de messages.

getconf

Aucune description actuellement disponible.

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 requises par chaque programme ou bibliothèque partagée spécifié sur la ligne de commande.

lddlibc4

Aucune description actuellement disponible.

locale

Aucune description actuellement disponible.

localedef

localedef compile les spécifications pour locale.

mtrace

Aucune description actuellement disponible.

nscd

nscd est un démon procurant un cache pour les requêtes DNS les plus courantes.

nscd_nischeck

Aucune description actuellement disponible.

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

Aucune description actuellement disponible.

rpcinfo

Aucune description actuellement disponible.

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 librairies partagées.

libBrokenLocale, libBrokenLocale_p

Aucune description actuellement disponible.

libSegFault

Aucune description actuellement disponible.

libanl, libanl_p

Aucune description actuellement disponible.

libbsd−compat

Aucune description actuellement disponible.

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 fourni 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

Aucune description actuellement disponible.

libg

Aucune description actuellement disponible.

libieee

Aucune description actuellement disponible.

libm, libm_p

libm est la bibliothèque de mathématiques.

libmcheck

Aucune description actuellement disponible.

libmemusage

Aucune description actuellement disponible.

libnsl, libnsl_p

Aucune description actuellement disponible.

libnss_compat, libnss_dns, libnss_files, libnss_hesiod, libnss_nis, libnss_nisplus

Aucune description actuellement disponible.

libpcprofile

Aucune description actuellement disponible.

libpthread, libpthread_p

Aucune description actuellement disponible.

libresolv, libresolv_p

Aucune description actuellement disponible.

librpcsvc, librpcsvc_p

Aucune description actuellement disponible.

librt, librt_p

Aucune description actuellement disponible.

libthread_db

Aucune description actuellement disponible.

libutil, libutil

Aucune description actuellement disponible.

Dépendances

Glibc−2.2.5 nécessite que les programmes suivants soient installés :

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

Créer les périphériques (Makedev−1.4)

```
Estimation du temps de construction:           1 minute
Estimation de l'espace disque requis:  57 Ko
```

Créer les périphériques

Note: le fichier MAKEDEV−1.4.bz2 que vous avez déballé n'est pas une archive, alors il ne créera pas de répertoire dans lequel vous pourrez entrer.

Créez les fichiers périphériques en exécutant les commandes suivantes:

```
cp MAKEDEV-1.4 /dev/MAKEDEV &&
cd /dev &&
chmod 754 MAKEDEV
```

Maintenant, suivant que vous allez utiliser devpts ou non, vous pouvez exécuter l'une des commandes:

Si vous ne comptez pas utiliser devpts, lancez:

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

Si vous comptez utiliser devpts, alors lancez:

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

Notez qui 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 max. de partitions.

Explication des commandes

`./MAKEDEV -v generic:` Ceci crée les périphériques génériques. Normalement, ces périphériques sont tous ceux dont vous avez besoin. Il est possible qu'il vous manque certains périphériques spéciaux, qui sont nécessaires pour votre configuration matérielle. Créez−les avec ./MAKEDEV −v <device>. L'option **`generic-nopty`** effectue un travail similaire mais évite certains périphériques inutiles si vous utilisez devpts.

Contenu de MAKEDEV−1.4

Programmes

MAKEDEV

Descriptions

MAKEDEV

MAKEDEV est un script qui peut aider à créer 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

MAKEDEV−1.4 nécessite que les programmes suivants soient installés:

bash: sh fileutils: chmod, chown, cp, ln, mknod, mv, rm grep: grep sh−utils: expr, id

Installer Man−pages−1.48

```
Estimation du temps de construction:           1 minute
Estimation de l'espace disque requis:  5 Mo
```

Installation de Man−pages

Installez Man−pages en lançant la commande suivante:

```
make install
```

Contenu de manpages−1.47

Fichiers support

Plusieurs pages de manuel qui ne viennent pas avec les packages.

Descriptions

manual pages

Les exemples de pages de manuel fournies décrivent toutes les fonctions C et C++, quelques fichiers importants /dev/ et autres.

Dépendances

Man−pages−1.47 nécessite que les programmes suivants soient installés:

bash: sh fileutils: install make: make

Installer Findutils–4.1

```
Estimation du temps de construction:           1 minute
Estimation de l'espace disque requis:  3 Mo
```

Installer Findutils

Avant que Findutils ne soit installé, il faut que le fichier patch de findutils soit déballé.

Installez Findutils en lançant les commandes suivantes:

```
patch -Np1 -i ../findutils-4.1.patch &&
./configure --prefix=/usr &&
make &&
make libexecdir=/usr/bin install
```

Note de conformité FHS

Par défaut, la localisation de la base de données de updatedb est dans /usr/var. Si vous préférez être conforme pour FHS, vous devriez 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 &&
make localstatedir=/var/lib/misc libexecdir=/usr/bin install
```

Explication des commandes

`patch -Np1 -i ../findutils-4.1.patch`: Ce patch est là pour réparer quelques erreurs de compilation en évitant des conflits de variables et en changeant de mauvaises syntaxes.

Contenu du package findutils–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 définis, il liste tous les fichiers du répertoire courant et de ses sous−répertoires.

frcode

updatedb exécute un programme nommé frcode 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, on peut alors créer un fichier pour contenir la liste de tous ces fichiers (un par ligne) et utiliser xargs pour exécuter cette commande sur chacun d'eux.

Dépendances

Findutils−4.1 nécessite que les programmes suivants soient installés:

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–3.1.0

```
Estimation du temps de construction:           2 minutes
Estimation de l'espace disque requis:  10 Mo
```

Installation de Gawk

Avertissement: ne lancez PAS **`make uninstall`** pour ce package si vous avez utilisé le patch *sed* pour changer 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:

```
cp awklib/Makefile.in awklib/Makefile.in.backup &&
sed -e '/^datadir/s/awk/gawk/' \
-e '/^libexecdir/s%/awk%%' awklib/Makefile.in.backup \
   > awklib/Makefile.in &&
./configure --prefix=/usr --libexecdir=/usr/bin &&
make &&
make install
```

Contenu de gawk–3.1.0

Pas encore vérifié.

Dépendances

Gawk–3.1.0 nécessite que les programmes suivants soient installés :

Aucune dépendance encore vérifiée.

Installer Ncurses–5.2

```
Estimation du temps de construction:           6 minutes
Estimation de l'espace disque requis:  29 Mo
```

Installation de Ncurses

Installez Ncurses en lançant les commandes suivantes:

```
./configure --prefix=/usr --libdir=/lib \
   --with-shared --disable-termcap &&
make &&
make install &&
cd /lib &&
mv *.a /usr/lib &&
chmod 755 *.5.2 &&
cd /usr/lib &&
ln -sf libncurses.a libcurses.a &&
ln -sf ../../lib/libncurses.so &&
ln -sf ../../lib/libcurses.so &&
ln -sf ../../lib/libform.so &&
ln -sf ../../lib/libpanel.so &&
ln -sf ../../lib/libmenu.so
```

Explication de la commande

--with-shared: Ceci valide la construction des fichiers de bibliothèques partagées ncurses.

--disable-termcap: Désactive la compilation du support termcap.

cd /lib && mv *.a /usr/lib : Ceci déplace tous les fichiers bibliothèques statiques de ncurses de /lib vers /usr/lib. /lib ne devrait contenir que les fichier partagés qui sont essentiels au système lorsque /usr peut ne pas être monté.

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 à la place.

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.

ln -sf ../../lib/libncurses.so etc: Ces liens symboliques sont créés pour améliorer l'installation. Une bonne pratique est d'avoir les fichiers *.so sous /usr/lib ainsi que sous /lib, pour s'assurer que le chargeur de lien sera capable de trouver les fichiers si il les cherche sous /lib ou /usr/lib..

Contenu

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 une entrée binaire de terminfo avec d'autres entrées 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 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 leurs valeurs par défaut avant d'initialiser le terminal de la même façon que tset.

tack

tack vérifie les actions de terminfo.

tic

Le programme tic est le compilateur des entrées de description de terminfo. Le programme traduit un fichier terminfo du 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 fournie 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

Ces bibliothèques qui forment la bibliothèque Ncurses sont utilisées pour afficher du texte à l'écran. Un exemple où ncurses est utilisé est dans le programme << make menuconfig>> . Les bibliothèques libncurses sont la base du système.

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

Ncurses−5.2 nécessite que les programmes suivants soient installés :

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:           2 minutes
Estimation de l'espace disque requis:  15 Mo
```

Installation de Vim

Si vous ne voulez pas de vim comme éditeur sur le système LFS, vous devez télécharger et installer un éditeur que vous préférez. Il y a quelques astuces sur la manière d'installer différents éditeurs disponibles à http://hints.linuxfromscratch.org/hints/. Les astuces actuellement disponibles concernent Emacs, Joe et nano.

Installez Vim en lançant les commandes suivantes:

```
./configure --prefix=/usr &&
make CPPFLAGS=-DSYS_VIMRC_FILE=\\\"/etc/vimrc\\\" &&
make install &&
cd /usr/bin &&
ln -sf vim 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 librairies soient installées. Pour plus d'informations, consultez la documentation de Vim.

Notes sur la conformité FHS

Le FHS dit que les éditeurs comme vim devraient utiliser /var/lib/<editor> pour leurs fichiers temporaires, comme les sauvegardes temporaires par exemple. Si vous voulez que Vim respecte le FHS, vous devriez utiliser les commandes suivantes à la place des précedentes.

```
./configure --prefix=/usr --localstatedir=/var/lib/vim &&
make CPPFLAGS=-DSYS_VIMRC_FILE=\\\"/etc/vimrc\\\" &&
make install &&
cd /usr/bin &&
ln -sf vim vi
```

Explication des commandes

`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

Programmes

ex (lien vers vim), rview (lien vers vim), rvim (lien vers vim), vi (lien vers vim), view (lien vers vim), vim, vimdiff (lien vers vim), vimtutor (lien vers vim) et xxd

Descriptions

ex

ex démarre vim dans le mode Ex.

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.

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.

vimdiff

vimdiff édite deux ou trois versions d'un fichier avec Vim et affiche les différences.

vimtutor

vimtutor démarre le tutoriel de Vim.

xxd

xxd crée un dump hexadécimal ou fait l'inverse.

Dépendances

Vim−6.0 nécessite que les programmes suivants soient installés:

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

Installer GCC–2.95.3

```
Estimation du temps de construction:           22 minutes
Estimation de l'espace disque requis:  148 Mo
```

Installation de GCC

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 est meilleur tel quel, donc nous vous recommandons de ne pas paramétrer CFALGS, CXXFLAGS et toutes autres variables/paramètres qui changeraient la configuration par défaut.

Installer GCC en exécutant les commandes suivantes. Ces commandes construiront les compilateurs C et C++. D'autres compilateurs sont accessibles avec le package gcc. Si vous voulez également construire tous les autres compilateurs n'entrez pas l'option ––enable–languages=c,c++ dans la commande de configuration. Voyez la documentation de GCC pour plus de détail sur les compilateurs disponibles.

Note: la construction des autres compilateurs n'est pas testée par les personnes qui travaillent sur LFS.

```
patch -Np1 -i ../gcc-2.95.3-2.patch &&
mkdir ../gcc-build &&
cd ../gcc-build &&
../gcc-2.95.3/configure --prefix=/usr --enable-shared \
   --enable-languages=c,c++ --enable-threads=posix &&
make bootstrap &&
make install &&
cd /lib &&
ln -sf ../usr/bin/cpp &&
cd /usr/lib &&
ln -sf ../bin/cpp &&
cd /usr/bin &&
ln -sf gcc cc
```

Contenu de gcc–2.95.3

Programmes

c++, c++filt, cc (lien vers gcc), cc1, cc1plus, collect2, cpp, cpp0, g++, gcc, gcov, protoize et unprotoize

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

c++filt est utilisé pour filtrer les symboles C++.

collect2

Aucune description actuellement disponible.

cpp, cpp0

cpp réalise un travail préliminaire sur un fichier source, comme inclure le contenu des fichiers d'entêtes dans ce code source. C'est une bonne idée de ne pas faire ceci manuellement pour gagner beaucoup de temps. Il suffit d'insérer une ligne comme #include <filename>. Ce préprocesseur insère le contenu de ce fichier dans le fichier source. C'est une des choses que réalise le préprocesseur.

gcov

Aucune description actuellement disponible.

protoize

Programme optionnel qui transforme des fonctions ou des définitions pré−ANSI dans le nouveau style des prototypes C ANSI (le fichier par défaut à rechercher est `/usr/lib/gcc-lib/<arch>/<version>/SYSCALLS.c.X`)

unprotoize

Programme optionnel qui transforme des prototypes réalisés avec protoize vers l'ancien style pré−ANSI (job correct seulement quand conversion avant protoize)

Bibliothèques

libgcc.a, libiberty.a, libstdc++.[a,so]

libgcc

libgcc.a est une bibliothèque utilisée par gcc lors de son exécution. La plupart du temps, sur la plupart des machines, libgcc.a n'est pas réellement nécessaire.

libiberty

libiberty est une collection de sous−routines utilisées par différents programmmes 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 utilisé 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.

Dépendances

GCC−2.95.3 nécessite que les programmes suivants soient installés:

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

Installation de Bison–1.34

```
Estimation du temps de construction :           1 minute
Estimation de l'espace disque requis :  3 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 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–1.31

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 yacc qui appelle bison avec l'option −y. Ceci est fait pour des raisons de compatibilté avec des programmes qui utilisent yacc au lieu de bison.

Dépendances

Bison−1.31 nécessite que les programmes suivants soient installés:

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:           1 minute
Estimation de l'espace disque requis:  2 Mo
```

Installation de Less

Installez Less en exécutant les commandes suivantes:

```
./configure --prefix=/usr --bindir=/bin &&
make &&
make install
```

Contenu de less–358

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 avec possibilité de défilement. 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

Less–358 nécessite que les programmes suivants soient installés:

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–1.17.2

```
Estimation du temps de construction:           2 minutes
Estimation de l'espace disque requis:  16 Mo
```

Installation de Groff

Installez Groff en exécutant les commandes suivantes:

```
./configure --prefix=/usr &&
make &&
make install &&
cd /usr/bin &&
ln -s soelim zsoelim &&
ln -s eqn geqn &&
ln -s tbl gtbl
```

Explications des commandes

`ln -s ...`: Ces liens symboliques sont nécessaires pour que des programmes tels que **`xman`** et d'autres traitant les documents groff/man, puissent fonctionner correctement.

Contenu de groff–1.17.2

Programmes

addftinfo, afmtodit, eqn, grn, grodvi, groff, grog, grolbp, grolj4, grops, grotty, hpftodit, indxbib, lkbib, lookbib, mmroff, neqn, nroff, pfbtops, pic, post–grohtml, pre–grohtml, refer, soelim, tbl, tfmtodit et troff

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.

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'.

hpftodit

hpftodit crée un fichier de font 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 en 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.

Dépendances

Groff−1.17.2 nécessite que les programmes suivants soient installés :

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 Man–1.5j

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

Installation de Man

Lancez les commandes suivantes pour installer man:

```
./configure –default &&
make &&
make install &&
mv /usr/share/misc/man.conf /etc
```

Vous voudrez peut–être jeter un oeil à l'astuce man à http://hints.linuxfromscratch.org/hints/man.txt qui traite du formatage et de la compression pour les pages de manuel.

Contenu de man–1.5j

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

Man−1.5i2 nécessite que les programmes suivants soient installés :

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 Perl–5.6.1

```
Estimation du temps de construction:           6 minutes
Estimation de l'espace disque requis:  35 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–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 awk vers perl.

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 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 modules de librairies style Perl5.

pod2html

pod2html convertit en HTML des fichiers au format pod.

pod2latex

pod2latex convertit en HTML des fichiers au format LaTeX.

pod2man

pod2man convertit en HTML des fichiers au format *roff.

pod2text

pod2text convertit des données pod en texte au format ASCII.

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.

Dépendances

Perl−5.6.1 nécessite queles programmes suivants soient installés :

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 M4−1.4

```
Estimation du temps de construction:           1 minute
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 m4−1.4

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

M4−1.4 nécessite que les programmes suivants soient installés:

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–4.1

```
Estimation du temps de construction:           1 minute
Estimation de l'espace disque requis:  10 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–4.0

Programmes

info, 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.

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, il convient également de 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

Texinfo−4.0 nécessite que les programmes suivants soient installés:

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 :           1 minute
Estimation de l'espace disque requis :  4 Mo
```

Installation de Autoconf

Installez Autoconf en lançant les commandes suivantes:

```
./configure --prefix=/usr &&
make &&
make install
```

Contenu de autoconf–2.52

Programmes

autoconf, autoheader, 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.

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 néssaire) 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 dont la racine est passée en argument de ligne de commande ou le répertoire courant si aucun n'est fourni. 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 les conditions 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

Autoconf−2.52 nécessite que les programmes suivants soient installés:

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

```
Estimation du temps de construction :           1 minute
Estimation de l'espace disque requis :  3 Mo
```

Installation de Automake

Installez Automake en lançant les commandes suivantes:

```
./configure --prefix=/usr &&
make install
```

Contenu de automake–1.5

Programmes

aclocal et automake

Descriptions

aclocal

Automake inclut un certain nombre de macros Autoconf qui peuvent être utilisées dans les packages; certaines d'entre elles sont actuellement requises 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

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.

Dépendances

Automake−1.5 nécessite que les programmes suivants soient installés:

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 :           3 minutes
Estimation de l'espace disque requis :  19 Mo
```

Installation de Bash

Installez Bash en lançant les commandes suivantes:

```
./configure --prefix=/usr --with-curses \
   --bindir=/bin &&
make &&
make install &&
cd /bin &&
ln -sf bash sh &&
exec /bin/bash --login
```

Contenu de bash–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

Bash–2.05a nécessite que les programmes suivants soient installés:

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 Flex–2.5.4a

```
Estimation du temps de construction:           1 minute
Estimation de l'espace disque requis:  3MB
```

Installation de Flex

Installez Flex en exécutant les commandes suivante:

```
./configure --prefix=/usr &&
make &&
make install
```

Quelques programmes ne reconnaissent pas flex et essaient de trouver le programme lex (flex est une (meilleure) alternative à lex). Donc pour satisfaire ces quelques programmes nous devrons créer un script lex, qui appelle flex et qui induit que celui–ci imite lex.

Créez un nouveau fichier `/usr/bin/lex` en exécutant ce qui suit:

```
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–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 utilisée exclusivement par les scanners C++.

lex

Nous créons un script yacc qui appelle 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

Aucune description actuellement disponible.

Dépendances

Flex−2.5.4a nécessite que les programmes suivants soient installés:

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

Installer File–3.37

```
Estimation du temps de construction:           1 minute
Estimation de l'espace disque requis:  2 Mo
```

Installation de File

Installez File en lançant les commandes suivantes:

```
touch aclocal.m4 configure Makefile.in stamp-h.in &&
./configure --prefix=/usr --datadir=/usr/share/misc &&
make &&
make install
```

Explications des commandes

`touch aclocal.m4 configure Makefile.in stamp-h.in`: Cette commande permet de contourner une erreur, qui arrive lors d'une compilation si automake–1.5 est installé, en changeant la date de modification de certains fichiers pour aujourd'hui. En changeant cette date, make va penser que les fichiers sont déjà à jour et qu'ils n'ont pas à être créer de nouveau.

Contenu de file–3.37

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

File–3.37 nécessite que les programmes suivants soient installés :

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:           1 minute
Estimation de l'espace disque requis:  5 Mo
```

Installation de Libtool

Installez Libtool en exécutant les commandes suivantes:

```
./configure --prefix=/usr &&
make &&
make install
```

Contenu de libtool–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,so]

Descriptions

libltdl

Libtool fournit une petite bibliothèque, appelée 'libltdl', qui a pour but de masquer aux programmeurs les différentes difficultés liées à l'ouverture dynamique des bibliothèques.

Dépendances

Libtool–1.4.2 nécessite que les programmes suivants soient installés:

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 Bin86–0.16.2

```
Estimation du temps de construction :           1 minute
Estimation de l'espace disque requis :  1 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 bin86–0.16.0

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 comprend que les fichiers objets produits par l'assembleur as86. Il peut les lier dans un exécutable I&D soit impur, soit séparé.

nm86

Aucune description actuellement disponible.

objdump86

Aucune description actuellement disponible.

size86

Aucune description actuellement disponible.

Dépendances

Bin86−0.16.0 nécessite que les programmes suivants soient installés:

bash: sh binutils: as, ld, strip fileutils: chmod, install, ln, mv gcc: cc, cc1, collect2, cpp0
make: make sed: sed

Installation de Binutils–2.12

```
Estimation du temps de construction :           6 minutes
Estimation de l'espace disque requis :  85 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 nous vous recommendons de supprimer CFLAGS, CXXFLAGS et toute autre variable/option qui pourrait changer l'optimisation par défaut du package.

Installez Binutils en lançant les commandes suivantes:

```
mkdir ../binutils-build &&
cd ../binutils-build &&
../binutils-2.12/configure --prefix=/usr --enable-shared &&
make tooldir=/usr &&
make tooldir=/usr install &&
make tooldir=/usr install-info
```

Explications des commandes

`make tooldir=/usr install-info`: Ceci installera les pages d'informations de binutil.

Contenu de binutils–2.11.2

Programmes

addr2line, ar, as, c++filt, 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.

c++filt

Le langage C++ fournit un mécanisme de surcharge, ce qui signifie qu'il est possible d'écrire plusieurs fonctions ayant le même nom (pourvu que chacune demande des paramètres de types différents). Tous les noms de fonction C++ sont encodés dans un nom d'assemblage de bas niveau (ce processus est connu sous le nom de broyage). Le programme c++filt fait l'association inverse : il décode (dé−broie) les noms de bas niveau en noms de niveau utilisateur, pour que l'éditeur de liens puisse prévenir un conflit entre ces fonctions surchargées.

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 objfile 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.

Fichiers de Librairies

libbfd.a, libiberty.a et libopcodes.a

Descriptions

libbfd

libbfd est une librairie de description des fichiers binaires.

libiberty

libiberty est un ensemble de sous−routines utilisé dans de divers programmes GNU dont getopt, obstack, strerror, strtol et strtoul.

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

Binutils−2.11.2 nécessite que les programmes suivants soient installés:

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

Installation de Bzip2–1.0.2

```
Estimation du temps de construction :           1 minute
Estimation de l'espace disque requis :  2 Mo
```

Installation de Bzip2

Installez Bzip2 en lançant les commandes suivantes:

```
make -f Makefile-libbz2_so &&
make &&
rm /usr/bin/bz* &&
make PREFIX=/usr install &&
cp bzip2-shared /bin/bzip2 &&
ln -s libbz2.so.1.0 libbz2.so &&
cp -a libbz2.so* /lib &&
cd /usr/lib &&
ln -sf ../../lib/libbz2.so &&
cd /usr/bin &&
rm bunzip2 bzcat bzip2 &&
rm /bin/bzless /bin/bzmore &&
mv bzip2recover bzless bzmore /bin &&
cd /bin &&
ln -sf bzip2 bunzip2 &&
ln -sf bzip2 bzcat
```

Bien que ce ne soit pas strictement une partie d'un système LFS de base, il est bon de préciser qu'un patch pour Tar peut être téléchargé pour permettre au programme tar de compresser et de décompresser facilement en utilisant bzip2/bunzip2. Avec un tar normal, vous devez utiliser des constructions comme "bzcat file.tar.bz|tar xv" ou "tar --use-compress-prog=bunzip2 -xvf file.tar.bz2" pour utiliser bzip2 et bunzip2 avec tar. Ce patch fournit l'option -j vous permettant de décompresser une archive Bzip2 avec "tar xvfj file.tar.bz2". Appliquer ce patch sera mentionné plus tard quand le paquetage Tar sera réinstallé.

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.

La raison pour laquelle nous n'utilisons pas **`make install`** est que le make install de bzip2 n'installe pas la libbz2.so partagée, ni les binaires de bzip2 qui sont liés à cette bibliothèque. Donc nous n'avons pas d'autres choix que d'installer les fichiers manuellement.

Contenu de bzip2–1.0.1

Programmes

bunzip2 (lien vers bzip2), bzcat (lien vers bzip2), bzip2 et bzip2recover

Descriptions

bunzip2

Bunzip2 décompresse les fichiers compressés avec bzip2.

bzcat

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

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.

Bibliothèques

libbz2.[a,so]

libbz2

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

Dépendances

Bzip2−1.0.1 nécessite que les programmes suivants soient installés:

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:           1 minute
Estimation de l'espace disque requis:  2 Mo
```

Installation de Ed

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 patch basé sur ed. Ceci arrive rarement car les patchs basé sur diff sont préférés de nos jours.

Installez Ed en lançant les commandes suivantes:

```
cp buf.c buf.c.backup &&
sed 's/int u/int u, sfd/' buf.c.backup | \
   sed '/.*\*mktemp.*/d' | \
   sed 's/.*if (mktemp.*/  sfd = mkstemp(sfn);\
   if ((sfd == -1) || (sfp = fopen (sfn, "w+")) == NULL)/' > buf.c &&
./configure --prefix=/usr &&
make &&
make install &&
mv /usr/bin/ed /usr/bin/red /bin
```

Explication des commandes

Les commandes sed réparent une vulnérabilité dans ed. 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 des fichiers qu'il ne devrait pas, de changer des permissions de divers fichiers, etc.

Contenu de ed–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

Ed–0.2 nécessite que les programmes suivants soient installés:

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 Gettext–0.11.1

```
Estimation du temps de construction:           1 minute
Estimation de l'espace disque requis:  11MB
```

Installation de Gettext

Installez Gettext en exécutant les commandes suivantes:

```
./configure --prefix=/usr &&
make &&
make install
```

Contenu de gettext–0.10.40 ;

Programmes

gettext, gettextize, msgcmp, msgcomm, msgfmt, msgmerge, msgunfmt, ngettext et xgettext

Descriptions

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.

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.

msgfmt

Le programme msgftm compile les traductions en code machine. Ceci est utile pour créer le fichier final de traduction du programme/package.

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 versions compilées sont disponibles.

ngettext

Le programme ngettext affiche la traduction dans le langage natif d'un message textuel dont la forme grammaticale dépend d'un nombre.

xgettext

Le programme xgettex extrait les lignes du message à partir des sources C. Cela permet la création du fichier modèle de traduction.

Dépendances

Gettext−0.10.40 nécessite que les programmes suivants soient installés :

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 Kbd–1.06

```
Estimation du temps de construction:           1 minute
Estimation de l'espace disque requis:  8 Mo
```

Installation de Kbd

Installer Kbd en lançant les commandes suivantes:

```
patch -Np1 -i ../kbd-1.06-2.patch &&
./configure &&
make &&
make install
```

Explications des commandes

`patch -Np1 -i ../kbd-1.06-2.patch`: Ce patch 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 patch change un Makefile pour que certains utilitaires (setlogcons, setvesablank and getunimap) qui ne sont pas installés par défaut, soient quand même installé.

Contenu de kbd–1.06

Programmes

chvt, deallocvt, dumpkeys, fgconsole, getkeycodes, getunimap, kbd_mode, kbdrate, loadkeys, loadunimap, mapscrn, openvt, psfaddtable (link to psfxtable), psfgettable (link to psfxtable), psfstriptable (link to 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 c'est en utilisant ce programme que vous pourrez y arriver.

setlogcons

setlogcons envoie les messages du noyau vers la 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

Kbd−1.06 nécessite que les programmes suivants soient installés:

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

```
Estimation du temps de construction :           1 minute
Estimation de l'espace disque requis :  2 Mo
```

Installation de Diffutils

Installez Diffutils en lançant les commandes suivantes :

```
./configure --prefix=/usr &&
make &&
make install
```

Contenu de diffutils–2.7

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

Diffutils–2.7 nécessite que les programmes suivants soient installés:

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 :           2 minutes
Estimation de l'espace disque requis :  21 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 à 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–1.25

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 est utilisé pour vérifier un système de fichiers Linux ext2. fsck.ext2 fait la même chose que e2fsck.

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 de 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

Aucune description actuellement disponible.

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.

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 parmi les UUID créés 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

Aucune description actuellement disponible.

libe2p

Aucune description actuellement disponible.

libext2fs

Aucune description actuellement disponible.

libss

Aucune description actuellement disponible.

libuuid

Aucune description actuellement disponible.

Dépendances

E2fsprogs−1.25 nécessite que les programmes suivants soient installés :

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 Fileutils–4.1

```
Estimation du temps de construction:           3 minutes
Estimation de l'espace disque requis:  16 Mo
```

Installation de Fileutils

Installez Fileutils en lançant les commandes suivantes:

```
./configure --prefix=/usr --bindir=/bin &&
make &&
make install &&
cd /usr/bin &&
ln -sf ../../bin/install
```

Contenu de fileutils–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 argument et chacun de ces 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

Fileutils−4.1 nécessite que les programmes suivants soient installés:

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 Grep–2.5

```
Estimation du temps de construction:           1 minute
Estimation de l'espace disque requis:  3 Mo
```

Installation de Grep

Installez Grep en lançant les commandes suivantes:

```
./configure --prefix=/usr --bindir=/bin &&
make &&
rm /bin/egrep /bin/fgrep &&
make install
```

Contenu de grep–2.4.2

Programmes

egrep, fgrep 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

Grep–2.4.2 nécessite que les programmes suivants soient installés :

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:           1 minute
Estimation de l'espace disque requis:  1 Mo
```

Installation de Gzip

Installez Gzip en lançant les commandes suivantes:

```
./configure --prefix=/usr &&
cp gzexe.in gzexe.in.backup &&
sed 's%"BINDIR"%/bin%' gzexe.in.backup > gzexe.in &&
make &&
make install &&
cd /usr/bin &&
mv gzip /bin &&
rm gunzip zcat &&
cd /bin &&
ln -sf gzip gunzip &&
ln -sf gzip zcat &&
ln -sf gunzip uncompress
```

Contenu de gzip–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 le codage Lempel–Ziv (LZ77).

zcat

zcat décompresse une liste de fichiers données en ligne de commande ou utilise l'entrée standard en lecture et écrit les données décompressées sur la sortie standard.

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

Gzip−1.2.4a nécessite que les programmes suivants soient installés:

bash: shbinutils: as, ld, nmfileutils: chmod, cp, install, ln, mv, rmgcc: cc1, collect2, cpp, cpp0, gccgrep: egrep, grepmake: r

Installer Lilo−22.2

```
Estimation du temps de construction:           1 minute
Estimation de l'espace disque requis:  3 Mo
```

Installation de Lilo

Nous avons choisi Lilo car nous nous sentons à l'aise avec, mais vous désirez peut−être regarder ailleurs. Quelqu'un a écrit une astuce sur GRUB, un chargeur alternatif, dans http://hints.linuxfromscratch.org/hints/grub−howto.txt.

Installez Lilo en exécutant 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.

Peut−être serez vous intéressé de savoir que quelqu'un a écrit une astuce sur la manière d'avoir un logo à la place du prompt ou du menu standard de Lilo. Jetez un oeil dessus à http://hints.linuxfromscratch.org/hints/bootlogo.txt .

Contenu de lilo−22.1

Programmes

lilo et mkrescue

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.

Dépendances

Lilo−22.1 nécessite que les programmes suivants soient installés:

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:           1 minute
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–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

Make–3.79.1 nécessite que les programmes suivants soient installés :

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–2.4.15

```
Estimation du temps de construction:           1 minute
Estimation de l'espace disque requis:  2 Mo
```

Installation de Modutils

Installez Modutils en lançant les commandes suivantes:

```
./configure &&
make &&
make install
```

Contenu de modutils–2.4.12

Programmes

depmod, genksyms, insmod, insmod_ksymoops_clean, kallsyms (lien vers insmod), kernelversion, ksyms, lsmod (lien vers insmod), modinfo, modprobe (lien vers insmod) et rmmod

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 n'ayant pas été accédés depuis 2 jours.

kallsyms

kallsyms extrait tous les symboles du kernel pour le débuggage.

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 à partir de l'ensemble des modules disponibles situés dans les arborescences prévues à cet effet.

rmmod

rmmod décharge les modules chargeables du noyau en cours d'utilisation.

Dépendances

Modutils−2.4.12 nécessite que les programmes suivants soient installés:

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:           1 minute
Estimation de l'espace disque requis:  1 Mo
```

Installation de Netkit−base

Installez Netkit−base en lançant les commandes suivantes:

```
./configure &&
make &&
make install &&
cd etc.sample &&
cp services protocols /etc
```

Il y a d'autres fichiers dans le répertoire `etc.sample` qui pourraient vous intéresser.

Contenu de netkit−base−0.17

Programmes

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

Netkit−base−0.17 nécessite que les programmes suivants soient installés:

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:           1 minute
Estimation de l'espace disque requis:  2 Mo
```

Installation de Patch

Installer Patch en lançant les commandes suivantes:

```
export CPPFLAGS=-D_GNU_SOURCE &&
./configure --prefix=/usr &&
unset CPPFLAGS &&
make &&
make install
```

Contenu de patch–2.5.4

Programmes

patch

Descriptions

patch

Le programme patch modifie un fichier en fonction d'un fichier de patch. Un fichier de patch 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 1Mo. 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'1Mo ou juste sous la forme d'un fichier de patch d'1Ko qui viendra mettre à jour la précédente version pour la rendre identique à la dernière version réalisée. Ainsi, si la première version avait déjà été téléchargée, un fichier de patch évite un deuxième téléchargement conséquent.

Dépendances

Patch–2.5.4 nécessite que les programmes suivants soient installés:

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:           1 minute
Estimation de l'espace disque requis:  170 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−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 quel matériel utilise quelles adresses d'entrées−sorties et quels canaux IRQ et DMA.

procinfo

procinfo récupère des informations à partir du répertoire /proc et les affiche proprement formatées 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

Procinfo−18 nécessite que les programmes suivants soient installés:

binutils: as, ld fileutils: install, mkdir gcc: cc1, collect2, cpp0, gcc make: make

Installer Procps–2.0.7

```
Estimation du temps de construction:           1 minute
Estimation de l'espace disque requis:  2 Mo
```

Installation de Procps

Installez Procps en lançant les commandes suivantes:

```
make &&
make XSCPT='' install &&
mv /usr/bin/kill /bin
```

Explication des Commandes

`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–2.0.7

Programmes

free, kill, oldps, pgrep, pkill, ps, skill, snice, sysctl, tload, top, uptime, 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 and 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.

uptime

uptime donne en une ligne les informations suivantes : le temps courant, depuis combien de temps le système tourne−t−il, combien d'utilisateurs sont actuellement connectés, et la charge moyenne du système dans les 1, 5, et 15 dernières minutes.

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 liés pour sauver de l'espace disque, en implémentant des fonctions communes une seule fois.

Dépendances

Procps−2.0.7 nécessite que les programmes suivants soient installés :

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–20.2

```
Estimation du temps de construction:           1 minute
Estimation de l'espace disque requis:  500 Ko
```

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:

```
cd /bin
ln -s killall 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 boot, 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–20.2

Programmes

fuser, killall, pidof (lien vers 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 utilisant les fichiers ou systèmes de fichiers qui sont spécifiés.

killall

killall envoie un signal à tous les processus faisant tourner l'une des commandes spécifiées.

pidof

Pidof trouve les identifiants de processus (pids) du programme nommé et imprime ces pids sur la sortie standard.

pstree

pstree présente les processus en cours d'exécution sous forme d'arborescence.

Dépendances

Psmisc−20.2 nécessite que les programmes suivants soient installés :

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 Reiserfsprogs–3.x.1b

```
Estimation du temps de construction:           1 minute
Estimation de l'espace disque requis:  7 Mo
```

Installation de Reiserfsprogs

Reiserfsprogs ne doit être installé seulement si vous comptez utiliser le système de fichiers reiserfs. Installez Reiserfsprogs en lançant les commandes suivantes:

```
./configure --mandir=/usr/share/man &&
make &&
make install
```

Explication des Commandes

`--mandir=/usr/share/man:` Ceci assure que les pages de man sont installées dans le répertoire correct, tandis que les programmes seront toujours installés dans `/sbin` comme ils devraient l'être.

Contenu de reiserfsprogs–3.x.0j

Programmes

debugreiserfs, mkreiserfs, reiserfsck, resize_reiserfs et unpack

Descriptions

debugreiserfs

debugreiserfs peut parfois aider à résoudre des problèmes des systèmes de fichiers reiserfs. S'il est appelé sans options il imprime le superblock de tout système de fichier reiserfs trouvé sur le périphérique.

mkreiserfs

mkreiserfs crée un système de fichiers reiserfs.

reiserfsck

reiserfsck vérifie un système de fichiers reiserfs.

resize_reiserfs

resize_reiserfs est utilisé pour redimensioner un système de fichiers reiserfs démonté.

unpack

Aucune description actuellement disponible.

Dépendances

Reiserfs−3.x.0j nécessite que les programmes suivants soient installés :

autoconf: autoconf, autoheader automake: aclocal, automake bash: sh binutils: ar, as, ld, ranlib diffutils: cmp fileutils: chmod, install, ls, rm 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 Sed–3.02

```
Estimation du temps de construction:           1 minute
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–3.02

Programmes

sed

Descriptions

sed

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

Dépendances

Sed–3.02 nécessite que les programmes suivants soient installés :

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:           2 minutes
Estimation de l'espace disque requis:  11 Mo
```

Installation de Sh–utils

Installez Shellutils en lançant les commandes suivantes:

```
./configure --prefix=/usr &&
make &&
make install &&
cd /usr/bin &&
mv basename date echo false hostname /bin &&
mv pwd sleep stty su test true uname /bin &&
mv chroot ../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'interpreteur bash, mais le FHS définit qu'ils devraient être un binaire `[`. Nous le créons de cette manière, pendant qu'on est encore dans le répertoire `/usr/bin` :

```
cd /bin &&
ln -sf test [
```

Contenu de sh–utils–2.0

Programmes

basename, chroot, date, dirname, echo, env, expr, factor, false, groups, hostid, hostname, 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 l'heure et la date courantes 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 le groupe auquel appartient l'utilisateur.

hostid

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

hostname

hostname met à jour ou affiche le nom courant de l'hôte.

id

id affiche les UIDs et GIDs, réels et effectifs, d'un utilisateur ou de l'utilisateur courant.

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 standard qui ne sera pas de type tty.

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 dans 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 l'identifiant effectif de l'utilisateur.

yes

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

Dépendances

Sh−utils−2.0 nécessite que les programmes suivants soient installés :

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 Net−tools−1.60

```
Estimation du temps de construction:           1 minute
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 majoité des cas. Il va vous être posé toute une série de questions sur le type de protocoles réseaux que vous avez activé dans votre noyau.

Les réponses par défaut vont permettre aux outils du package de fonctionner 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 capables 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−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 est utilisé pour manipuler 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 définit le nom du système.

hostname

hostname est utilisé pour définir ou afficher le nom d'hôte du système.

ifconfig

ifconfig est la commande principale pour configurer les interfaces réseau.

nameif

nameif nomme les interfaces réseaux, basées sur les adresses MAC.

netstat

netstat est un utilitaire ayant de nombreux usages; il affiche les connections réseaux, 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 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

Net−tools−1.60 nécessite que les programmes suivants soient installés:

bash: bash, sh binutils: ar, as, ld fileutils: install, ln, ls, mv, rm gcc: cc, cc1, collect2, cpp0
make: make sh−utils: echo

Installer Shadow–4.0.3

```
Estimation du temps de construction:           3 minutes
Estimation de l'espace disque requis:  6 Mo
```

Installation de Shadow Password Suite

Avant d'installer ce package, vous devriez jeter un oeil à l'astuce lfs http://hints.linuxfromscratch.org/hints/shadowpasswd_plus.txt. Celle–ci examine la manière de rendre votre système plus sûr en ce qui concerne les mots de passe et comment tirer le meilleur parti de ce package.

Installez la Shadow Password Suite en lançant les commandes suivantes:

```
./configure --prefix=/usr --enable-shared &&
make &&
make install &&
cd etc &&
cp limits login.access /etc &&
sed 's%/var/spool/mail%/var/mail%' login.defs.linux > /etc/login.defs &&
cd /usr/sbin &&
ln -sf vipw vigr &&
rm /bin/vipw &&
mv /bin/sg /usr/bin &&
cd /lib &&
mv libmisc.*a libshadow.*a /usr/lib &&
cd /usr/lib &&
ln -s ../../lib/libshadow.so
```

Explication des commandes

`cp limits login.access /etc:` Ces fichiers n'ont pas été installés 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 -sf vipw vigr`: D'après la page man de vipw, vigr devrait être un lien. Comme la procédure d'installation de shadow ne crée pas les liens symboliques, nous les créons manuellement.

Contenu de shadow–20001016

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, les coordonnées professionnelles, le numéro de téléphone au bureau, et le numéro de téléphone personnel pour un compte utilisateur.

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

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.

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 connexions, /var/log/lastlog. Le nom de login, le port, et l'heure de dernière connexion seront affichés.

login

login est utilisé pour établir une nouvelle session sur le système.

logoutd

logoutd applique les restrictions de temps de connexion et de port de connexion définies 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 informationspour mettre à jour un groupe d'utilisateurs existants 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 informations du système d'authentification.

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 exécute une commande avec un identifiant de groupe différent.

su

Change l'identifiant d'utilisateur et de groupe. Il remplace le programme su du package Shellutils.

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 relatives au nom de login spécifié.

usermod

usermod modifie les fichiers du système de gestion de comptes en focntion 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

libshadow.[a,so]

Descriptions

libshadow

libshadow apporte des fonctions communes aux programmes shadow.

Dépendances

Shadow−20001016 nécessite que les programmes suivants soient installés :

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:           1 minute
Estimation de l'espace disque requis:  710 Ko
```

Installation de Sysklogd

Installez Sysklogd en lançant les commandes suivantes:

```
make &&
make install
```

Contenu de sysklogd–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 utilisent. 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

Sysklogd–1.4.1 nécessite que les programmes suivants soient installés:

binutils: as, ld, strip fileutils: install gcc: cc1, collect2, cpp0, gcc make: make

Installer Sysvinit–2.84

```
Estimation du temps de construction:           1 minute
Estimation de l'espace disque requis:  630 Ko
```

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 src/init.c.backup &&
sed 's/\(.*\)\(Sending processes\)\(.*\)/\1\2 started by init\3/' \
   src/init.c.backup > src/init.c
```

Installez Sysvinit en lançant les commandes suivantes:

```
make –C src &&
make –C src install
```

Contenu de sysvinit–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 demande ensuite au noyau de s'arrêter, de redémarrer ou de mettre le système hors tension. Si halt ou reboot sont appelés alors que le système ne se trouve pas dans un niveau d'exécution 0 ou 6, la commande shutdown sera exécutée à la place (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 sur lesquelles les utilisateurs peuvent se connecter. 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) d'après le nom de programme et affiche ces identifiants sur la sortie standard.

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 du niveau d'exécution, puis affiche le niveau d'exécution 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 utilisateur unique (ceci est déterminé dans le fichier /etc/inittab). Init essaie aussi d'exécuter sulogin lorsqu'on le chargeur de boot l'appelle avec l'option −b (e.g., LILO).

telinit

telinit envoie les signaux appropriés à init pour lui spécifier dans quel niveau d'exécution 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

Sysvinit−2.84 nécessite que les programmes suivants soient installés:

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:           1 minute
Estimation de l'espace disque requis:  7 Mo
```

Installation de Tar

Si vous voulez pouvoir utiliser directement des fichiers bzip2 avec tar, vous pouvez utiliser le patch tar disponible depuis le site FTP de LFS. Ce patch ajoutera l'option −j à tar, qui fonctionne comme l'option −z (qui peut être utilisée pour les fichiers gzip).

Appliquez le patch en lançant la commande suivante:

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

Installez Tar en lançant les commandes suivantes depuis le répertoire de plus haut niveau:

```
./configure --prefix=/usr --libexecdir=/usr/bin \
   --bindir=/bin &&
make &&
make install
```

Contenu de tar−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

Tar−1.13 nécessite que les programmes suivants soient installés :

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 Textutils–2.0

```
Estimation du temps de construction:           1 minute
Estimation de l'espace disque requis:  15 Mo
```

Installation de Textutils

Installez Textutils en lançant les commandes suivantes:

```
./configure --prefix=/usr &&
make &&
make install &&
mv /usr/bin/cat /usr/bin/head /bin
```

Contenu de textutils–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 l'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 extrait d'un fichier des fragments séparés par un ou plusieurs motifs, dans des fichiers xx01, xx02, ..., et affiche sur la sortie standard le compte d'octets de chaque segment.

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, et 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

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 chaque fichier spécifié, et une ligne de totaux si plusieurs fichiers ont été spécifiés.

Dépendances

Textutils−2.0 nécessite que les programmes suivants soient installés :

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–2.11o

```
Estimation du temps de construction:           1 minute
Estimation de l'espace disque requis:  9 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 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–2.11n

Programmes

agetty, arch, blockdev, cal, cfdisk, chkdupexe, col, colcrt, colrm, column, ctrlaltdel, cytune, ddate, dmesg, elvtune, fdformat, fdisk, fsck.minix, getopt, hexdump, hwclock, ipcrm, ipcs, isosize kill, line, logger, look, losetup, mcookie, mkfs, mkfs.bfs, mkfs.minix, mkswap, more, mount, namei, 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, tunelp, ul, umount, vidmode, 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.minix

fsck.minix exécute une vérification logique d'un système de fichiers Linux MINIX.

getopt

getops gère les options de commande de la même façon que la fonction C getopt.

hexdump

hexdump affiche le ou les fichiers spécifiés, 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

kill

kill envoie un signal spécifié au processus spécifié.

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.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.

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 pour la pagination et le swap.

swapon

swapon active les périphériques et les fichiers pour la pagination et le 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

Util−linux−2.11n nécessite que les programmes suivants soient installés:

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−1.9

```
Estimation du temps de construction :           1 minute
Estimation de l'espace disque requis :  23 Ko
```

Installation de LFS−Bootscripts

Nous allons utiliser des scripts de démarrage de style SysV. Nous avons choisi ce style parce que c'est utilisé couramment et que nous les connaissons bien. Si vous souhaitez essayer autre chose, quelqu'un a écrit une astuce LFS pour des scripts de démarrage de style BSD sur http://hints.linuxfromscratch.org/hints/bsd−init.txt.

Si vous décidez d'utiliser le style BSD, ou tout autre style de scripts, vous pouvez 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−1.9

Scripts

checkfs, cleanfs, functions, halt, loadkeys, localnet, mountfs, 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.

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.

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

bootscripts−1.9 nécessite que les programmes suivants soient installés:

fileutils: chown, cp

Désinstaller les anciennes librairies NSS

Si vous avez copié les fichiers des librairies NSS depuis le système Linux normal vers le système LFS (car le système normal utilise Glibc−2.0), il est temps de les désinstaller en exécutant:

```
rm /lib/libnss*.so.1 /lib/libnss*2.0*
```

Configurer les logiciels essentiels

Maintenant que tous les logiciels sont installés, tout ce dont nous avons besoin pour que les programmes fonctionnent 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 inclu 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 exécutant:

```
cd /etc &&
ln -sf ../usr/share/zoneinfo/<tzselect's output> 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 localtime
```

Ou:

```
ln -sf ../usr/share/zoneinfo/Canada/Eastern localtime
```

Configurer le chargeur dynamique

Par défaut, le chargeur dynamique (`/lib/ld-linux.so.2`) cherche à travers `/lib` et `/usr/lib` les 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 éxécutant 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 des utilisateurs/groupes, les supprimer et plus encore. Nous n'allons pas expliquer ce que 'password shadowing' veut dire. Tout là dessus 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:respawn:/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:

```
cd /usr/share/kbd/keymaps &&
ln -s <path/to/keymap> 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 defkeymap.map.gz
```

Une deuxième option pour configurer le 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.

Si vous n'aviez pas créer le fichier defkeymap.map.gz et que vous souhaitez utiliser le plan de codage par défaut (US), encore une fois, vous n'avez rien besoin de faire. Le noyau est compilé avec un plan de codage par défaut qui sera bon pour vous, donc ne lancez pas la prochaine commande.

Lancez la commande suivante:

```
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.

Le 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 connection 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 pour 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 trois chiffres après la lettre initiale. Le K signifie d'arrêter (kill) un service, et le S (start) d'en démarrer un. Les chiffres déterminent l'ordre d'exécution des scripts, de 000 à 999 ; 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. Quant 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, pas du nôtre). Les fichiers 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 tels que les timezones, UTC, et la variable d'environnement TZ.

Configuration du script loadkeys

Vous ne devez utiliser ce script que si vous ne possédez pas un clavier américain à 101 touches.

Le fichier /etc/sysconfig/keyboard contient une information dont le script loadkeys a besoin pour fonctionner. Ce fichier contient la variable LAYOUT indiquant à loadkeys quel fichier de description du clavier doit être chargé.

Créez un nouveau fichier `/etc/sysconfig/keyboard` en lançant la commande suivante:

```
cat > /etc/sysconfig/keyboard << "EOF"
# Début de /etc/sysconfig/keyboard

LAYOUT=<chemin-de-description-clavier>

# Fin de /etc/sysconfig/keyboard
EOF
```

Remplacez **<chemin-de-description-clavier>** par le chemin de description clavier que vous avez sélectionné. Par exemple, si vous avez choisi un clavier américain, vous devriez mettre **/usr/share/kbd/keymaps/i386/qwerty/us.map.gz**

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 sur les fichiers de log toutes les vingt minutes par défaut. Si vous préférez l'activer, éditer 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.

Créer le fichier /etc/hosts

Si une carte réseau doit être configurée, vous devez choisir l'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 aliases
```

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.

Configuration de la passerelle par défaut

Si vous êtes sur un réseau, vous aurez 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 (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 changées 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 donc non 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–2.4.18

```
Temps de construction estimé:           Depends on options selected
Espace disque nécessaire estimé:  Depends on options selected
```

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 à quoi servent les autres options.

Une chose que vous pourriez faire, est de prendre le fichier `.config` des sources du noyau de votre distribution hôte et le copier dans `$LFS/usr/src/linux`. De cette façon vous n'avez pas à configurer la totalité du noyau depuis le début et pouvez utiliser vos valeurs actuelles. Si vous choisissez de le faire, commencez par lancer la commande **`make mrproper`**, ensuite copiez le fichier `.config` par dessus, ensuite 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`**).

Les commandes suivantes sont utilisées pour construire le noyau:

```
cd /usr/src/linux &&
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

Linux–2.4.17 nécessite que les programmes suivants soient installés :

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 pour le moment).

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 inutilisable 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).

Tout d'abord nous sortirons du chroot et copierons le fichier lfskernel sur le système de base:

```
logout
cp $LFS/boot/lfskernel /boot
```

L'étape suivante est d'ajouter une 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.

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 de lilo 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 votre système LFS de tous les symboles de débogage. 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 megas en enlevant les symboles de débogage. Ce processus n'a pas d'autres inconvénients que de vous empêcher de deboguer 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 problèmes. 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 instable (essentiellement en détruisant les modules du noyau et les librairies 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 risques dans des circonstances normales. Cela ne débarasse pas les fichiers de quoique ce soit de vital. Il est également sans dangers d'utiliser −−strip−all sur des programmes classiques (ne pas utiliser sur des librairies − elles seraient détruites) mais cette méthode n'est pas aussi sûre et l'espace gagné est à peine 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 de man pour d'autres options de strip que vous pourriez utiliser. L'idée générale est de ne pas lancer strip sur des librairies (autre que −−strip−debug) pour ne pas prendre de risques.

```
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−3.3. 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 installé sur votre système. Cela peut être un fichier vide en lançant :

```
touch $LFS/etc/lfs-3.3
```

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 ayez 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
```

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
```

Si vous vous demandez: "Bien, où aller maintenant?" vous serez heureux d'apprendre que quelqu'un à écrit un topo LFS sur le sujet à http://hints.linuxfromscratch.org/hints/afterlfs.txt. Sur la même note, si vous n'êtes pas seulement débutant avec LFS, mais aussi débutant Linux en général, vous pouvez trouver un topo très intéressant pour débutant à: http://hints.linuxfromscratch.org/hints/newbie.txt.

N'oubliez pas qu'il y a plusieurs listes de diffusions LFS auxquelles vous pouvez vous inscrire en cas de besoin d'aide, conseils, etc. Pour plus d'informations, se référer à Chapitre 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

Cet annexe décrit les aspects suivants des packages installés dans ce livre:

- 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'allons pas ré écrire la totalité de la page man, mais simplement les éléments principaux pour permettre la compréhension de l'utilité du package. Pour disposer de tous les détails sur un programme,nous vous suggérons de commencer par lire la page man complète en plus de cette annexe.

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 envoyer un email à la liste de diffusion. 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 courante du package. Vérifier les dépendances demande un travail important, donc il peut y avoir 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. Si nous mettons à jour vers une nouvelle version majeure, nous devons nous assurer que les dépendances ont aussi été vérifiées en même temps.

Autoconf

Site officiel de téléchargement

Autoconf (2.53): ftp://ftp.gnu.org/gnu/autoconf/

Contenu de autoconf–2.52

Programmes

autoconf, autoheader, 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.

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 néssaire) 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 dont la racine est passée en argument de ligne de commande ou le répertoire courant si aucun n'est fourni. 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 les conditions 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

Autoconf−2.52 nécessite que les programmes suivants soient installés:

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): ftp://ftp.gnu.org/gnu/automake/

Contenu de automake−1.5

Programmes

aclocal et automake

Descriptions

aclocal

Automake inclut un certain nombre de macros Autoconf qui peuvent être utilisées dans les packages; certaines d'entre elles sont actuellement requises 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

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.

Dépendances

Automake−1.5 nécessite que les programmes suivants soient installés:

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−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

Bash−2.05a nécessite que les programmes suivants soient installés:

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 (0.16.2): http://www.cix.co.uk/~mayday/

Contenu de bin86–0.16.0

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 comprend que les fichiers objets produits par l'assembleur as86. Il peut les lier dans un exécutable I&D soit impur, soit séparé.

nm86

Aucune description actuellement disponible.

objdump86

Aucune description actuellement disponible.

size86

Aucune description actuellement disponible.

Dépendances

Bin86–0.16.0 nécessite que les programmes suivants soient installés:

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 (2.12): ftp://ftp.gnu.org/gnu/binutils/

Contenu de binutils–2.11.2

Programmes

addr2line, ar, as, c++filt, 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.

c++filt

Le langage C++ fournit un mécanisme de surcharge, ce qui signifie qu'il est possible d'écrire plusieurs fonctions ayant le même nom (pourvu que chacune demande des paramètres de types différents). Tous les noms de fonction C++ sont encodés dans un nom d'assemblage de bas niveau (ce processus est connu sous le nom de broyage). Le programme c++filt fait l'association inverse : il décode (dé–broie) les noms de bas niveau en noms de niveau utilisateur, pour que l'éditeur de liens puisse prévenir un conflit entre ces fonctions surchargées.

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 objfile 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.

Fichiers de Librairies

libbfd.a, libiberty.a et libopcodes.a

Descriptions

libbfd

libbfd est une librairie de description des fichiers binaires.

libiberty

libiberty est un ensemble de sous−routines utilisé dans de divers programmes GNU dont getopt, obstack, strerror, strtol et strtoul.

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

Binutils−2.11.2 nécessite que les programmes suivants soient installés:

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 (1.34): ftp://ftp.gnu.org/gnu/bison/

Contenu de bison−1.31

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 yacc qui appelle bison avec l'option −y. Ceci est fait pour des raisons de compatibilté avec des programmes qui utilisent yacc au lieu de bison.

Dépendances

Bison−1.31 nécessite que les programmes suivants soient installés:

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 bzip2−1.0.1

Programmes

bunzip2 (lien vers bzip2), bzcat (lien vers bzip2), bzip2 et bzip2recover

Descriptions

bunzip2

Bunzip2 décompresse les fichiers compressés avec bzip2.

bzcat

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

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.

Bibliothèques

libbz2.[a,so]

libbz2

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

Dépendances

Bzip2−1.0.1 nécessite que les programmes suivants soient installés:

bash: sh binutils: ar, as, ld, ranlib fileutils: cp, ln, rm gcc: cc1, collect2, cpp0, gcc make: make

Diffutils

Site officiel de

Diff Utils (2.8): ftp://ftp.gnu.org/gnu/diffutils/

Contenu de diffutils–2.7

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

Diffutils–2.7 nécessite que les programmes suivants soient installés:

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−1.25

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 est utilisé pour vérifier un système de fichiers Linux ext2. fsck.ext2 fait la même chose que e2fsck.

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 de 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

Aucune description actuellement disponible.

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.

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 parmi les UUID créés 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

Aucune description actuellement disponible.

libe2p

Aucune description actuellement disponible.

libext2fs

Aucune description actuellement disponible.

libss

Aucune description actuellement disponible.

libuuid

Aucune description actuellement disponible.

Dépendances

E2fsprogs−1.25 nécessite que les programmes suivants soient installés :

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/

Contenu de ed−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

Ed−0.2 nécessite que les programmes suivants soient installés:

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 (3.37): ftp://ftp.gw.com/mirrors/pub/unix/file/

Contenu de file–3.37

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

File–3.37 nécessite que les programmes suivants soient installés :

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

File Utils (4.1): ftp://ftp.gnu.org/gnu/fileutils/

Contenu de fileutils–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 argument et chacun de ces 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

Fileutils−4.1 nécessite que les programmes suivants soient installés:

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

Find Utils (4.1): ftp://ftp.gnu.org/gnu/findutils/ Patch Find Utils (4.1): ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/ http://ftp.linuxfromscratch.org/lfs−packages/3.3/

Contenu du package findutils−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 définis, il liste tous les fichiers du répertoire courant et de ses sous−répertoires.

frcode

updatedb exécute un programme nommé frcode 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, on peut alors créer un fichier pour contenir la liste de tous ces fichiers (un par ligne) et utiliser xargs pour exécuter cette commande sur chacun d'eux.

Dépendances

Findutils−4.1 nécessite que les programmes suivants soient installés:

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−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 utilisée exclusivement par les scanners C++.

lex

Nous créons un script yacc qui appelle 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

Aucune description actuellement disponible.

Dépendances

Flex−2.5.4a nécessite que les programmes suivants soient installés:

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 (3.1.0) : ftp://ftp.gnu.org/pub/gnu/gawk/

Contenu de gawk–3.1.0

Pas encore vérifié.

Dépendances

Gawk–3.1.0 nécessite que les programmes suivants soient installés :

Aucune dépendance encore vérifiée.

GCC

Site officiel de téléchargement

GCC (2.95.3): ftp://ftp.gnu.org/pub/gnu/gcc/ GCC Patch (2.95.3−2): ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/ http://ftp.linuxfromscratch.org/lfs−packages/3.3/

Contenu de gcc−2.95.3

Programmes

c++, c++filt, cc (lien vers gcc), cc1, cc1plus, collect2, cpp, cpp0, g++, gcc, gcov, protoize et unprotoize

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

c++filt est utilisé pour filtrer les symboles C++.

collect2

Aucune description actuellement disponible.

cpp, cpp0

cpp réalise un travail préliminaire sur un fichier source, comme inclure le contenu des fichiers d'entêtes dans ce code source. C'est une bonne idée de ne pas faire ceci manuellement pour gagner beaucoup de temps. Il suffit d'insérer une ligne comme #include <filename>. Ce préprocesseur insère le contenu de ce fichier dans le fichier source. C'est une des choses que réalise le préprocesseur.

gcov

Aucune description actuellement disponible.

protoize

Programme optionnel qui transforme des fonctions ou des définitions pré−ANSI dans le nouveau style des prototypes C ANSI (le fichier par défaut à rechercher est

`/usr/lib/gcc-lib/<arch>/<version>/SYSCALLS.c.X`)

unprotoize

Programme optionnel qui transforme des prototypes réalisés avec protoize vers l'ancien style pré−ANSI (job correct seulement quand conversion avant protoize)

Bibliothèques

libgcc.a, libiberty.a, libstdc++.[a,so]

libgcc

libgcc.a est une bibliothèque utilisée par gcc lors de son exécution. La plupart du temps, sur la plupart des machines, libgcc.a n'est pas réellement nécessaire.

libiberty

libiberty est une collection de sous−routines utilisées par différents programmmes 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 utilisé 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.

Dépendances

GCC−2.95.3 nécessite que les programmes suivants soient installés:

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 (0.11.1): ftp://ftp.gnu.org/gnu/gettext/

Contenu de gettext–0.10.40 ;

Programmes

gettext, gettextize, msgcmp, msgcomm, msgfmt, msgmerge, msgunfmt, ngettext et xgettext

Descriptions

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.

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.

msgfmt

Le programme msgftm compile les traductions en code machine. Ceci est utile pour créer le fichier final de traduction du programme/package.

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 versions compilées sont disponibles.

ngettext

Le programme ngettext affiche la traduction dans le langage natif d'un message textuel dont la forme grammaticale dépend d'un nombre.

xgettext

Le programme xgettex extrait les lignes du message à partir des sources C. Cela permet la création du fichier modèle de traduction.

Dépendances

Gettext−0.10.40 nécessite que les programmes suivants soient installés :

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/

Contenu de glibc−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

Aucune description actuellement disponible.

gencat

gencat génère des catalogues de messages.

getconf

Aucune description actuellement disponible.

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 requises par chaque programme ou bibliothèque partagée spécifié sur la ligne de commande.

lddlibc4

Aucune description actuellement disponible.

locale

Aucune description actuellement disponible.

localedef

localedef compile les spécifications pour locale.

mtrace

Aucune description actuellement disponible.

nscd

nscd est un démon procurant un cache pour les requêtes DNS les plus courantes.

nscd_nischeck

Aucune description actuellement disponible.

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

Aucune description actuellement disponible.

rpcinfo

Aucune description actuellement disponible.

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 librairies partagées.

libBrokenLocale, libBrokenLocale_p

Aucune description actuellement disponible.

libSegFault

Aucune description actuellement disponible.

libanl, libanl_p

Aucune description actuellement disponible.

libbsd−compat

Aucune description actuellement disponible.

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 fourni 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

Aucune description actuellement disponible.

libg

Aucune description actuellement disponible.

libieee

Aucune description actuellement disponible.

libm, libm_p

libm est la bibliothèque de mathématiques.

libmcheck

Aucune description actuellement disponible.

libmemusage

Aucune description actuellement disponible.

libnsl, libnsl_p

Aucune description actuellement disponible.

libnss_compat, libnss_dns, libnss_files, libnss_hesiod, libnss_nis, libnss_nisplus

Aucune description actuellement disponible.

libpcprofile

Aucune description actuellement disponible.

libpthread, libpthread_p

Aucune description actuellement disponible.

libresolv, libresolv_p

Aucune description actuellement disponible.

librpcsvc, librpcsvc_p

Aucune description actuellement disponible.

librt, librt_p

Aucune description actuellement disponible.

libthread_db

Aucune description actuellement disponible.

libutil, libutil

Aucune description actuellement disponible.

Dépendances

Glibc−2.2.5 nécessite que les programmes suivants soient installés :

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−2.4.2

Programmes

egrep, fgrep 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

Grep−2.4.2 nécessite que les programmes suivants soient installés :

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 (1.17.2): ftp://ftp.gnu.org/gnu/groff/

Contenu de groff−1.17.2

Programmes

addftinfo, afmtodit, eqn, grn, grodvi, groff, grog, grolbp, grolj4, grops, grotty, hpftodit, indxbib, lkbib, lookbib, mmroff, neqn, nroff, pfbtops, pic, post−grohtml, pre−grohtml, refer, soelim, tbl, tfmtodit et troff

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.

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'.

hpftodit

hpftodit crée un fichier de font 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 en 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.

Dépendances

Groff−1.17.2 nécessite que les programmes suivants soient installés :

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/ Gzip Patch (1.2.4a): ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/ http://ftp.linuxfromscratch.org/lfs−packages/3.3/

Contenu de gzip−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 le codage Lempel−Ziv (LZ77).

zcat

zcat décompresse une liste de fichiers données en ligne de commande ou utilise l'entrée standard en lecture et écrit les données décompressées sur la sortie standard.

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

Gzip−1.2.4a nécessite que les programmes suivants soient installés:

bash: shbinutils: as, ld, nmfileutils: chmod, cp, install, ln, mv, rmgcc: cc1, collect2, cpp, cpp0, gccgrep: egrep, grepmake: r

Kbd

Site officiel de téléchargement

Kbd (1.06): ftp://ftp.win.tue.nl/pub/linux−local/utils/kbd/ Kbd Patch (1.06−2): ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/ http://ftp.linuxfromscratch.org/lfs−packages/3.3/

Contenu de kbd−1.06

Programmes

chvt, deallocvt, dumpkeys, fgconsole, getkeycodes, getunimap, kbd_mode, kbdrate, loadkeys, loadunimap, mapscrn, openvt, psfaddtable (link to psfxtable), psfgettable (link to psfxtable), psfstriptable (link to 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 c'est en utilisant ce programme que vous pourrez y arriver.

setlogcons

setlogcons envoie les messages du noyau vers la 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

Kbd−1.06 nécessite que les programmes suivants soient installés:

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

Linux kernel

Site officiel de téléchargement

Linux Kernel (2.4.18): ftp://ftp.kernel.org/pub/linux/kernel/

Contenu de kernel–2.4.17

Programmes

le noyau linux et les en–têtes du noyau linux

Descriptions

Noyau linux

Le noyau de 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 remplacés lors d'une mise à jour du kernel. Ils sont essentiels pour compiler un grand nombre de logiciels.

Dépendances

Linux–2.4.17 nécessite que les programmes suivants soient installés :

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

Less

Site officiel de téléchargement

Less (374): ftp://ftp.gnu.org/gnu/less/

Contenu de less−358

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 avec possibilité de défilement. 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

Less−358 nécessite que les programmes suivants soient installés:

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 (1.9): ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/
http://ftp.linuxfromscratch.org/lfs−packages/3.3/

Contenu de LFS−bootscripts−1.9

Scripts

checkfs, cleanfs, functions, halt, loadkeys, localnet, mountfs, 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.

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.

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

bootscripts−1.9 nécessite que les programmes suivants soient installés:

fileutils: chown, cp

Libtool

Site officiel de téléchargement

Libtool (1.4.2): ftp://ftp.gnu.org/gnu/libtool/

Contenu de libtool−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,so]

Descriptions

libltdl

Libtool fournit une petite bibliothèque, appelée 'libltdl', qui a pour but de masquer aux programmeurs les différentes difficultés liées à l'ouverture dynamique des bibliothèques.

Dépendances

Libtool−1.4.2 nécessite que les programmes suivants soient installés:

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−22.1

Programmes

lilo et mkrescue

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.

Dépendances

Lilo−22.1 nécessite que les programmes suivants soient installés:

bash: sh bin86: as86, ld86 binutils: as, ld, strip fileutils: cp, dd, ln gcc: cc, cc1, collect2, cpp0
make: make sed: sed textutils: cat

M4

Site officiel de téléchargement

M4 (1.4): ftp://ftp.gnu.org/gnu/m4/

Contenu de m4−1.4

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

M4−1.4 nécessite que les programmes suivants soient installés:

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−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

Make−3.79.1 nécessite que les programmes suivants soient installés :

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 (1.4): ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/
http://ftp.linuxfromscratch.org/lfs−packages/3.3/

Contenu de MAKEDEV−1.4

Programmes

MAKEDEV

Descriptions

MAKEDEV

MAKEDEV est un script qui peut aider à créer 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

MAKEDEV−1.4 nécessite que les programmes suivants soient installés:

bash: sh fileutils: chmod, chown, cp, ln, mknod, mv, rm grep: grep sh−utils: expr, id

Man

Site officiel de téléchargement

Man (1.5j): ftp://ftp.win.tue.nl/pub/linux−local/utils/man/

Contenu de man−1.5j

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

Man−1.5i2 nécessite que les programmes suivants soient installés :

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 (1.48): ftp://ftp.kernel.org/pub/linux/docs/manpages/

Contenu de manpages−1.47

Fichiers support

Plusieurs pages de manuel qui ne viennent pas avec les packages.

Descriptions

manual pages

Les exemples de pages de manuel fournies décrivent toutes les fonctions C et C++, quelques fichiers importants /dev/ et autres.

Dépendances

Man−pages−1.47 nécessite que les programmes suivants soient installés:

bash: sh fileutils: install make: make

Modutils

Site officiel de téléchargement

Modutils (2.4.15): ftp://ftp.kernel.org/pub/linux/utils/kernel/modutils/

Contenu de modutils–2.4.12

Programmes

depmod, genksyms, insmod, insmod_ksymoops_clean, kallsyms (lien vers insmod), kernelversion, ksyms, lsmod (lien vers insmod), modinfo, modprobe (lien vers insmod) et rmmod

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 n'ayant pas été accédés depuis 2 jours.

kallsyms

kallsyms extrait tous les symboles du kernel pour le débuggage.

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 à partir de l'ensemble des modules disponibles situés dans les arborescences prévues à cet effet.

rmmod

rmmod décharge les modules chargeables du noyau en cours d'utilisation.

Dépendances

Modutils−2.4.12 nécessite que les programmes suivants soient installés:

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/

Contenu

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 une entrée binaire de terminfo avec d'autres entrées 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 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 leurs valeurs par défaut avant d'initialiser le terminal de la même façon que tset.

tack

tack vérifie les actions de terminfo.

tic

Le programme tic est le compilateur des entrées de description de terminfo. Le programme traduit un fichier terminfo du 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 fournie 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

Ces bibliothèques qui forment la bibliothèque Ncurses sont utilisées pour afficher du texte à l'écran. Un exemple où ncurses est utilisé est dans le programme << make menuconfig>> . Les bibliothèques libncurses sont la base du système.

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

Ncurses−5.2 nécessite que les programmes suivants soient installés :

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−0.17

Programmes

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

Netkit−base−0.17 nécessite que les programmes suivants soient installés:

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−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 est utilisé pour manipuler 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 définit le nom du système.

hostname

hostname est utilisé pour définir ou afficher le nom d'hôte du système.

ifconfig

ifconfig est la commande principale pour configurer les interfaces réseau.

nameif

nameif nomme les interfaces réseaux, basées sur les adresses MAC.

netstat

netstat est un utilitaire ayant de nombreux usages; il affiche les connections réseaux, 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 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

Net−tools−1.60 nécessite que les programmes suivants soient installés:

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−2.5.4

Programmes

patch

Descriptions

patch

Le programme patch modifie un fichier en fonction d'un fichier de patch. Un fichier de patch 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 1Mo. 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'1Mo ou juste sous la forme d'un fichier de patch d'1Ko qui viendra mettre à jour la précédente version pour la rendre identique à la dernière version réalisée. Ainsi, si la première version avait déjà été téléchargée, un fichier de patch évite un deuxième téléchargement conséquent.

Dépendances

Patch−2.5.4 nécessite que les programmes suivants soient installés:

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 (5.6.1): http://www.perl.com/

Contenu de perl–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 awk vers perl.

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 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 modules de librairies style Perl5.

pod2html

pod2html convertit en HTML des fichiers au format pod.

pod2latex

pod2latex convertit en HTML des fichiers au format LaTeX.

pod2man

pod2man convertit en HTML des fichiers au format *roff.

pod2text

pod2text convertit des données pod en texte au format ASCII.

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.

Dépendances

Perl−5.6.1 nécessite queles programmes suivants soient installés :

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−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 quel matériel utilise quelles adresses d'entrées−sorties et quels canaux IRQ et DMA.

procinfo

procinfo récupère des informations à partir du répertoire /proc et les affiche proprement formatées 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

Procinfo−18 nécessite que les programmes suivants soient installés:

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/

Contenu de procps–2.0.7

Programmes

free, kill, oldps, pgrep, pkill, ps, skill, snice, sysctl, tload, top, uptime, 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 and 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.

uptime

uptime donne en une ligne les informations suivantes : le temps courant, depuis combien de temps le système tourne−t−il, combien d'utilisateurs sont actuellement connectés, et la charge moyenne du système dans les 1, 5, et 15 dernières minutes.

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 liés pour sauver de l'espace disque, en implémentant des fonctions communes une seule fois.

Dépendances

Procps−2.0.7 nécessite que les programmes suivants soient installés :

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 (20.2): http://download.sourceforge.net/psmisc/
ftp://download.sourceforge.net/pub/sourceforge/psmisc/

Contenu de psmisc–20.2

Programmes

fuser, killall, pidof (lien vers 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 utilisant les fichiers ou systèmes de fichiers qui sont spécifiés.

killall

killall envoie un signal à tous les processus faisant tourner l'une des commandes spécifiées.

pidof

Pidof trouve les identifiants de processus (pids) du programme nommé et imprime ces pids sur la sortie standard.

pstree

pstree présente les processus en cours d'exécution sous forme d'arborescence.

Dépendances

Psmisc–20.2 nécessite que les programmes suivants soient installés :

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

Reiserfsprogs

Site officiel de téléchargement

Reiserfs (3.x.1b): ftp://ftp.namesys.com/pub/reiserfsprogs/

Contenu de reiserfsprogs−3.x.0j

Programmes

debugreiserfs, mkreiserfs, reiserfsck, resize_reiserfs et unpack

Descriptions

debugreiserfs

debugreiserfs peut parfois aider à résoudre des problèmes des systèmes de fichiers reiserfs. S'il est appelé sans options il imprime le superblock de tout système de fichier reiserfs trouvé sur le périphérique.

mkreiserfs

mkreiserfs crée un système de fichiers reiserfs.

reiserfsck

reiserfsck vérifie un système de fichiers reiserfs.

resize_reiserfs

resize_reiserfs est utilisé pour redimensioner un système de fichiers reiserfs démonté.

unpack

Aucune description actuellement disponible.

Dépendances

Reiserfs−3.x.0j nécessite que les programmes suivants soient installés :

autoconf: autoconf, autoheader automake: aclocal, automake bash: sh binutils: ar, as, ld, ranlib
diffutils: cmp fileutils: chmod, install, ls, rm 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

Sed

Site officiel de téléchargement

Sed (3.02): ftp://ftp.gnu.org/gnu/sed/

Contenu de sed−3.02

Programmes

sed

Descriptions

sed

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

Dépendances

Sed−3.02 nécessite que les programmes suivants soient installés :

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 Password Suite

Site officiel de téléchargement

Shadow Password Suite (4.0.3): ftp://ftp.pld.org.pl/software/shadow/

Contenu de shadow–20001016

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, les coordonnées professionnelles, le numéro de téléphone au bureau, et le numéro de téléphone personnel pour un compte utilisateur.

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

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.

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 connexions, /var/log/lastlog. Le nom de login, le port, et l'heure de dernière connexion seront affichés.

login

login est utilisé pour établir une nouvelle session sur le système.

logoutd

logoutd applique les restrictions de temps de connexion et de port de connexion définies 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 informationspour mettre à jour un groupe d'utilisateurs existants 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 informations du système d'authentification.

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 exécute une commande avec un identifiant de groupe différent.

su

Change l'identifiant d'utilisateur et de groupe. Il remplace le programme su du package Shellutils.

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 relatives au nom de login spécifié.

usermod

usermod modifie les fichiers du système de gestion de comptes en focntion 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

libshadow.[a,so]

Descriptions

libshadow

libshadow apporte des fonctions communes aux programmes shadow.

Dépendances

Shadow−20001016 nécessite que les programmes suivants soient installés :

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/ Patch Sh−utils (2.0): ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/ http://ftp.linuxfromscratch.org/lfs−packages/3.3/

Contenu de sh−utils−2.0

Programmes

basename, chroot, date, dirname, echo, env, expr, factor, false, groups, hostid, hostname, 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 l'heure et la date courantes 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 le groupe auquel appartient l'utilisateur.

hostid

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

hostname

hostname met à jour ou affiche le nom courant de l'hôte.

id

id affiche les UIDs et GIDs, réels et effectifs, d'un utilisateur ou de l'utilisateur courant.

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 standard qui ne sera pas de type tty.

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 dans 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 l'identifiant effectif de l'utilisateur.

yes

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

Dépendances

Sh−utils−2.0 nécessite que les programmes suivants soient installés :

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–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 utilisent. 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

Sysklogd−1.4.1 nécessite que les programmes suivants soient installés:

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–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 demande ensuite au noyau de s'arrêter, de redémarrer ou de mettre le système hors tension. Si halt ou reboot sont appelés alors que le système ne se trouve pas dans un niveau d'exécution 0 ou 6, la commande shutdown sera exécutée à la place (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 sur lesquelles les utilisateurs peuvent se connecter. 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) d'après le nom de programme et affiche ces identifiants sur la sortie standard.

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 du niveau d'exécution, puis affiche le niveau d'exécution 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 utilisateur unique (ceci est déterminé dans le fichier /etc/inittab). Init essaie aussi d'exécuter sulogin lorsqu'on le chargeur de boot l'appelle avec l'option −b (e.g., LILO).

telinit

telinit envoie les signaux appropriés à init pour lui spécifier dans quel niveau d'exécution 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

Sysvinit−2.84 nécessite que les programmes suivants soient installés:

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/ Tar Patch (1.13): ftp://ftp.linuxfromscratch.org/lfs−packages/3.3/ http://ftp.linuxfromscratch.org/lfs−packages/3.3/

Contenu de tar−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

Tar−1.13 nécessite que les programmes suivants soient installés :

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 (4.1): ftp://ftp.gnu.org/gnu/texinfo/

Contenu de texinfo−4.0

Programmes

info, 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.

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, il convient également de 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

Texinfo−4.0 nécessite que les programmes suivants soient installés:

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

Text Utils (2.0): ftp://ftp.gnu.org/gnu/textutils/

Contenu de textutils–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 l'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 extrait d'un fichier des fragments séparés par un ou plusieurs motifs, dans des fichiers xx01, xx02, ..., et affiche sur la sortie standard le compte d'octets de chaque segment.

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, et 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

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 chaque fichier spécifié, et une ligne de totaux si plusieurs fichiers ont été spécifiés.

Dépendances

Textutils−2.0 nécessite que les programmes suivants soient installés :

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 (2.11o): ftp://ftp.win.tue.nl/pub/linux−local/utils/util−linux/

Contenu de util−linux−2.11n

Programmes

agetty, arch, blockdev, cal, cfdisk, chkdupexe, col, colcrt, colrm, column, ctrlaltdel, cytune, ddate, dmesg, elvtune, fdformat, fdisk, fsck.minix, getopt, hexdump, hwclock, ipcrm, ipcs, isosize kill, line, logger, look, losetup, mcookie, mkfs, mkfs.bfs, mkfs.minix, mkswap, more, mount, namei, 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, tunelp, ul, umount, vidmode, 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.minix

fsck.minix exécute une vérification logique d'un système de fichiers Linux MINIX.

getopt

getops gère les options de commande de la même façon que la fonction C getopt.

hexdump

hexdump affiche le ou les fichiers spécifiés, 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

kill

kill envoie un signal spécifié au processus spécifié.

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.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.

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 pour la pagination et le swap.

swapon

swapon active les périphériques et les fichiers pour la pagination et le 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

Util−linux−2.11n nécessite que les programmes suivants soient installés:

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/

Contenu

Programmes

ex (lien vers vim), rview (lien vers vim), rvim (lien vers vim), vi (lien vers vim), view (lien vers vim), vim, vimdiff (lien vers vim), vimtutor (lien vers vim) et xxd

Descriptions

ex

ex démarre vim dans le mode Ex.

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.

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.

vimdiff

vimdiff édite deux ou trois versions d'un fichier avec Vim et affiche les différences.

vimtutor

vimtutor démarre le tutoriel de Vim.

xxd

xxd crée un dump hexadécimal ou fait l'inverse.

Dépendances

Vim−6.0 nécessite que les programmes suivants soient installés:

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

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.linuxdoc.org

- Administration Réseau sous Linux
- From−PowerUp−To−Bash−Prompt−HOWTO

Autres

- Les différentes pages man et info distribuées avec les packages.