Linux From Scratch

Table of Contents

Linux From Scratch....1
Gerard Beekmans....1

Dédicace....2

Préface....6

A qui s'adresse ce livre....7

A qui ne s'adresse pas ce livre....8

Organisation....9
Première Partie – Introduction....9
Deuxième partie – Installation du système LFS....9
Troisième partie – Annexes....9

I. Première Partie – Introduction....10

Chapitre 1. Introduction....11

Introduction....12

Comment les choses se dérouleront–elles....13

Versions du livre....14
Miroirs HTTP....14
Miroirs FTP....14

Remerciements....16

Liste des changements....17

Listes de diffusion et archives....32
lfs–discuss....32
lfs–apps....32
lfs–announce....32
lfs–security....33
alfs–discuss....33
alfs–docs....33
alfs–ipc....33
alfs–profile....33
Comment s'inscrire ?....33
Comment se désabonner ?....34
Archives des listes de diffusion....34

Points de contact....35

Chapitre 2. Information Importante....36

Table of Contents

A propos de $LFS....37

Comment télécharger le logiciel....38

Comment installer le logiciel....39

Télécharger les scripts de démarrage....41

Télécharger les commandes LFS....42

II. Deuxième Partie – Installation du système LFS....43

Chapitre 3. Paquetages que vous devez télécharger....44

Chapitre 4. Préparation d'une nouvelle partition....50

Introduction....51

Création d'une nouvelle partition....52

Création d'un système de fichiers ext2 dans la nouvelle partition....53

Montage d'une nouvelle partition....54

Création des répertoires....55

Chapitre 5. Préparation du système LFS....56

Introduction....57

Installer Bash....58
- Installation de Bash....58
- Explication des commandes....58
- Contenu....59
- Description....59

Installer Binutils....60
- Installation de Binutils....60
- Explication des commandes....60
- Contenu....60
- Description....60
  - ld....60
  - as....61
  - ar....61
  - nm....61
  - objcopy....61
  - objdump....61
  - ranlib....61

Table of Contents

size....61
strings....62
strip....62
c++filt....62
addr2line....62
nlmconv....62

Installer Bzip2....63
Installation de Bzip2....63
Explication des commandes....63
Contenu....63
Description....63
bzip2....64
bunzip2....64
bzcat....64
bzip2recover....64

Installer Diffutils....65
Installation de Diffutils....65
Explication des commandes....65
Contenu....65
Description....65
cmp et diff....65
diff3....65
sdiff....66

Installer Fileutils....67
Installation de Fileutils....67
Explication de la commande....67
Contenu....67
Description....67
chgrp....67
chmod....67
chown....68
cp....68
dd....68
df....68
ls, dir et vdir....68
dircolors....68
du....68
install....68
ln....69
mkdir....69
mkfifo....69
mknod....69
mv....69
rm....69
rmdir....69

Table of Contents

sync....69
touch....69

Installer GCC sur le système normal si nécessaire....71
Installation de GCC sur le système normal si nécessaire....71
Explication des commandes....71
Contenu....72
Description....72
Compilateur....72
Préprocesseur....72
Bibliothèque C++....72

Installer GCC sur le système LFS....73
Installation de GCC sur le système LFS....73
Explication des commandes....73
Contenu....73
Description....73
Compilateur....74
Préprocesseur....74
Bibliothèque C++....74

Installer le noyau Linux....75
Installation du noyau Linux....75
Explication des commandes....75
Contenu....76
Description....76

Installer Glibc....77
Installation de Glibc....77
Copier les anciennes bibliothèques NSS....78
Explication des commandes....78
Contenu....79
Description....79

Installer Grep....80
Installation de Grep....80
Contenu....80
Description....80
egrep....80
fgrep....80
grep....80

Installer Gzip....81
Installation de Gzip....81
Contenu....81
Description....81
gunzip....81
gzexe....81

Table of Contents

gzip....81
zcat....81
zcmp....82
zdiff....82
zforce....82
zgrep....82
zmore....82
znew....82

Installer Make....83
Installation de Make....83
Contenu....83
Description....83

Installer Sed....84
Installation de Sed....84
Contenu....84
Description....84

Installer Shellutils....85
Installation de Sh−utils....85
Contenu....85
Description....85
basename....85
chroot....85
date....85
dirname....85
echo....86
env....86
expr....86
factor....86
false....86
groups....86
hostid....86
hostname....86
id....86
logname....87
nice....87
nohup....87
pathchk....87
pinky....87
printenv....87
printf....87
pwd....87
seq....87
sleep....88
stty....88
su....88

Table of Contents

tee....88
test....88
true....88
tty....88
uname....88
uptime....88
users....89
who....89
whoami....89
yes....89

Installer Tar....90
Installation de Tar....90
Contenu....90
Description....90
tar....90
rmt....90

Installer Textutils....91
Installation de Textutils....91
Contenu....91
Description....91
cat....91
cksum....91
comm....91
csplit....91
cut....92
expand....92
fmt....92
fold....92
head....92
join....92
md5sum....92
nl....92
od....92
paste....93
pr....93
ptx....93
sort....93
split....93
sum....93
tac....93
tail....93
tr....93
tsort....94
unexpand....94
uniq....94
wc....94

Table of Contents

Créer les fichiers passwd et group....95

Monter le système de fichiers $LFS/proc....96

Chapitre 6. Installer les logiciels du système de base....97

Introduction....98

A propos des symboles de déboguage....99

Créer $LFS/root/.bash_profile....100

Entrer dans l'environnement chroot....101

Créer les fichiers de périphérique....102
- Installation de MAKEDEV....102
- Créer les entrées de /dev....102
- Explication des commandes....102
- Contenu....103
- Description....103

Installer Man−pages....104
- Installation de Man−pages....104
- Contenu....104
- Description....104

Installer Ed....105
- Installation de Ed....105
- Contenu....105
- Description....105

Installer Patch....106
- Installation de Patch....106
- Contenu....106
- Description....106

Installer Findutils....107
- Installer Findutils....107
- Contenu....107
- Description....107
  - find....107
  - locate....107
  - updatedb....107
  - xargs....108

Installer Mawk....109
- Installation de Mawk....109
- Contenu....109

Table of Contents

Description....109
mawk....109

Installer Ncurses....110
Installation de Ncurses....110
Explication des commandes....110
Contenu....110
Description....110
Les bibliothèques....110
tic....110
infocmp....111
clear....111
tput....111
toe....111
tset....111

Installer Vim....112
Installation de Vim....112
Contenu....112
Description....112
ctags....112
etags....112
ex....113
gview....113
gvim....113
rgview....113
rgvim....113
rview....113
rvim....113
view....113
vim....113
vimtutor....114
xxd....114

Installer GCC....115
Installation de GCC....115
Contenu....115
Description....115
Compilateur....115
Préprocesseur....115
Bibliothèque C++....115

Installer Bison....116
Installation de Bison....116
Explication des commandes....116
Contenu....116
Description....116

Table of Contents

Installer Less....118
Installation de Less....118
Contenu....118
Description....118

Installer Groff....119
Installation de Groff....119
Contenu....119
Description....119
addftinfo....119
afmtodit....119
eqn....119
grodvi....119
groff....120
grog....120
grohtml....120
grolj4....120
grops....120
grotty....120
hpftodit....120
indxbib....120
lkbib....121
lookbib....121
neqn....121
nroff....121
pfbtops....121
pic....121
psbb....121
refer....121
soelim....122
tbl....122
tfmtodit....122
troff....122

Installer Man....123
Installation de Man....123
Explication des commandes....123
Contenu....123
Description....123
man....123
apropos....123
whatis....124
makewhatis....124

Installer Perl....125
Installation de Perl....125
Contenu....125
Description....125

Table of Contents

Installer M4 ..... **126**
Installation de M4 126
Contenu 126
Description 127

Installer Texinfo **128**
Installation de Texinfo 128
Contenu 128
Description 128
info 128
install–info 128
makeinfo 128
texi2dvi 128
texindex 129

Installer Autoconf **130**
Installation d'Autoconf 130
Contenu 130
Description 130
autoconf 130
autoheader 130
autoreconf 130
autoscan 130
autoupdate 131
ifnames 131

Installer Automake **132**
Installation d'Automake 132
Contenu 132
Description 132
aclocal 132
automake 132

Installer Bash **133**
Installation de Bash 133
Contenu 133
Description 133

Installer Flex **134**
Installation de Flex 134
Contenu 134
Description 134

Installer File **135**
Installation de File 135
Contenu 135
Description 135

Table of Contents

Installer Libtool....136
Installation de Libtool....136
Contenu....136
Description....136
libtool....136
libtoolize....136
Bibliothèque ltdl....136

Installer Bin86....137
Installation de Bin86....137
Contenu....137
Description....137
as86....137
as86_encap....137
ld86....137
objdump86....137
nm86....137
size86....138

Installer Binutils....139
Installation de Binutils....139
Contenu....139
Description....139
ld....139
as....139
ar....139
nm....139
objcopy....140
objdump....140
ranlib....140
size....140
strings....140
strip....140
c++filt....141
addr2line....141
nlmconv....141

Installer Bzip2....142
Installation de Bzip2....142
Explication des commandes....142
Contenu....142
Description....143
bzip2....143
bunzip2....143
bzcat....143
bzip2recover....143

Installer Gettext....144

Table of Contents

Installation de Gettext....144
Contenu....144
Description....144
gettext....144

Installer Console−tools....145
Installation de Console−tools....145
Contenu....145
Description....145
charset....145
chvt....145
codepage....145
consolechars....146
deallocvt....146
dumpkeys....146
fgconsole....146
fix_bs_and_del....146
font2psf....146
getkeycodes....146
kbd_mode....146
loadkeys....146
loadunimap....147
mapscrn....147
mk_modmap....147
openvt....147
psfaddtable....147
psfgettable....147
psfstriptable....147
resizecons....147
saveunimap....147
screendump....148
setfont....148
setkeycodes....148
setleds....148
setmetamode....148
setvesablank....148
showcfont....148
showkey....148
splitfont....148
unicode_start....149
unicode_stop....149
vcstime....149
vt−is−UTF8....149
writevt....149

Installer Console−data....150
Installation de Console−data....150
Contenu....150

Table of Contents

Installer Diffutils....151
Installation de Diffutils....151
Contenu....151
Description....151
cmp et diff....151
diff3....151
sdiff....151

Installer E2fsprogs....152
Installation de E2fsprogs....152
Contenu....152
Description....152
chattr....152
lsattr....152
uuidgen....152
badblocks....152
debugfs....153
dumpe2fs....153
e2fsck et fsck.ext2....153
e2label....153
fsck....153
mke2fs et mkfs.ext2....153
mklost+found....153
tune2fs....153

Installer Fileutils....155
Installation de Fileutils....155
Contenu....155
Description....155
chgrp....155
chmod....155
chown....155
cp....155
dd....156
df....156
ls, dir et vdir....156
dircolors....156
du....156
install....156
ln....156
mkdir....156
mkfifo....157
mknod....157
mv....157
rm....157
rmdir....157
sync....157
touch....157

Table of Contents

Installer Grep....158
Installation de Grep....158
Contenu....158
Description....158
egrep....158
fgrep....158
grep....158

Installer Gzip....159
Installation de Gzip....159
Contenu....159
Description....159
gunzip....159
gzexe....159
gzip....159
zcat....160
zcmp....160
zdiff....160
zforce....160
zgrep....160
zmore....160
znew....160

Installer Ld.so....161
Installation de Ld.so....161
Contenu....161
Description....161
ldconfig....161
ldd....162

Installer Lilo....163
Installation de Lilo....163
Contenu....163
Description....163

Installer Make....164
Installation de Make....164
Contenu....164
Description....164

Installer Modutils....165
Installation de Modutils....165
Contenu....165
Description....165
depmod....165
genksyms....165
insmod....165
insmod_ksymoops_clean....165

Table of Contents

kerneld....166
kernelversion....166
ksyms....166
lsmod....166
modinfo....166
modprobe....166
rmmod....166

Installer Procinfo....167
Installation de Procinfo....167
Explication des commandes....167
Contenu....167
Description....167

Installer Procps....168
Installation de Procps....168
Contenu....168
Description....168
free....168
kill....168
oldps and ps....168
skill....168
snice....168
sysctl....169
tload....169
top....169
uptime....169
vmstat....169
w....169
watch....169

Installer Psmisc....170
Installation de Psmisc....170
Contenu....170
Description....170
fuser....170
killall....170
pstree....170

Installer Sed....171
Installation de Sed....171
Contenu....171
Description....171

Installer Shellutils....172
Installation de Shellutils....172
Contenu....172
Description....172

Table of Contents

basename....172
chroot....172
date....172
dirname....172
echo....173
env....173
expr....173
factor....173
false....173
groups....173
hostid....173
hostname....173
id....173
logname....174
nice....174
nohup....174
pathchk....174
pinky....174
printenv....174
printf....174
pwd....174
seq....174
sleep....175
stty....175
su....175
tee....175
test....175
true....175
tty....175
uname....175
uptime....175
users....176
who....176
whoami....176
yes....176

Installer Shadowpwd....177
Installation de la Suite de Masquage des mots de passe....177
Explication des commandes....177
Contenu....177
Description....177
chage....177
chfn....177
chsh....178
expiry....178
faillog....178
gpasswd....178
lastlog....178

Table of Contents

login....178
newgrp....178
passwd....178
sg....178
su....179
chpasswd....179
dpasswd....179
groupadd....179
groupdel....179
groupmod....179
grpck....179
grpconv....179
grpunconv....180
logoutd....180
mkpasswd....180
newusers....180
pwck....180
pwconv....180
pwunconv....180
useradd....180
userdel....180
usermod....181
vipw et vigr....181

Installer Sysklogd....182
Installation de Sysklogd....182
Contenu....182
Description....182
klogd....182
syslogd....182

Installer Sysvinit....183
Installation de Sysvinit....183
Contenu....183
Description....183
pidof....183
last....183
lastb....184
mesg....184
utmpdump....184
wall....184
halt....184
init....184
killall5....184
poweroff....184
reboot....185
runlevel....185
shutdown....185

Table of Contents

sulogin....185
telinit....185

Installer Tar....186
Installation de Tar....186
Contenu....186
Description....186
tar....186
rmt....186

Installer Textutils....187
Installation de Textutils....187
Contenu....187
Description....187
cat....187
cksum....187
comm....187
csplit....187
cut....188
expand....188
fmt....188
fold....188
head....188
join....188
md5sum....188
nl....188
od....188
paste....189
pr....189
ptx....189
sort....189
split....189
sum....189
tac....189
tail....189
tr....189
tsort....190
unexpand....190
uniq....190
wc....190

Installer Utillinux....191
Installation de Util−Linux....191
Explication des commandes....191
Contenu....191
Description....191
arch....191
dmesg....191

Table of Contents

kill....192
more....192
mount....192
umount....192
agetty....192
blockdev....192
cfdisk....192
ctrlaltdel....192
elvtune....192
fdisk....193
fsck.minix....193
hwclock....193
kbdrate....193
losetup....193
mkfs....193
mkfs.bfs....193
mkfs.minix....193
mkswap....193
sfdisk....194
swapoff....194
swapon....194
cal....194
chkdupexe....194
col....194
colcrt....194
colrm....194
column....194
cytune....195
ddate....195
fdformat....195
getopt....195
hexdump....195
ipcrm....195
ipcs....195
logger....195
look....195
mcookie....196
namei....196
rename....196
renice....196
rev....196
script....196
setfdprm....196
setsid....196
setterm....196
ul....197
whereis....197
write....197

Table of Contents

ramsize....197
rdev....197
readprofile....197
rootflags....197
swapdev....197
tunelp....197
vidmode....198

Suppression des anciens fichiers de la bibliothèque NSS....199

Configurer les logiciels essentiels....200
Configurer Vim....200
Configurer Glibc....200
Configurer le Chargeur Dynamique....201
Configurer Lilo....202
Configurer Sysklogd....202
Configurer la Suite Masquage des Mots de passe (Shadow passwords)....203
Configurer Sysvinit....203
Créer les fichiers /var/run/utmp, /var/log/wtmp et /var/log/btmp....204
Créer le mot de passe de root....204

Chapitre 7. Création des scripts de démarrage du système....206

Introduction....207

Création des répertoires....208

Création du script rc....209

Création du script rcS....214

Création du script functions....215

Création du script checkfs....224

Création du script halt....227

Création du script loadkeys....228

Création du script mountfs....229

Création du script de redémarrage (reboot)....231

Création du script sendsignals....232

Création du script setclock....233
Création du fichier /etc/sysconfig/clock....234

Table of Contents

Création du script sysklogd....235

Création du script modèle (template)....237

Création des liens symboliques et réglage des permissions....239

Création du fichier/etc/fstab file....240

Chapitre 8. Rendre amorçable le système LFS....241

Introduction....242

Installer un noyau....243

Ajouter une entrée à LILO....244

Redémarrer le système....245

Chapitre 9. Configurer un réseau de base....246

Introduction....247

Installer les logiciels de réseau....248
Installer Netkit−base....248
Installer Net−tools....248

Créer le script de démarrage /etc/init.d/localnet....249
Configurer les permissions et le lien symbolique....250

Créer le fichier /etc/sysconfig/network....251

Créer le fichier /etc/hosts....252

Créer le script /etc/init.d/ethnet....254
Ajouter la passerelle par défaut à /etc/sysconfig/network....256
Créer les fichiers de configuration NIC....256
Configurer les permissions et les liens symboliques....257

III. Troisième Partie − Annexes....258

Annexe A. Description des paquets....259

Introduction....260

Glibc....261
Contenu....261
Description....261

Table of Contents

Noyau Linux **262**
Contenu 262
Description 262

Ed **263**
Contenu 263
Description 263

Patch **264**
Contenu 264
Description 264

GCC **265**
Contenu 265
Description 265
Compilateur 265
Préprocesseur 265
Bibliothèque C++ 265

Bison **266**
Contenu 266
Description 266

Mawk **267**
Contenu 267
Description 267
mawk 267

Findutils **268**
Contenu 268
Description 268
find 268
locate 268
updatedb 268
xargs 268

Ncurses **269**
Contenu 269
Description 269
Les bibliothèques 269
tic 269
infocmp 269
clear 269
tput 269
toe 270
tset 270

Less **271**

Table of Contents

Contenu 271
Description 271

Groff 272
Contenu 272
Description 272
addftinfo 272
afmtodit 272
eqn 272
grodvi 272
groff 272
grog 272
grohtml 273
grolj4 273
grops 273
grotty 273
hpftodit 273
indxbib 273
lkbib 273
lookbib 273
neqn 274
nroff 274
pfbtops 274
pic 274
psbb 274
refer 274
soelim 274
tbl 274
tfmtodit 275
troff 275

Man 276
Contenu 276
Description 276
man 276
apropos 276
whatis 276
makewhatis 276

Perl 277
Contenu 277
Description 277

M4 278
Contenu 278
Description 278

Texinfo 279

Table of Contents

Contenu....279
Description....279
info....279
install−info....279
makeinfo....279
texi2dvi....279
texindex....279

Autoconf....280
Contenu....280
Description....280
autoconf....280
autoheader....280
autoreconf....280
autoscan....280
autoupdate....280
ifnames....281

Automake....282
Contenu....282
Description....282
aclocal....282
automake....282

Bash....283
Contenu....283
Description....283

Flex....284
Contenu....284
Description....284

Binutils....285
Contenu....285
Description....285
ld....285
as....285
ar....285
nm....285
objcopy....285
objdump....285
ranlib....286
size....286
strings....286
strip....286
c++filt....286
addr2line....286
nlmconv....287

Table of Contents

Bzip2 ..... **288**
Contenu 288
Description 288
bzip2 288
bunzip2 288
bzcat 288
bzip2recover 288

Diffutils **289**
Contenu 289
Description 289
cmp et diff 289
diff3 289
sdiff 289

E2fsprogs **290**
Contenu 290
Description 290
chattr 290
lsattr 290
uuidgen 290
badblocks 290
debugfs 290
dumpe2fs 290
e2fsck et fsck.ext2 291
e2label 291
fsck 291
mke2fs et mkfs.ext2 291
mklost+found 291
tune2fs 291

File **292**
Contenu 292
Description 292

Fileutils **293**
Contenu 293
Description 293
chgrp 293
chmod 293
chown 293
cp 293
dd 293
df 293
ls, dir et vdir 294
dircolors 294
du 294
install 294

Table of Contents

ln....294
mkdir....294
mkfifo....294
mknod....294
mv....295
rm....295
rmdir....295
sync....295
touch....295

Gettext....296
Contenu....296
Description....296
gettext....296

Grep....297
Contenu....297
Description....297
egrep....297
fgrep....297
grep....297

Gzip....298
Contenu....298
Description....298
gunzip....298
gzexe....298
gzip....298
zcat....298
zcmp....298
zdiff....298
zforce....299
zgrep....299
zmore....299
znew....299

Ld.so....300
Contenu....300
Description....300
ldconfig....300
ldd....300

Libtool....301
Contenu....301
Description....301
libtool....301
libtoolize....301
Bibliothèque ltdl....301

Table of Contents

Bin86....302
Contenu....302
Description....302
as86....302
as86_encap....302
ld86....302
objdump86....302
nm86....302
size86....302

Lilo....303
Contenu....303
Description....303

Make....304
Contenu....304
Description....304

Shellutils....305
Contenu....305
Description....305
basename....305
chroot....305
date....305
dirname....305
echo....305
env....305
expr....305
factor....306
false....306
groups....306
hostid....306
hostname....306
id....306
logname....306
nice....306
nohup....306
pathchk....307
pinky....307
printenv....307
printf....307
pwd....307
seq....307
sleep....307
stty....307
su....307
tee....308
test....308

Table of Contents

true....308
tty....308
uname....308
uptime....308
users....308
who....308
whoami....308
yes....309

Suite de Masquage des mots de passe....310
Contenu....310
Description....310
chage....310
chfn....310
chsh....310
expiry....310
faillog....310
gpasswd....310
lastlog....311
login....311
newgrp....311
passwd....311
sg....311
su....311
chpasswd....311
dpasswd....311
groupadd....311
groupdel....312
groupmod....312
grpck....312
grpconv....312
grpunconv....312
logoutd....312
mkpasswd....312
newusers....312
pwck....313
pwconv....313
pwunconv....313
useradd....313
userdel....313
usermod....313
vipw et vigr....313

Modutils....314
Contenu....314
Description....314
depmod....314
genksyms....314

Table of Contents

insmod....314
insmod_ksymoops_clean....314
kerneld....314
kernelversion....314
ksyms....315
lsmod....315
modinfo....315
modprobe....315
rmmod....315

Procinfo....316
Contenu....316
Description....316

Procps....317
Contenu....317
Description....317
free....317
kill....317
oldps and ps....317
skill....317
snice....317
sysctl....317
tload....317
top....318
uptime....318
vmstat....318
w....318
watch....318

Vim....319
Contenu....319
Description....319
ctags....319
etags....319
ex....319
gview....319
gvim....319
rgview....319
rgvim....319
rview....320
rvim....320
view....320
vim....320
vimtutor....320
xxd....320

Psmisc....321

Table of Contents

Contenu....321
Description....321
fuser....321
killall....321
pstree....321

Sed....322
Contenu....322
Description....322

Sysklogd....323
Contenu....323
Description....323
klogd....323
syslogd....323

Sysvinit....324
Contenu....324
Description....324
pidof....324
last....324
lastb....324
mesg....324
utmpdump....324
wall....324
halt....325
init....325
killall5....325
poweroff....325
reboot....325
runlevel....325
shutdown....325
sulogin....326
telinit....326

Tar....327
Contenu....327
Description....327
tar....327
rmt....327

Textutils....328
Contenu....328
Description....328
cat....328
cksum....328
comm....328
csplit....328

Table of Contents

cut....328
expand....328
fmt....328
fold....329
head....329
join....329
md5sum....329
nl....329
od....329
paste....329
pr....329
ptx....329
sort....330
split....330
sum....330
tac....330
tail....330
tr....330
tsort....330
unexpand....330
uniq....330
wc....331

Util Linux....332
Contenu....332
Description....332
arch....332
dmesg....332
kill....332
more....332
mount....332
umount....332
agetty....333
blockdev....333
cfdisk....333
ctrlaltdel....333
elvtune....333
fdisk....333
fsck.minix....333
hwclock....333
kbdrate....333
losetup....334
mkfs....334
mkfs.bfs....334
mkfs.minix....334
mkswap....334
sfdisk....334
swapoff....334

Table of Contents

swapon....334
cal....334
chkdupexe....335
col....335
colcrt....335
colrm....335
column....335
cytune....335
ddate....335
fdformat....335
getopt....335
hexdump....336
ipcrm....336
ipcs....336
logger....336
look....336
mcookie....336
namei....336
rename....336
renice....336
rev....337
script....337
setfdprm....337
setsid....337
setterm....337
ul....337
whereis....337
write....337
ramsize....337
rdev....338
readprofile....338
rootflags....338
swapdev....338
tunelp....338
vidmode....338

Console−tools....339
Contenu....339
Description....339
charset....339
chvt....339
codepage....339
consolechars....339
deallocvt....339
dumpkeys....339
fgconsole....340
fix_bs_and_del....340
font2psf....340

Table of Contents

getkeycodes....340
kbd_mode....340
loadkeys....340
loadunimap....340
mapscrn....340
mk_modmap....340
openvt....341
psfaddtable....341
psfgettable....341
psfstriptable....341
resizecons....341
saveunimap....341
screendump....341
setfont....341
setkeycodes....341
setleds....342
setmetamode....342
setvesablank....342
showcfont....342
showkey....342
splitfont....342
unicode_start....342
unicode_stop....342
vcstime....342
vt−is−UTF8....343
writevt....343

Console−data....344
Contenu....344

Man−pages....345
Contenu....345
Description....345

Annexe B. Ressources....346

Introduction....347

Livres....348

HOWTOs et Guides....349

Autres....350

Annexe C. Sites de téléchargement officiels....351

Linux From Scratch

Gerard Beekmans

Copyright © 1999, 2000, 2001 par Gerard Beekmans

Ce livre décrit le processus de création de votre propre système Linux à partir de zéro, depuis une distribution Linux déjà installée, en utilisant seulement les sources des logiciels nécessaires.

Copyright (c) 1999−2001, 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 of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of LinuxFromScratch nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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 au soutient indéfectible *Beverly Beekmans*.

Table des matières

Préface
- *A qui s'adresse ce livre*
- *A qui ne s'adresse pas ce livre*
- *Organisation*
  - *Première Partie − Introduction*
  - *Deuxième partie − Installation du système LFS*
  - *Troisième partie − Annexes*

I. Première Partie − Introduction
- *1. Introduction*
  - *Introduction*
  - *Comment les choses se dérouleront−elles*
  - *Versions du livre*
  - *Remerciements*
  - *Liste des changements*
  - *Listes de diffusion et archives*
  - *Points de contact*
- *2. Information Importante*
  - *A propos de $LFS*
  - *Comment télécharger le logiciel*
  - *Comment installer le logiciel*
  - *Télécharger les scripts de démarrage*
  - *Télécharger les commandes LFS*

II. Deuxième Partie − Installation du système LFS
- *3. Paquetages que vous devez télécharger*
- *4. Préparation d'une nouvelle partition*
  - *Introduction*
  - *Création d'une nouvelle partition*
  - *Création d'un système de fichiers ext2 dans la nouvelle partition*
  - *Montage d'une nouvelle partition*
  - *Création des répertoires*
- *5. Préparation du système LFS*
  - *Introduction*
  - *Installer Bash*
  - *Installer Binutils*
  - *Installer Bzip2*
  - *Installer Diffutils*
  - *Installer Fileutils*
  - *Installer GCC sur le système normal si nécessaire*
  - *Installer GCC sur le système LFS*
  - *Installer le noyau Linux*
  - *Installer Glibc*
  - *Installer Grep*
  - *Installer Gzip*
  - *Installer Make*
  - *Installer Sed*

Installer Shellutils
Installer Tar
Installer Textutils
Créer les fichiers passwd et group
Monter le système de fichiers $LFS/proc
6. Installer les logiciels du système de base
Introduction
A propos des symboles de déboguage
Créer $LFS/root/.bash_profile
Entrer dans l'environnement chroot
Créer les fichiers de périphérique
Installer Man−pages
Installer Ed
Installer Patch
Installer Findutils
Installer Mawk
Installer Ncurses
Installer Vim
Installer GCC
Installer Bison
Installer Less
Installer Groff
Installer Man
Installer Perl
Installer M4
Installer Texinfo
Installer Autoconf
Installer Automake
Installer Bash
Installer Flex
Installer File
Installer Libtool
Installer Bin86
Installer Binutils
Installer Bzip2
Installer Gettext
Installer Console−tools
Installer Console−data
Installer Diffutils
Installer E2fsprogs
Installer Fileutils
Installer Grep
Installer Gzip
Installer Ld.so
Installer Lilo
Installer Make
Installer Modutils
Installer Procinfo
Installer Procps
Installer Psmisc
Installer Sed

Installer Shellutils
Installer Shadowpwd
Installer Sysklogd
Installer Sysvinit
Installer Tar
Installer Textutils
Installer Utillinux
Suppression des anciens fichiers de la bibliothèque NSS
Configurer les logiciels essentiels
7. Création des scripts de démarrage du système
Introduction
Création des répertoires
Création du script rc
Création du script rcS
Création du script functions
Création du script checkfs
Création du script halt
Création du script loadkeys
Création du script mountfs
Création du script de redémarrage (reboot)
Création du script sendsignals
Création du script setclock
Création du script sysklogd
Création du script modèle (template)
Création des liens symboliques et réglage des permissions
Création du fichier/etc/fstab file
8. Rendre amorçable le système LFS
Introduction
Installer un noyau
Ajouter une entrée à LILO
Redémarrer le système
9. Configurer un réseau de base
Introduction
Installer les logiciels de réseau
Créer le script de démarrage /etc/init.d/localnet
Créer le fichier /etc/sysconfig/network
Créer le fichier /etc/hosts
Créer le script /etc/init.d/ethnet
III. Troisième Partie − Annexes
A. Description des paquets
Introduction
Glibc
Noyau Linux
Ed
Patch
GCC
Bison
Mawk
Findutils
Ncurses
Less

Groff
Man
Perl
M4
Texinfo
Autoconf
Automake
Bash
Flex
Binutils
Bzip2
Diffutils
E2fsprogs
File
Fileutils
Gettext
Grep
Gzip
Ld.so
Libtool
Bin86
Lilo
Make
Shellutils
Suite de Masquage des mots de passe
Modutils
Procinfo
Procps
Vim
Psmisc
Sed
Sysklogd
Sysvinit
Tar
Textutils
Util Linux
Console−tools
Console−data
Man−pages
B. Ressources
Introduction
Livres
HOWTOs et Guides
Autres
C. Sites de téléchargement officiels

Préface

A qui s'adresse ce livre

Ce livre est destiné aux utilisateurs de Linux qui veulent en apprendre plus sur le fonctionnement interne de Linux et savoir comment les différents morceaux du système d'exploitation s'assemblent. Ce livre vous guidera pas à pas tout au long de la création de votre système Linux personnalisé, à partir de zéro, en utilisant le code source des logiciels dont vous avez besoin.

Ce livre est aussi destiné aux utilisateurs de Linux qui veulent s'éloigner des distributions existantes, libres ou commerciales, qui sont souvent surchargées. Utiliser une distribution existante vous force aussi à adopter la structure du système de fichiers, des scripts de démarrage, etc... que ses créateurs ont choisie. Avec ce livre, vous pouvez créer votre structure et votre méthode personnelles, exactement comme vous les préférez (elles peuvent être basées sur celles que le livre fournit)

De plus, si vous êtes soucieux de la sécurité, vous ne voulez pas vous reposer sur des paquetages pré−compilés. A l'inverse, vous voulez compiler tous les programmes de zéro et les installer vous même. Cela peut constituer une autre raison pour laquelle vous voudriez construire un système Linux personnalisé.

Voici, parmi beaucoup d'autre, un petit nombre de raisons qui expliquent pourquoi certains veulent construire leur système Linux personnel. Si vous en êtes, ce livre vous est destiné.

A qui ne s'adresse pas ce livre

Les utilisateurs qui ne veulent pas construire un système Linux entièrement à partir de zéro n'ont pas besoin de lire ce livre. Cependant, si vous voulez en apprendre plus sur ce qui se passe en coulisse, en particulier ce qui arrive entre le moment où vous mettez en marche l'ordinateur, et celui où vous voyez l'invite de la ligne de commande, lisez le HOWTO "From Power Up To Bash Prompt" (P2B). De la même façon que ce livre, cet HOWTO construit un système nu, mais se concentre plus sur l'installation d'un système démarrable que sur celle d'un système complet.

Pour décider si vous devez lire ce livre ou le HOWTO P2B, vous pouvez vous poser cette question : Est−ce que mon principal objectif est d'obtenir un système Linux fonctionnel que j'aurais construit moi−même et, chemin faisant, d'apprendre à quoi sert chaque composant du système, ou est−ce que mon principal objectif est seulement la partie apprentissage ? Si vous voulez construire et apprendre, lisez ce livre. Si vous voulez simplement apprendre, alors il est sans doute plus indiqué de lire le HOWTO P2B.

Le HOWTO "From Power Up To Bash Prompt" peut être obtenu à l'adresse http://www.netspace.net.au/~gok/power2bash/

Organisation

Ce livre est divisé en les parties suivantes. Bien qu'il y ait de nombreuses informations dupliquées dans certaines parties, cette organisation facilite la lecture du livre, pour ne pas mentionner que, pour moi, elle est la façon la plus aisée de le maintenir à jour.

Première Partie – Introduction

la première partie apporte une information générale sue ce livre (versions, où l'obtenir, liste des modifications, listes de diffusions, et comment me contacter). Elle explique aussi un petit nombre d'aspects importants dont vous avez besoin et que vous voulez lire avant de commencer à construire un système LFS.

Deuxième partie – Installation du système LFS

la deuxième partie vous guide à travers l'installation du système LFS qui sera la fondation du reste du système. Quelle que soit l'utilisation que vous choisirez pour votre système LFS flambant neuf, il sera construit sur la fondation qui est installée dans cette partie.

Troisième partie – Annexes

la troisième partie contient diverses annexes.

I. Première Partie – Introduction

Table des matières

1. Introduction
2. Information Importante

Chapitre 1. Introduction

Introduction

Ayant utilisé un certain nombre de distributions Linux différentes, je n'ai jamais été complètement satisfait par l'une d'elle. Je n'aimais pas la manière dont les scripts de démarrage étaient organisés, la façon dont certains programmes étaient configurés par défaut, et plein d'autres choses. J'en suis venu à considérer que si je voulais être totalement satisfait avec un système Linux, je devais le construire moi−même en partant de zéro, l'idéal étant de n'utiliser que le code source. Sans jamais utiliser un quelconque paquetage précompilé. Aucune aide venant d'un cdrom ou d'une disquette de démarrage qui installerait des utilitaires de base. Votre système Linux actuel vous servirait à construire le vôtre.

Au début, cette idée insensée semblait extrêmement difficile à réaliser et, par moments, presque impossible. La raison de la plupart des problèmes était due à mon manque de connaissance de certains programmes et de marches à suivre. Après avoir fait un tri de tous les types de problèmes de dépendance, de compilation, etc... un système Linux personnalisé à été créé et rendu totalement opérationnel. Je l'ai appelé LFS, ce qui signifie LinuxFromScratch (NdT : en français, "Linux en partant de zéro").

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. Vous n'avez besoin d'aucune disquette de démarrage. Nous utiliserons un système Linux existant comme base (puisque nous avons besoin d'un compilateur, d'un éditeur de liens, d'un éditeur de texte et d'autres outils).

Si vous n'avez pas encore installé Linux, vous ne pourrez pas utiliser correctement ce livre. Je vous conseille d'installer tout d'abord une distribution Linux. Le choix de cette distribution importe peu. Il n'est pas nécessaire non plus d'avoir la dernière version, même si elle ne doit pas être trop vieille. Si elle date de moins d'un an, vous ne devriez pas avoir de problème. Vous règlerez vous–même de nombreuses difficultés si votre système normal utilise le glibc–2.1 ou ultérieure. La libc5 n'est pas supportée par cet ouvrage, bien qu'il ne soit pas impossible d'utiliser un système avec la libc5 si vous n'avez pas le choix.

Versions du livre

Ceci est le livre LFS INTEL version 2.4.4 daté du Le 23 Janvier 2001. Si cette version date de plus d'un mois, vous devriez jeter un coup d'oeil sur le site web et vérifier s'il n'existe pas une version plus récente disponible au téléchargement.

Vous trouverez ci−dessous une liste de nos sites miroirs HTTP et FTP en date du 19 décembre 2000. Cette liste peut ne plus être exacte. Pour obtenir les dernières informations, consultez notre site web à http://www.linuxfromscratch.org

---

Miroirs HTTP

- Columbus, Ohio, Etats Unis − http://www.linuxfromscratch.org/intro/
- Etats Unis − http://lfs.sourceforge.net/intro/
- Canmore, Alberta, Canada − http://www.ca.linuxfromscratch.org/intro/
- Braunschweig, Niedersachsen, Allemagne − http://www.de.linuxfromscratch.org/intro/
- Mainz, Allemagne, Europe − http://lfs.linux−provider.net/intro/
- Australie (accessible seulement depuis la Nouvelle−Zélande et l'Australie) − http://lfs.mirror.aarnet.edu.au/intro/

Miroirs FTP

- Columbus, Ohio, Etats Unis − ftp://packages.linuxfromscratch.org
- Canmore, Alberta, Canada [interface FTP à l'archive FTP] − ftp://ftp.ca.linuxfromscratch.org
- Canmore, Alberta, Canada [inteface HTTP à l'archive FTP] − http://ftp.ca.linuxfromscratch.org
- Mainz, Allemagne, Europe [FTP interface to FTP archive] − ftp://ftp.linux−provider.net/pub/lfs/
-

Mainz, Allemagne, Europe [interface HTTP à l'archive FTP] – http://ftp.linux–provider.net/lfs/

- Australie (accessible seulement depuis la Nouvelle–Zélande et l'Australie) – ftp://mirror.aarnet.edu.au/pub/lfs/

Remerciements

J'aimerais remercier les personnes et organisations suivantes pour leur contribution au projet LinuxFromScratch :

- Bryan Dumm pour avoir fourni le matériel qui fait tourner linuxfromscratch.org, ainsi que http://www.bcpub.com, miroir de lfs.bcpub.com.
- DREAMWVR.COM pour leur parrainage, se traduisant par la mise en place de ressources dédiées à LFS et à des projets s'y rattachant.
- Jan Niemann pour avoir fourni http://helga.lk.etc.tu−bs.de comme miroir de 134.169.139.209.
- Johan Lenglet pour avoir initié le projet de traduction en Français à http://www.fr.linuxfromscratch.org.
- Michael Peters pour s'être occupé des modifications relatives au PowerPC d'Apple.
- VA Linux Systems qui, par le biais de Linux.com, ont donné une station de travail VA Linux 420 (anciennement StartX SP2) au projet.
- Jesse Tie Ten Quee qui a donné un graveur Yamaha 8824E CD−RW.
- Jesse Tie Ten Quee pour avoir fourni quasar.highos.com comme miroir de www.ca.linuxfromscratch.org.
- O'Reilly pour avoir donné des livres sur SQL et PHP.
- Robert Briggs pour avoir donné les noms de domaine linuxfromscratch.org et linuxfromscratch.com.
- Torsten Westermann pour faire tourner les miroirs http et ftp à lfs.linux−provider.net.
- D'innombrables contributeurs qui permettent à ce livre d'exister, en faisant suggestions, tests et autres rapports de bugs via la mailing list.

Liste des changements

Si, par exemple, un changement est indiqué pour le chapitre 5, cela signifie (en général) que le même changement a été fait pour les autres architectures.

2.4.4 − Le 23 Janvier 2001

- Chapitre 1: ajout de lfs−security à la liste des listes de discussion disponibles.
- Chapitre 1: mise à jour de la liste des sites miroirs.
- Chapitre 5: Bash avait encore l'option −−with−ncurses qui est fausse (il aurait été équivalent de dire −−with−foo−bar). Cela a été changé en −−with−curses (comme cela avait été déjà fait dans le chapitre 6).
- Chapitre 5: au lieu de CPPFLAGS=−Dvar=value ./configure pendant l'installation de diffutils, grep et sed, nous utilisons maintenant export CPPFLAGS=−Dvar=value && ./configure && unset CPPFLAGS. Cela corrige des problèmes sur certains systèmes pour lesquels la précédente construction ne fonctionnait pas bien.
- Chapitres 5 et 6: ajout du paramètre −−libexecdir à la commande configure de fileutils. Cela évite la création du répertoire $LFS/usr/libexec.
- Chapitres 5 et 6: ajout du paramètre −−libexecdir à la commande configure de tar. Cela évite la création du répertoire $LFS/usr/libexec.
- Chapitre 6: déplacement en toute première position de l'installation des pages man. Ainsi, nous n'avons pas à nous soucier des fichiers qui seraient écrasés par la commande make install de ce package. Cela installera toutes les pages man dont il dispose, et quand nous installerons les packages au chapitre 6, ils installeront leurs propres pages man en remplacement de celles de man−pages.
- Chapitre 6: ajout de la copie des pages man après l'installation de console−tools.
- Chapitre 6: mise à disposition d'un patch pour sysvinit. Lisez les notes d'installation pour savoir à quoi sert ce patch.
- Chapitre 6: suppression des optimisations de compilation du livre. Thomas "Balu" Walter a transformé cela en un LFS−Hint.
-

Chapitre 6: suppression de l'exécution de localedef. Apparemment, cela n'est pas nécessaire.

- Chapitre 6: ajout des liens symboliques compress et uncompress lors de l'installation de gzip.

- Chapitre 6: au moment d'entrer dans l'environnement chroot, utilisation de chemins absolus pour les programme env et bash au lieu de compter sur une initialisation correcte de $PATH.

- Chapitre 6: forçage de la variable libexecdir lors de l'installation de findutils. Comme le script configure de findutils ne reconnaît pas le paramètre libexecdir, nous forcerons la variable lors de l'étape make install.

- Chapitre 6: au lieu de faire un sed sur le fichier Makefile lors de l'installation de procinfo, procps et psmisc, nous redirigeons la sortie de sed vers make et construisons les packages de cette manière, qui est plus efficace.

- Chapitre 6: utilisation de sed pour modifier le fichier MCONFIG.

- Chapitre 6: au lieu d'utiliser cp −avi pour copier les fichier du package man−pages, nous utilisons maintenant cp −dRiv. C'est presque identique à −avi, sauf que cela ne conserve pas les attributs des fichiers. Autrement, les fichiers du package seraient installés avec comme propriétaire l'UID 1000 au lieu de root, ce qui n'est pas une bonne chose.

- Chapitre 6: mention de la section éditeurs des LFS−Hints qui contient des alternatives à vim, dans le cas où vous ne voudriez pas l'installer sur votre système.

- Chapitre 6: ajout du patch pour sysklogd−1.4. Le sysklogd extrait du package est fourni avec un klogd erroné −− incapable d'intercepter les messages du noyau. Le patch résoud ce problème.

- Chapitre 6: au lieu d'avoir deux lignes d'échec séparées dans le fichier inittab (f1:0:... et f2:6:...), elles sont regroupées sur une seule ligne (ft:06:respawn:/sbin/sulogin).

- Chapitre 7: ajout des commentaires aux scripts de démarrage.

- Chapitre 7: modification de la fonction startup dans le script rc. Il n'est pas nécessaire de faire une distinction entre les fichiers qui ont l'extension .sh et ceux qui ne l'ont pas. La commande stty onclr a également été supprimée. Celle−ci ne semble plus nécessaire.

- Chapitre 7: quand un processus est tué en utilisant la fonction killproc dans le script functions, attendre deux secondes avant de continuer pour permettre au kill de se terminer (cela prend parfois

un peu de temps avant que tous les processus ne se terminent).

- Chapitre 7: quand la fonction print_status du script functions est appelée sans paramètre, éviter l'arrêt complet du script appelant, mais retourner simplement un code d'erreur de 1. Cette fonction n'est pas essentielle, elle n'aura donc pas de conséquences si elle ne s'exécute pas correctement.

- Chapitre 7: regroupement du script umountfs avec mountfs.

- Chapitre7 : correction d'un bug mineur dans la fonction status_proc du script functions. On lit "$i ne tourne pas" −− ce devrait être "$1 ne tourne pas".

- Chapitre7 : la fonction print_error_msg dans le script rc demande maintenant à l'utilisateur d'appuyer sur une touche avant de continuer. Ainsi, l'utilisateur peut noter certaines informations avant que tout soit susceptible de disparaître.

- Chapitres 7 et 9: passage de deux à trois chiffres pour les liens symboliques des scripts de démarrage. Cela permet d'ajouter plus facilement de nouveaux scripts de démarrage avant ou après les autres scripts.

- Chapitre 9: découpage de la page des scripts de démarrage du réseau en plusieurs pages, de la même manière que sont présentés les scripts de démarrage dans le chapitre 7.

- Chapitre 9: ajout d'un test de GATEWAY dans le script ethnet. Si la variable GATEWAY n'est pas initialisée, la passerelle par défaut sera mise en place.

- Chapitre 9: ajout de l'option restart aux scripts localnet et ethnet.

- Chapitre 9: suppression du script K ethnet à l'arrêt ou au redémarrage. Les programmes halt et reboot sont appelés avec le paramètre −i qui stoppe toutes les interfaces réseau juste avant l'arrêt ou le redémarrage.

- Chapitre 9: suppression de −−prefix=/usr pour netkit−base. Cela ne sert à rien.

- Annexe A: ajout d'une description pour blockdev dans le package util−linux.

- Annexe C: mise à jour du lien vers le site officiel de téléchargement de util−linux.

- Annexe C: mise à jour du lien vers le site officiel de téléchargement de man−pages.

2.4.3 – 21 novembre 2000

- L'archive LFS FTP a été déplacée sur un nouveau serveur qui est accessible sous le nom packages.linuxfromscratch.org. La raison de ce déplacement est que ce nouveau serveur dispose de plus de bande passante.

- Au lieu de lui faire créer les fichiers en utilisant vim ou un autre éditeur, le lecteur peut maintenant simplement copier et coller une commande qui crée le fichier, sous la forme "cat > fichier_de_sortie << EOF" suivie du texte à insérer dans le fichier, et quand une ligne contenant EOF est lue par cat, il s'arrête de lire et écrit le fichier (sans inclure le EOF). Ce sera plus pratique pour installer les scripts, pour que vous puissiez rendre l'installation de LFS entièrement automatique.

- Ajout d'explications sur les commandes exécutées, pour rendre plus clair ce qui est fait et pour quelle raison, lors de l'installation des packages.

- Chapitre 1: mise à jour de la liste des miroirs HTTP et ajout de la liste des miroirs FTP. Cette liste est à jour au 14 novembre 2000.

- Chapitre 5: lors de l'installation de Bash, changement de l'option de configure −−with−curses en −−with−ncurses. Cela semble corriger la compilation de bash sur certaines distributions qui n'installent pas correctement ncurses.

- Chapitre 5: au lieu de faire remplacer par l'utilisateur, lors de l'installation de gcc, <host> par ce qui apparaît dans $LFS/usr/lib/gcc−lib, vous pouvez utiliser *. Cela ne créera aucun problème car * sera remplacé par un seul répertoire, et ln n'aura donc aucun problème avec lui. Cela permet une automatisation plus facile.

- Chapitre 6: mention du paramètre −e dans le script Configure de perl, ce qui lui évite de vous poser des questions après la création du script config.sh.

- Chapitre 6: suppression de la création du lien symbolique /usr/bin/install −− ce lien a déjà été créé plus tôt, au chapitre 5.

- Chapitre 6: ajout de la création de /var/log/lastlog, où utmp, btmp et wtmp sont créés.

- Chapitre 6: quand le script yacc est créé dans la section Bison, exécution d'un chmod 755 sur celui−ci afin de pouvoir exécuter le script.

- Chapitre 6: changements esthétiques dans le fichier inittab. Au lieu d'utiliser /dev/tty[1−6] comme paramètres de agetty, nous utilisons maintenant simplement dev[1−6]. Cela génère une sortie plus

jolie avec les commandes comme w.

- Chapitre 7: modification de tous les scripts pour utiliser des chemins absolus au lieu de compter sur l'initialisation de $PATH.

- Chapitre 7: dans fstab, remplacement de "none /proc proc defaults 0 0" par "proc /proc proc defaults 0 0". Sur des problèmes de montage, vous pourrez obtenir "none: device or resource busy" au lieu de "proc: device or resourced busy".

- Annexe C: correction de quelques liens erronés.

2.4.2 – 11 octobre 2000

- Chapitre 3: des versions plus récentes étaient mentionnées, mais les liens pointaient toujours vers les versions anciennes. En outre, j'avais oublié de mettre les nouvelles versions des packages dans l'archive ftp. Les deux problèmes sont maintenant corrigés.

- Chapitre 5: au lieu de regarder le nom des fichiers de la bibliothèque C pour déterminer quelle bibliothèque C votre système Linux de base utilise, nous obtenons ceci en exécutant "strings /lib/libc* | grep "release version"" à la place.

- Chapitres 5 et 6: le système de fichiers proc doit être monté dans le chapitre 5 avant d'entrer dans l'environnement chroot, car le programme mount ne sera pas encore disponible après avoir exécuté chroot.

- Chapitre 6: correction d'un bug HTML dans l'installation de GCC qui provoquait l'apparition du caractère CR dans certains navigateurs.

2.4.1 – 10 octobre 2000

- Suppression des prompts bash des commandes. Cela permettra plus facilement de copier & coller les commandes du livre vers la ligne de commande. Taper toutes ces commandes est une bonne chose les premières fois, mais cela devient vite fastidieux. Vous pouvez bien sûr utiliser des scripts pour le faire, mais ce n'est pas l'objectif de ce livre. C'est le but d'un autre projet (alfs.linuxfromscratch.org).

- Interversion des chapitres 8 et 9. Maintenant, nous commençons par redémarrer d'abord, puis nous installons le réseau. Précédemment, les programmes réseau ne fonctionnaient pas à moins que le système normal et LFS ne tournent tous deux avec la même version du noyau, ce qui n'est souvent pas le cas. Intervertir les chapitres élimine cet éventuel problème.

-

Chapitre 3: tous les packages ont été déplacés sur download.linuxfromscratch.org et les liens mis à jour en conséquence. Les sites officiels de téléchargement pour tous les packages sont listés dans l'appendice C.

- Chapitres 5 et 6: déplacement de l'exécution de localedef, qui était après l'installation de Glibc dans le chapitre 5, après l'exécution de chroot au chapitre 6. C'était une erreur (le seul vrai bug dans la version 2.4) de le faire dans le chapitre 5.

- Chapitre 6: installation de Vim en premier lieu. Ainsi, si vous avez besoin d'éditer quelque chose, vous avez maintenant un éditeur disponible. Cela entraîne aussi le déplacement de deux autres packages pour satisfaire les dépendances.

- Chapitre 6: quand nous utilisons sed pour modifier un fichier Makefile, nous exécutons maintenant "make −f Makefile2" au lieu de "mv Makefile2 Makefile && make".

- Chapitre 6: ajout de la ligne "publickey: files" dans le fichier nsswitch.conf. Ceci est nécessaire quand vous utilisez un noyau 2.4 pour vous connecter correctement.

- Chapitre 6: ajout du script /usr/bin/yacc qui exécute bison avec l'option −y qui émule les conventions de nom de fichier de sortie de yacc. Cela est dû au fait qu'il y a quelques packages qui comptent sur yacc et ne peuvent pas (encore) fonctionner avec bison.

- Chapitre 6: modification du script /usr/sbin/makewhatis après l'installation du package man. Le script /usr/sbin/makewhatis nécessite de positionner la variable AWK= à /usr/bin/mawk.

- Chapitre 7: ajout du modèle de script. Ainsi, vous pouvez facilement ajouter de nouveaux scripts de démarrage sans avoir à les écrire à partir de rien.

2.4 – 28 août 2000

- Découpage du livre en deux ouvrages différents, l'un pour Intel, l'autre pour PPC.

- Chapitre 4: ajout des répertoires mail et dev/pts dans la section "Création des répertoires".

- Chapitre 5: tout ce qui est fait depuis chroot et après a été déplacé dans un nouveau chapitre.

- Chapitre 6: déplacement de la partie optimisation juste avant d'entrer dans l'environnement chroot. C'est un gaspillage d'utiliser les optimisations du compilateur pour les packages compilés statiquement puisque, de toute manière, ils seront remplacés.

-

Chapitre 6: pour entrer dans l'environnement chroot, nous allons d'abord dans le répertoire $LFS/root. Certaines anciennes versions de chroot ont des problèmes quand votre répertoire de lancement n'est pas dans l'environnement chroot. De plus, nous n'exécutons pas bash directement dans l'environnement chroot, mais nous lançons le programme "env", ce qui nous permet d'entrer dans un nouvel environnement qui n'a que CFLAGS et CXXFLAGS d'initialisés.

- Chapitre 6: quelques personnes ont eu des problèmes pour compiler M4 dans l'environnement chroot. Des instructions sont fournies sur la manière d'installer ce package statiquement pour les utilisateurs concernés.

- Chapitre 6: nous ne pouvons pas déplacer le programme mv avec lui−même durant l'installation dynamique du package fileutils. Nous le copions donc tout d'abord dans /bin, puis nous supprimons /usr/bin/mv.

- Chapitre 5: ajout de "make localedata/install−locales" lors de l'installation de Glibc. Ceci installe les fichiers locaux qu'utilisent plusieurs applications (notamment les applications GDK) si vous avez un système supportant NLS (ce qu'est LFS, mais sans les fichiers locaux, c'est quasiment inutile).

- Chapitre 6: déplacement de l'installation de vim avant Lilo, puisque vous pourriez vouloir éditer le fichier Makefile de Lilo pour ajouter des optimisations de compilateur.

- Chapitre 6: déplacement de l'installation de la suite Masquage des mots de passe après sh−utils. Sinon, sh−utils remplace la version "su" de cette suite par la sienne, ce qui n'est pas souhaitable.

- Chapitre 6: modification de la manière d'entrer dans l'environnement chroot. Nous utilisons la commande "env" pour créer un environnement vierge, afin que les variables d'environnement du système Linux normal n'interfèrent pas avec l'environnement chroot. La seule variable initialisée en entrant dans l'environnement chroot est la variable HOME.

- Chapitre 6: par compatibilité avec la nouvelle manière d'entrer dans chroot, le fichier $LFS/root/.bash_profile a été créé pour initialiser certaines variables comme TERM, CFLAGS, CXXFLAGS et toutes celles que vous jugerez nécessaires.

2.3.7 − 3 août 2000

- Tous les chapitres: suppression des balises SGML <blockquote>, pour que le contenu des fichiers ne soit plus indenté. Ceci permet de copier−coller facilement du livre dans vos fichiers sans avoir besoin de reformater manuellement les fichiers pour se débarrasser des indentations.

- Chapitre 4: ajout de /var/tmp à la commande "chmod 1777 tmp usr/tmp".

-

Chapitre 4: les commandes mkdir ont été rendues moins répétitives en utilisant une boucle for−loop pour la création des répertoires dans $LFS/usr et $LFS/usr/local.

- Chapitre 5: déplacement de la commande chmod 754 pour MAKEDEV après l'exécution de sed.
- Chapitre 5: changement de l'ordre d'installation des packages, plus proche d'un ordre alphabétique.
- Chapitre 5: après l'installation de console−tools, le fichier /usr/share/defkeymap.kmap.gz est créé; il sera utilisé par le script loadkeys.
- Chapitre 5: suppression de "gcc −c watch.c" de "Installation de Procps". Merci de nous faire savoir si cela est encore nécessaire sur certains matériels.
- Chapitre 5: ajout du lien symbolique /usr/bin/install, car il semble qu'au moins un package (sysklogd) a le chemin de install codé en dur dans son fichier Makefile.
- Chapitre 5: après l'installation de gettext, nous avons un fichier /po−mode.el. Ce fichier sera déplacé dans /usr/share/gettext, où il a probablement sa place.
- Chapitre 5: au lieu de passer l'option −−with−root−prefix=/ au script configure de e2fsprogs, nous passons maintenant −−with−root−prefix.
- Chapitre 5: quand gzip est installé et que les fichiers ont été déplacés dans /bin, le lien en dur entre les fichiers est supprimé. Nous déplaçons donc simplement gzip dans /bin, et créons un lien symbolique entre gzip et gunzip.
- Chapitre 5: dans l'environnement chroot, changement de l'ordre d'installation de quelques packages dont les dépendances ont évolué avec le temps.
- Chapitre 5: le fichier inittab a été légèrement modifié pour mieux supporter le niveau d'exécution mono−utilisateur (single user). Quand vous passez au niveau d'exécution S, s ou 1, il fera maintenant correctement son travail.
- Chapitre 6: modification de la commande loadkeys dans le script loadkeys. La nouvelle commande est loadkeys −d, qui charge le fichier /usr/share/keymaps/defkeymap.kmap.gz.
- Chapitre 6: modification de ". /etc/init.d/functions" en "source /etc/init.d/functions".
-

Chapitre 6: suppression de la commande "rm /fastboot" du script checkfs.

2.3.6 – 19 juillet 2000

- Chapitre 3: ré–ordonnancement de la liste de téléchargement des logiciels, pour qu'elle corresponde de nouveau avec l'ordre dans lequel les packages sont utilisés (le premier package listé est le premier package utilisé dans le livre, le second de la liste sera le second utilisé dans le livre, etc.).
- Chapitre 3: ajout des tailles de fichier des packages téléchargés.
- Chapitre 3: suppression du package start–stop–daemon.
- Chapitre 3: ajout des patchs de findutils et glibc à la liste des packages.
- Chapitre 3: ajout du package man–pages à la liste des packages.
- Chapitre 4: déplacement de la création des fichiers $LFS/dev dans le chapitre 5, après l'entrée dans l'environnement chroot. Ceci est fait parce que les GID du système normal et du système LFS peuvent différer, et que le script MAKEDEV dépend des GID.
- Chapitre 5: ajout de l'installation du package man–pages.
- Chapitre 5: ajout de quelques groupes couramment utilisés dans le fichier /etc/group lors de sa création (ce sont les groupes requis par le script MAKEDEV).
- Chapitre 5: le fichier /proc/devices est copié dans $LFS/proc, au profit du script MAKEDEV. La présence de ce fichier assure la création correcte des fichiers de périphérique.
- Chapitre 5: changement de mise en page. Chaque installation de package a maintenant sa propre page. De plus, le texte de l'appendice A pour chaque package est inclu avec les instructions d'installation, pour que vous puissiez lire à quoi se rapporte un package pendant (ou avant, ou après) son installation.
- Chapitre 5: suppression des patchs pour diffutils, grep, gzip et sed, qui réglaient les problèmes de lien statique. Les problèmes sont plutôt corrigés en passant des arguments de compilation au préprocesseur C (cpp).
- Chapitre 5: ajout de l'option ––disable–termcap pour demander le non–support de termcap (si vous voulez savoir pourquoi termcap n'est plus utilisé, lisez le fichier INSTALL livré avec le package

Ncurses).

- Chapitre 5: ajout de quelques fichiers manquants du package fileutils à la commande "mv".
- Chapitre 5: suppression de l'installation du package start−stop−daemon.
- Chapitre 5: suppression des paramètres −e pour les lignes de commande de make.
- Chapitre 5: au lieu d'éditer les fichiers Makefile de procinfo, procps et psmisc avec un éditeur de texte, la commande sed est utilisée.
- Chapitre 6: ajout du script setclock, pour le cas où votre horloge matérielle ne serait pas positionnée à GMT.
- Chapitre 6: suppression de l'utilisation du programme start−stop−daemon, et remplacement par des fonctions appropriées qui utilisent les programmes comme pidof et kill pour réaliser les mêmes tâches, mais avec un meilleur contrôle sur ce qui se passe.
- Chapitre 6: ajout des fonctions loadproc et killproc au fichier /etc/init.d/functions, qui remplacent les fonctions que le programme start−stop−daemon réalisait auparavant.
- Chapitre 6: quand le script checkfs est lancé sans erreurs, il affiche maintenant un OK vert.
- Chapitre 6: quand /fastboot ou /forcefsck existent, ils ne seront pas supprimés par le script checkfs, mais par le script mountfs, dès que la partition racine aura été remontée en mode lecture/écriture.
- Chapitres 6 et 7: au lieu d'exécuter un fichier avec ". /etc/init.d/functions", "source /etc/init.d/functions" est maintenant utilisé. Ceci est plus facile à lire, et est plus clair pour des personnes qui ne connaissent pas bien le langage script.
- Annexe A: suppression de start−stop−daemon.
- Annexe B: suppression de quelques objets non utilisés dans le livre et des parties Howto (les références à Sendmail et ISP−Hookup−HOWTO).

2.3.5 – 19 juin 2000

- Chapitre 3: mise à jour du site de téléchargement de LILO.

Chapitre 3: mise à jour du site de téléchargement de la Suite de Masquage de mots de passe.

- Chapitre 3: mise à jour du site de téléchargement de Flex.
- Chapitre 3: mise à jour du site de téléchargement de File.
- Chapitre 3: ajout de netkit−base et net−tools à la partie des packages obligatoires.
- Chapitre 5: un patch glibc−2.1.3 est disponible si vous avez des problèmes pour compiler glibc sur une machine avec bash−2.04.
- Chapitre 5: ajout de l'optimisation du compilateur.
- Chapitre 5: ajout de la création du mot de passe de root dans "Configurer les logiciels essentiels"
- Chapitre 5: le package Linux86 a été remplacé par le package Bin86.
- Chapitre 5: insertion d'informations sur la manière d'optimiser les compilations.
- Chapitre 5: déplacement de l'installation de Groff et Man avant Perl. Ainsi, Perl sait où et comment installer les pages de man.
- Chapitre 5: modification de l'option local−prefix de GCC en /usr/local au lieu de /usr (c'était un reliquat du temps où /usr/local était un lien symbolique sur /usr).
- Chapitre 5: correction des commandes quand un patch est utilisé et que le nom de fichier du patch contient un suffixe .gz.
- Chapitre 5: ajout de −−disable−nls à chaque commande de configuration dans la partie "Préparer le système LFS...", qui n'en a pas encore besoin.
- Chapitre 5: ajout de l'installation de bash−2.03, pour avoir un shell utilisable pour compiler les packages qui violent les standards POSIX en ce qui concerne les caractères valides dans les noms de variable.
- Chapitre 5: ajout de l'installation de console−tools et console−data pour les personnes qui n'ont pas un clavier US.

Chapitre 5: déplacement du programme ed dans le répertoire /bin, conformément au standard FHS.

- Chapitres 6 et 7: implémentation de LSB, recommandé pour le schéma des niveaux de démarrage.
- Chapitres 6 et 7: implémentation de "scripts de démarrage fantaisistes". Quand une erreur survient dans un script de démarrage, il écrit toujours FAILED, mais le texte est en rouge. Quand quelque chose se passe bien, il écrit toujours OK, mais le texte est en vert.
- Chapitre 6: ajout des scripts loadkeys pour les personnes qui n'ont pas un clavier US.
- Chapitre 6: ajout du répertoire /etc/sysconfig à "Créer les répertoires".
- Chapitre 6: renommage du script de démarrage checkroot en checkfs. Le script vérifie également, maintenant, d'autres systèmes de fichiers.
- Chapitre 6: mise à jour du script de démarrage mountfs pour monter tous les systèmes de fichiers indiqués dans le fichier /etc/fstab, et ne pas positionner l'option noauto.
- Chapitre 6: après la vérification par checkfs de l'existence de /fastboot ou /forcecheck, il supprimera ces fichiers.
- Chapitres 6 et 7: modification des droits des scripts de démarrage de 755 en 754.
- Chapitre 7: déplacement des informations spécifiques au système pour la configuration de hostname et ethernet dans le fichier /etc/sysconfig/network.
- Chapitre 7: suppression de la commande de passerelle par défaut.
- Chapitre 7: correction typographique dans le script ethnet (NETMAKSK −> NETMASK).
- Chapitre 7: un patch net−tools est disponible pour corriger un bogue mineur dans le package (noms de variable illégaux sur lesquels bash−2.04 se plaint).

2.3.4 − 5 juin 2000

- Chapitre 5: correction de la configuration des fichiers d'en−tête du noyau.
-

Chapitre 5: correction de la configuration de lilo.

2.3.3 – 15 mai 2000

- Modification du point de montage par défaut de /mnt/xxx en /mnt/lfs (où xxx était le nom de la partition, comme hda5, sda5, etc.). La raison de ce changement est de faciliter les instructions multi−plateformes.
- Chapitre 4: modification des droits par défaut des répertoires $LFS/root et $LFS/tmp en respectivement 0750 et 1777.
- Chapitre 5: suppression du mot de passe codé du fichier passwd. A la place, un fichier sans mot de passe positionné est créé. Le mot de passe de root peut être positionné par l'utilisateur quand le système est redémarré sous LFS (après le chapitre 8).
- Chapitre 5: correction de la commande de compilation de procps pour watch.c. Cela devrait compiler sans problème maintenant.
- Chapitre 5: correction de l'installation du patch gzip (utilisation d'un mauvais nom de fichier dans la commande patch).
- Chapitre 5: modification de l'entrée dans l'environnement chroot pour positionner bash comme shell de connexion.
- Chapitre 5: la configuration du noyau a été déplacée dans ce chapitre, parce qu'elle doit être faite avant la compilation de programmes comme e2fsprogs et lilo.
- Chapitre 6: correction du script rc. Il vérifie maintenant si le niveau d'exécution précédent a lancé un service avant d'essayer de l'arrêter dans le nouveau niveau d'exécution. De plus, si un service est déjà démarré dans le niveau d'exécution précédent, il ne tentera pas de le démarrer de nouveau dans le nouveau niveau d'exécution. Merci à Jason Pearce pour avoir apporté ce script corrigé.
- Chapitre 7: correction du script ethnet −− suppression des parenthèses des variables d'environnement et de la commande pour ajouter un acheminement. La commande ifconfig utilisée pour lancer le périphérique eth positionne déjà ce routage.

2.3.2 – 18 avril 2000

- Chapitre 4.7: changement du propriétaire des fichiers $LFS/dev/*.
-

Correction de nombreuses erreurs typographiques survenues durant le passage de LinuxDoc DTD (2.2 et inférieurs) en DocBook DTD (2.3.1 et plus).

- Léger déplacement des chapitres et application d'une nouvelle structuration du livre. Les installations pour Intel, Apple PowerPC et d'autres systèmes à venir seront mises dans la partie du livre qui leur est dédiée.

- Après la préparation du système pour installer les logiciels système de base, nous ne redémarrons plus le système; à la place, nous mettons en place un environnement chroot. Ceci aura le même effet, sans avoir à redémarrer.

- Le PowerPC d'Apple a maintenant ses propres chapitres dédiés. Ceci devrait notablement améliorer la lisibilité.

- Tous les chapitres optionnels ont été supprimés. LFS suit une philosophie "nous fournissons les fondations, à vous de construire le reste de la maison".

- Remplacement des packages corrigés par des fichiers patch. Ainsi, vous pourrez voir ce qui doit être modifié dans un package pour qu'il compile correctement.

2.3.1 – 12 avril 2000

- Chapitre 4.4: ajout du lien symbolique *LFS/usr/info qui pointe sur $LFS/usr/share/info.

- Chapitre 7.3.1: ajout d'une seconde variante à "swap–line" dans un fichier fstab.

- Chapitre 7.3.2: suppression de $LFS des commandes.

- Chapitre 7.4.43: ajout du lien symbolique vi.

- Chapitre 9.2.5: amélioration du script ethnet pour inclure les informations de routage.

- Chapitre 10.1.2: correction du sous–répertoire "mqueue" manquant dans mkdir /var/spool –> /mkdir /var/spool/mqueue.

- Chapitre 10.1.4: mise à jour du fichier de configuration de sendmail avec quelques options nécessaires.

-

Chapitre 10.1.7: correction du chemin erroné du répertoire /etc/init.d/rc2.d –> /etc/rc2.d.

Listes de diffusion et archives

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

- lfs−discuss
- lfs−apps
- lfs−announce
- lfs−security
- alfs−discuss
- alfs−docs
- alfs−ipc
- alfs−profile

lfs−discuss

La liste de diffusion lfs−discuss discute exclusivement de ce qui concerne le LFS−BOOK. Si vous avez des problèmes avec le livre, si vous voulez signaler un bug ou deux ou si vous avez des suggestions pour améliorer le livre, utilisez cette liste de diffusion.

Tout autre courrier électronique doit être posté sur la liste lfs−apps.

lfs−apps

La liste lfs−apps s'occupe de tout ce qui n'entre pas dans le domaine de la liste lfs−discuss.

lfs−announce

La liste lfs−announce est une liste modérée. Vous pouvez vous y inscrire, mais vous ne pourrez y poster aucun message. Cette liste est utilisée pour annoncer la mise à disposition des nouvelles versions stables. Si vous souhaitez également être informé au sujet des versions de développement, vous devrez rejoindre la liste

lfs−discuss. Si vous êtes déjà sur cette liste, s'inscrire sur lfs−announce ne présente que peu d'intérêt car tout ce qui y est posté le sera également sur lfs−discuss.

lfs−security

La liste de diffusion lfs−security s'occupe des questions de sécurité. Si vous avez des inquiétudes concernant la sécurité ou si vous avez entendu dire qu'un package utilisé par LFS a posé des problèmes de sécurité, vous pouvez en faire part sur cette liste.

alfs−discuss

La liste alfs−discuss traite du développement d'ALFS (Automated LinuxFromScratch). Le but de ce projet est de développer un outil d'installation capable d'installer automatiquement un système LFS à votre place. Son objectif principal est d'accélérer la compilation en vous évitant d'entrer manuellement les commandes nécessaires pour configurer, compiler et installer les packages.

alfs−docs

ALFS−docs est le projet de documentation ALFS qui crée et maintient tout ce qui concerne la documentation ALFS.

alfs−ipc

La liste alfs−ipc traite de ce qui concerne la communication interprocessus de ALFS.

alfs−profile

La liste alfs−profile traite du développement du profil et de la DTD XML ALFS.

Comment s'inscrire ?

Vous pouvez vous inscrire à n'importe laquelle des listes de diffusion mentionnées plus haut en envoyant un email à listar@linuxfromscratch.org et en écrivant *subscribe nom−de−la−liste* comme sujet du message.

Vous pouvez, si vous le souhaitez, vous inscrire à de multiples listes en même temps en n'envoyant qu'un seul email. Dans ce cas, écrivez ce que vous voulez dans le sujet, du moment qu'il ne s'agit pas d'une commande valide, "hello" par exemple. Puis écrivez les commandes d'inscription dans le corps du message. L'email ressemblera à :

To: listar@linuxfromscratch.org

Subject: hello

subscribe lfs−discuss
subscribe lfs−apps
subscribe alfs−discuss

Une fois que vous avez envoyé l'email, le programme Listar vous enverra un email demandant confirmation de votre demande d'inscription. Une fois que vous aurez envoyé votre email de confirmation, Majordomo vous enverra de nouveau un email indiquant la (les) liste(s) sur la(les)quelle(s) vous vous êtes inscrits. Cet email contiendra également un message d'introduction concernant la(les)−dite(s) liste(s).

Comment se désabonner ?

Vous pouvez vous désabonner d'une liste diffusion en envoyant un email à listar@linuxfromscratch.org et en écrivant *unsubscribe nom−de−la−liste* comme sujet du message.

Vous pouvez, si vous le souhaitez, vous désabonner de plusieurs listes en même temps en n'envoyant qu'un seul email. Dans ce cas, écrivez ce que vous voulez dans le sujet, du moment qu'il ne s'agit pas d'une commande valide, "hello" par exemple. Puis écrivez les commandes de désabonnement dans le corps du message. L'email ressemblera à :

To: listar@linuxfromscratch.org
Subject: hello

unsubscribe lfs−discuss
unsubscribe lfs−apps
unsubscribe alfs−discuss

Une fois que vous avez envoyé l'email, le programme Listar vous renverra un email vous demandant confirmation de votre demande de désabonnement. Après que vous aurez renvoyé cet email de confirmation, Listar vous enverra de nouveau un email vous indiquant que vous avez été désabonné de la / des liste(s) de diffusion.

Archives des listes de diffusion

Chacune des listes de diffusion publiques possède une archive à laquelle vous pouvez accéder pour trouver des informations sur les sujets déjà postés sur cette liste. Vous pouvez les trouver à http://archive.linuxfromscratch.org/mail−archives

---

Points de contact

Envoyez de préférence tous vos emails vers la liste lfs−discuss

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

Chapitre 2. Information Importante

A propos de $LFS

S'il vous plaît, lisez ce qui suit attentivement : tout au long de ce livre vous verrez fréquemment le nom de variable $LFS. $LFS doit à chaque fois être remplacé par le répertoire de la partition montée qui contient le système LFS. Comment créer et où monter cette partition sera expliqué en détail plus loin au chapitre 4. Dans mon cas, la partition LFS est montée sur /mnt/lfs. Si je lis moi−même ce livre, et si je vois $LFS quelque part, je ferai semblant d'avoir lu /mnt/lfs. Si je lis que je dois exécuter cette commande : cp inittab $LFS/etc, j'exécuterai en réalité ceci : cp inittab /mnt/lfs/etc

Il est important que vous fassiez ceci quel que soit l'endroit où vous le lisez ; que ce soit dans une commande à entrer à l'invite, ou dans un fichier à éditer ou à créer.

Si vous le voulez, vous pouvez créer la variable d'environnement LFS. De cette façon, vous pourrez taper $LFS littéralement, au lieu de la remplacer par quelque chose comme /mnt/lfs. Ceci est obtenu en exécutant : export LFS=/mnt/lfs

Si je lis cp inittab $LFS/etc, je tape littéralement cp inittab $LFS/etc et le shell remplace automatiquement cette commande par cp inittab /mnt/lfs/etc.

N'oubliez pas d'initialiser la variable LFS à chaque fois. Si vous n'avez pas initialisé la variable et si vous l'utilisez dans une commande, $LFS sera ignorée, et ce qui reste sera exécuté. Si la variable $LFS n'est pas initialisée, la commande cp inittab $LFS/etc copiera le fichier inittab dans le répertoire /etc, ce qui écrasera le inittab de votre système. Un fichier comme inittab n'est pas un gros problème parce qu'il peut facilement être restauré, mais si vous faites cette erreur pendant l'installation de la librairie C, vous pouvez endommager bien des choses.

Une des façons d'être sûr que la variable $LFS est toujours initialisée est de l'ajouter dans votre fichier /root/.bash_profile et/ou dans votre fichier /root/.bashrc. De cette manière, chaque fois que vous utilisez 'su' pour devenir root et installer LFS, la variable $LFS est positionnée pour vous.

Comment télécharger le logiciel

Tout au long de ce document, je supposerais que vous avez rangés tous les paquetages que vous avez téléchargés quelque part dans $LFS/usr/src.

La convention que j'utilise est d'avoir un répertoire $LFS/usr/src/sources. Sous sources, on trouvera le répertoire 0−9 et les répertoires a à z. Un paquetage tel sysvinit−2.78.tar.gz sera rangé sous LFS/usr/src/sources/s/. Un paquetage tel bash−2.04.tar.gz sera rangé sous LFS/usr/src/sources/b/, et ainsi de suite. Vous n'avez pas à suivre cette convention, bien sur, je ne faisais que vous donner un exemple. Il est préférable de laisser les paquetages hors de $LFS/usr/src et de les déplacer dans un sous−répertoire, de façon à garder un répertoire $LFS/usr/src, dans lequel on pourra désarchiver les paquetages et travailler avec.

Le chapitre suivant contient la liste de tous les paquetages que vous devez télécharger, mais la partition qui va contenir notre système LFS n'est pas encore créée. Donc, stockez temporairement les fichiers là où vous le voulez, et rappelez−vous de les recopier dans $LFS/usr/src/ quand vous aurez fini le chapitre où vous préparerez une nouvelle partition (chapitre 4).

Comment installer le logiciel

Avant que vous ne puissiez commencer à faire quelque chose avec un paquetage, vous devez d'abord le désarchiver. Souvent, vous trouverez les paquetages archivés par tar puis compressés par gzip (vous pouvez le déterminer en regardant l'extension du fichier. Les fichiers archivés par tar puis compressés par gzip ont l'extension .tar.gz ou .tgz, par exemple). Je ne rappellerais pas à chaque fois comment décompresser et désarchiver une archive. Je vais vous le dire une seule fois, dans ce paragraphe. Il y a aussi parfois la possibilité de télécharger un fichier .tar.bz2. un tel fichier est archivé par tar puis compressé par le programme bzip2. Bzip2 atteint une meilleure compression que le plus communément utilisé gzip. pour utiliser des archives bz2, vous devez avoir installé le programme bzip2. la plupart des distributions, si ce n'est toutes, sont fournies avec ce programme, donc il est fort probable qu'il soit déjà installé sur votre système. Dans le cas contraire, installez le en utilisant les outils d'installation de votre distribution.

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

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

Quand vous avez un fichier archivé par tar puis compressé par gzip, vous le désarchivez en exécutant une des deux commandes suivantes, selon le format du nom du fichier:

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

Quand vous avez un fichier archivé par tar puis compressé par bzip2, vous le désarchivez en exécutant:

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

Certains programmes tar (la plupart, mais pas tous, de nos jours) sont légèrement modifiés pour être capable d'utiliser des fichiers bzip2 directement en utilisant soit le paramètre I soit le paramètre y, qui fonctionne de la même façon que le paramètre z pour les archives compressées par gzip.

Quand vous avez un fichier archivé par tar, vous le désarchivez en éxécutant:

```
tar xvf filename.tar
```

Une fois l'archive désarchivée, un nouveau répertoire est créé sous le répertoire courant (ce document suppose que vous désarchivez les archives sous le répertoire $LFS/usr/src). Vous devez vous placer dans ce nouveau répertoire avant de continuer à suivre les instructions d'installation. Donc, à chaque fois que le livre est sur le point d'installer un programme, c'est à vous de désarchiver l'archive de ses sources.

Une fois que vous avez installé un paquetage, vous pouvez faire deux choses avec. Vous pouvez, soit supprimer le répertoire qui contient les sources, soit le garder. je ne vois pas d'objections si vous décidez de le

garder. Mais si vous avez de nouveau besoin du même paquetage dans un chapitre ultérieur, vous devrez supprimer le répertoire avant de le réutiliser. Si vous ne faites pas cela, vous pourriez avoir des problèmes car les anciens réglages seraient réutilisés (réglages qui s'appliquent à votre système Linux normal mais pas toujours à votre système LFS). Faire simplement un make clean ou make distclean ne garantit pas forcément un nettoyage complet de l'arborescence des sources. le script de configuration peut aussi comporter des fichiers dans des sous−répertoires variés qui ne sont pas toujours supprimés par l'exécution d'un make clean.

Il y a une exception à cette règle : ne retirez pas l'arborescence des sources du noyau linux. De nombreux programmes ont besoin des en−têtes du noyau, c'est donc le seul répertoire que vous ne devez pas supprimer, si vous avez encore l'intention de compiler des logiciels.

Télécharger les scripts de démarrage

Taper tous les scripts de démarrage des chapitres 7 et 9 peut être une tâche longue et fastidieuse, sans compter qu'elle est facilement source d'erreurs.

Afin de vous faire gagner un peu de temps, vous pouvez télécharger ces scripts aux adresses suivantes: http://download.linuxfromscratch.org/bootscripts/ ou ftp://download.linuxfromscratch.org/bootscripts/.

Télécharger les commandes LFS

"LFS Commands" est une archive tar contenant les fichiers qui décrivent les commandes d'installation pour les packages utilisés dans ce livre. Le contenu de ces fichiers peut être directement copié vers votre shell afin d'installer les packages, sauf pour quelques uns nécessitant une ou deux modifications (par exemple, lorsque vous installez le package "console−tools", vous devez choisir la disposition des touches du clavier, choix qui ne peut être fait à votre place).

Ces fichiers peuvent également être utilisés pour voir en un coup d'oeil quelles sont les commandes qui ont changé entre deux versions de LFS. Vous pouvez télécharger l'archive tar "LFS Commands" pour la version actuelle et pour la précédente, et lancer un "diff" sur les fichiers. Ainsi vous pouvez voir quels packages ont vu leurs instructions d'installation changées, et modifier en conséquence vos propres scripts, ou bien réinstaller le package si cela vous semble nécessaire.

Les "LFS Commands" peuvent être téléchargées de:http://download.linuxfromscratch.org/lfs−commands/ ou ftp://download.linuxfromscratch.org/lfs−commands/.

II. Deuxième Partie – Installation du système LFS

Table des matières

3. Paquetages que vous devez télécharger
4. Préparation d'une nouvelle partition
5. Préparation du système LFS
6. Installer les logiciels du système de base
7. Création des scripts de démarrage du système
8. Rendre amorçable le système LFS
9. Configurer un réseau de base

Chapitre 3. Paquetages que vous devez télécharger

Ci−dessous, il y a une liste de tous les paquetages que vous avez besoin de télécharger pour construire le système de base. Les numéros de version affichés correspondent à des versions de logiciel qui sont connues pour fonctionner, et qui sont la base du livre. Si vous rencontrez des problèmes que vous ne pouvez résoudre vous−même, télécharger la version indiquée par le livre (au cas où vous auriez téléchargé une version plus récente).

Si le serveur packages.linuxfromscratch.org n'autorise plus les connexions, essayez un de nos sites miroirs. Les adresses des sites miroirs se trouvent au Chapitre 1 − Version du livre.

Dans Annexe C − Sites officiels de téléchargement, nous avons fourni la liste des sites officiels de téléchargement des paquetages listés ci−dessous. L'archive LFS FTP ne contient que les versions des paquetages recommandés par ce livre. Si vous cherchez des versions plus récentes que celles listées ici, vous pouvez jeter un oeil à l'annexe C.

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, vous êtes priés de lire Chapitre 2 − Comment installer le logiciel.

La liste ci−dessous a été mise à jour le 5 janvier 2001.

- Browse FTP: ftp://packages.linuxfromscratch.org
- Bash (2.04) − 1,307 KB:
  ftp://packages.linuxfromscratch.org/pub/common−packages/bash−2.04.tar.bz2
- Binutils (2.10.1) 5,523 KB:
  ftp://packages.linuxfromscratch.org/pub/common−packages/binutils−2.10.1.tar.bz2
- Bzip2 (1.0.1) 454 KB:
  ftp://packages.linuxfromscratch.org/pub/common−packages/bzip2−1.0.1.tar.gz
- Diff Utils (2.7) 247 KB:
  ftp://packages.linuxfromscratch.org/pub/common−packages/diffutils−2.7.tar.bz2
- File Utils (4.0) 801 KB:
  ftp://packages.linuxfromscratch.org/pub/common−packages/fileutils−4.0.tar.bz2
- GCC (2.95.2) 9,555 KB:
  ftp://packages.linuxfromscratch.org/pub/common−packages/gcc−2.95.2.tar.bz2
-

Linux Kernel (2.2.18) 14,921 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/linux−2.2.18.tar.bz2

- Glibc (2.1.3) 6,308 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/glibc−2.1.3.tar.bz2

- Glibc−crypt (2.1) 36 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/glibc−crypt−2.1.tar.bz2

- Glibc−linuxthreads (2.1.3) 112 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/glibc−linuxthreads−2.1.3.tar.bz2

- Glibc Patch (2.1.3) − 1 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/glibc−2.1.3.patch.gz

- Grep (2.4.2) 382 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/grep−2.4.2.tar.bz2

- Gzip (1.2.4a) 178 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/gzip−1.2.4a.tar.bz2

- Gzip Patch (1.2.4a) 1 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/gzip−1.2.4a.patch.gz

- Make (3.79.1) 749 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/make−3.79.1.tar.bz2

- Sed (3.02) 221 KB: ftp://packages.linuxfromscratch.org/pub/common−packages/sed−3.02.tar.bz2

- Sh−utils (2.0) 824 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/sh−utils−2.0.tar.bz2

- Tar (1.13) 730 KB: ftp://packages.linuxfromscratch.org/pub/common−packages/tar−1.13.tar.bz2

- Tar Patch (1.13) 2 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/gnutarpatch.txt

-

Text Utils (2.0) 1,040 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/textutils−2.0.tar.bz2

- MAKEDEV (2.5) − 11 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/MAKEDEV−2.5.tar.bz2
- Bison (1.28) 321 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/bison−1.28.tar.bz2
- Mawk (1.3.3) 168 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/mawk1.3.3.tar.bz2
- Patch (2.5.4) 149 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/patch−2.5.4.tar.bz2
- Find Utils (4.1) 226 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/findutils−4.1.tar.bz2
- Find Utils Patch (4.1) 1 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/findutils−4.1.patch.gz
- Ncurses (5.2) 1,307 KB:
ftp://packages.linuxfromscratch.org/pub/intel−packages/ncurses−5.2.tar.bz2
- Less (358) 178 KB: ftp://packages.linuxfromscratch.org/pub/common−packages/less−358.tar.bz2
- Groff (1.16.1) 1,173 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/groff−1.16.1.tar.bz2
- Man (1.5h1) 156 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/man−1.5h1.tar.bz2
- Perl (5.6.0) 4,327 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/perl−5.6.0.tar.bz2
- M4 (1.4) 249 KB: ftp://packages.linuxfromscratch.org/pub/common−packages/m4−1.4.tar.bz2
-

Texinfo (4.0) 812 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/texinfo−4.0.tar.bz2

- Autoconf (2.13) 333 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/autoconf−2.13.tar.bz2
- Automake (1.4) 277 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/automake−1.4.tar.bz2
- Flex (2.5.4a) 278 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/flex−2.5.4a.tar.bz2
- File (3.33) 126 KB: ftp://packages.linuxfromscratch.org/pub/common−packages/file−3.33.tar.bz2
- Libtool (1.3.5) 361 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/libtool−1.3.5.tar.bz2
- Bin86 (0.15.4) 111 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/bin86−0.15.4.tar.bz2
- Gettext (0.10.35) 525 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/gettext−0.10.35.tar.bz2
- Console−tools (0.2.3) 490 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/console−tools−0.2.3.tar.bz2
- Console−tools (0.2.3) Patch: 4 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/console−tools−0.2.3.patch.gz
- Console−data (1999.08.29) 418 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/console−data−1999.08.29.tar.bz2
- E2fsprogs (1.19) 808 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/e2fsprogs−1.19.tar.bz2
- Ed (0.2) 158 KB: ftp://packages.linuxfromscratch.org/pub/common−packages/ed−0.2.tar.bz2
-

Ld.so (1.9.9) 280 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/ld.so−1.9.9.tar.bz2

- Lilo (21.6) 172 KB: ftp://packages.linuxfromscratch.org/pub/intel−packages/lilo−21.6.tar.bz2
- Modutils (2.4.0) 195 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/modutils−2.4.0.tar.bz2
- Vim−rt (5.7) 905 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/vim−5.7−rt.tar.bz2
- Vim−src (5.7) 963 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/vim−5.7−src.tar.bz2
- Procinfo (17) 21 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/procinfo−17.tar.bz2
- Procps (2.0.7) 153 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/procps−2.0.7.tar.bz2
- Psmisc (19) 20 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/psmisc−19.tar.bz2
- Shadow Password Suite (20000902) 557 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/shadow−20000902.tar.bz2
- Sysklogd (1.4) 67 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/sysklogd−1.4.tar.bz2
- Sysklogd Patch (1.4) 0.5 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/sysklogd−1.4.patch
- Sysvinit (2.78) 90 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/sysvinit−2.78.tar.bz2
- Sysvinit Patch (2.78) 1 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/sysvinit−2.78.patch
-

Util Linux (2.10r) 883 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/util−linux−2.10r.tar.bz2

- Man−pages (1.33) 475 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/man−pages−1.33.tar.bz2

- Netkit−base (0.17) 49 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/netkit−base−0.17.tar.bz2

- Net−tools (1.57) 187 KB:
ftp://packages.linuxfromscratch.org/pub/common−packages/net−tools−1.57.tar.bz2

- Taille totale de tous les paquetages Intel : 61 377 Ko (59,94 Mo).

Chapitre 4. Préparation d'une nouvelle partition

Introduction

Dans ce chapitre, on va préparer la partition qui accueillera le système LFS. On créera une nouvelle partition, puis, à l'intérieur, un système de fichiers ext2, et enfin la structure des répertoires. Quand ce sera fait, nous pourrons continuer avec le chapitre suivant et démarrer la construction d'un nouveau système Linux à partir de zéro.

Création d'une nouvelle partition

Avant de pouvoir construire notre nouveau système Linux, nous devons avoir une partition Linux vide dans laquelle nous pourrons créer notre nouveau système. je vous recommande une partition d'une taille d'environ 750 MB. Cela vous donne assez d'espace pour stocker toutes les archives et pour compiler tous les paquetages sans avoir à se préoccuper de manquer de l'espace disque temporaire nécessaire. Si vous avez déjà une partition Linux Native disponible, vous pouvez passer cette sous−section.

Démarrez le programme fdisk (ou tout autre programme fdisk que vous préférez) avec comme option le disque dur approprié (comme /dev/hda, si vous voulez créer une nouvelle partition sur le disque maître primaire IDE); créez une partition Linux Native, écrivez la table de partition et sortez du programme fdisk. Si vous obtenez un message vous disant de redémarrer votre système pour être sur que la table de partition est mise à jour, redémarrez votre système maintenant avant de continuer. Rappelez vous de l'identification de votre nouvelle partition; Ce peut être quelque chose comme hda11 (comme dans mon cas). Cette partition nouvellement créée sera appelée la partition LFS dans ce livre.

Création d'un système de fichiers ext2 dans la nouvelle partition

Dès que la partition est créée, nous devons créer un nouveau système de fichiers ext2 dans cette partition. Pour créer un nouveau système de fichiers ext2, nous utilisons la commande mke2fs. Entrez la nouvelle partition en tant qu'unique option et le système de fichiers sera créé. Si votre partition était hda11, vous exécuteriez :

```
mke2fs /dev/hda11
```

Montage d'une nouvelle partition

Maintenant que nous avons créé le système de fichiers ext2, il est prêt à être utilisé. Tout ce que nous avons à faire pour pouvoir y accéder (comme par exemple le lire ou écrire des rendez−vous dedans) c'est de le monter. Si vous le montez sous /mnt/lfs, vous pouvez accéder à la partition en allant au répertoire /mnt/lfs, puis faire tout ce que vous avez à faire. Ce livre supposera que vous avez monté la partition dans un sous−répertoire de /mnt. Le répertoire que vous choisirez n'a pas d'importance, mais prenez garde à vous souvenir de votre choix.

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.

ce répertoire (/mnt/lfs) est la variable $LFS que vous avez vue plus tôt. Donc, si vous lisez quelque part d'exécuter "cp inittab $LFS/etc", vous taperez en fait "cp inittab /mnt/lfs/etc". Ou, si vous voulez utiliser la variable d'environnement $LFS, exécutez **`export LFS=/mnt/lfs`** maintenant.

Création des répertoires

Créons l'arborescence des répertoires dans la partition LFS en conformité au standard FHS qui se trouve en http://www.pathname.com/fhs/. Exécuter les commandes suivantes créera les répertoires nécessaires :

```
cd $LFS
mkdir bin boot dev dev/pts etc home lib mnt proc root sbin
tmp  var
for dirname in $LFS/usr $LFS/usr/local
   do
   mkdir $dirname
   cd $dirname
   mkdir bin etc include lib sbin share  src tmp var
   ln -s share/man man
   ln -s share/doc doc
   ln -s share/info info
   cd $dirname/share
   mkdir dict doc info locale man nls misc  terminfo zoneinfo
   cd $dirname/share/man
   mkdir man1 man2 man3 man4 man5 man6 man7  man8
done
cd $LFS/var
mkdir lock log mail run spool tmp
```

Normalement, les répertoires sont créés avec le mode de permission 755, ce qui n'est pas souhaité pour tous les répertoires. Je n'ai pas vérifié dans le FHS si ils suggèrent des modes par défaut pour certains répertoires, alors je ne changerais les permissions que pour un petit nombre de répertoires pour lesquels ce changement a du sens. Le premier changement est le mode 0750 pour le répertoire $LFS/root. Ceci pour s'assurer que tout le monde ne peut pas entrer dans le répertoire /root (vous feriez la même chose pour les répertoires /home/username). Le deuxième changement est le mode 1777 pour les répertoires /tmp. De cette façon, tous les utilisateurs peuvent écrire dans le répertoire /tmp si ils en ont besoin. Le sticky (1) bit empêche les utilisateurs de supprimer les fichiers des autres utilisateurs, ce qu'ils pourraient normalement faire car le répertoire est configuré de façon à permettre à quiconque (propriétaire, groupe, monde) d'écrire dans ce répertoire.

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

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

Chapitre 5. Préparation du 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, vous aurez un système linux fonctionnel. Les chapitres restant traitent de la mise en place du réseau, de la création des scripts de démarrage et de l'addition d'une entrée au fichier lilo.conf pour pouvoir démarrer votre système LFS.

Dans ce chapitre, on procédera à l'édition de liens statiques des logiciels. Ces programmes seront réinstallés au chapitre suivant avec une édition de liens dynamiques. La raison de commencer par une version statique est qu'il est possible que notre système Linux normal et notre système LFS ne partagent 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.

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

Nous sommes sur le point d'installer le premier ensemble de packages. Ces packages seront, comme cela a été expliqué plus haut, liés statiquement.

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'information (principalement au sujet d'utilisation incorrecte, mais pas illégale, de la syntaxe de C ou de C++. Souvent, le standard du langage C change mais les packages utilisent encore le vieux standard, ce qui n'est pas un problème).

Avant de commencer, assurez−vous que vous avez positionné la variable LFS, si vous avez décidé de l'utiliser, en exécutant la commande suivante :

```
echo $LFS
```

Installer Bash

Installation de Bash

Installez Bash en exécutant les commandes suivantes :

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

Si vous rencontrez des erreurs en compilant bash qui vous disent ne pas trouver "–lcurses", exécutez les deux commandes suivantes pour créer le lien symbolique manquant (jusqu'à présent, nous n'avons jamais rencontré une distribution pour laquelle ce lien symbolique du paquetage libncurses est correct, excepté les systèmes LFS):

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

Note : Normalement, le fichier libncurses.a réside dans le répertoire /usr/lib, mais il peut aussi être dans /lib (comme dans un système LFS). Donc, vérifiez pour vous assurer si vous devez exécuter la commande ln dans /usr/lib ou dans /lib

---

Explication des commandes

`--enable-static-link:` Cette option de configuration entraîne l'édition de liens statique de Bash

`--prefix=$LFS/usr:` Cette option de configuration installe tous les fichiers de Bash sous le répertoire $LFS/usr, qui devient le répertoire /usr après un chroot dans $LFS ou quand vous redémarrez votre système sous LFS.

`--bindir=$LFS/bin:` Ceci installe les fichiers exécutables dans $LFS/bin. Nous faisons ceci car nous voulons que bash soit dans /bin, et non dans /usr/bin. Une des raisons est la suivante : votre partition (ndt: répertoire, en fait) /usr pourrait être une partition séparée qui doit être montée à un moment donné. Avant que cette partition ne soit montée, vous devez disposer de bash (il sera difficile d'exécuter les scripts de démarrage sans un shell, par exemple).

`--disable-nls:` Ceci interdit la construction de NLS (National Language Support, Support de la Langue Nationale). Ce serait actuellement uniquement une perte de temps car bash sera réinstallé au chapitre suivant.

`--with-curses:` Ceci a pour effet de lier Bash à la bibliothèque curses au lieu de la bibliothèque par défaut termcap, qui devient obsolète.

`ln -s bash sh:` Cette commande crée le lien symbolique sh qui pointe sur bash. La plupart des scripts s'exécutent via 'sh'; sh étant un lien symbolique vers le shell par défaut du système. Etant donné que les programmes et les scripts ne savent pas quel shell vous utilisez par défaut (cela pourrait être bash, ksh, korn, tch, csh ou d'autres) ils utilisent le lien symbolique commun sh qui, si le système est correctement configuré, pointe toujours vers le shell par défaut du système.

Les signes **`&&`** à la fin de chaque ligne indiquent que la commande suivante ne doit être exécutée que si la commande précédente se termine avec une valeur de retour de 0, indiquant le succès. Si vous copiez&collez toutes ces commandes dans le shell, vous voulez vous assurez que, si ./configure échoue, make ne sera pas exécuté, que si make échoue, make install ne sera pas exécuté et ainsi de suite.

Contenu

Le paquet Bash contient le programme bash.

Description

bash est le SHell Bourne−Again, qui est un interpréteur de commandes couramment utilisé sur les systèmes Unix. bash est un programme qui lit l'entrée standard, c'est−à−dire le clavier. Vous tapez quelque chose et le programme évalue ce que vous avez entré, et en fait quelque chose, comme lancer un programme.

Installer Binutils

Installation de Binutils

Installez Binutils en exécutant les commandes suivantes :

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

Explication des commandes

`make -e:` Le paramètre –e indique à make que les variables d'environnement ont priorité sur les variables définies dans le(s) fichier(s) Makefile. Ceci est nécessaire pour pouvoir faire avec succès l'édition de liens statique de binutils.

`LDFLAGS=-all-static:` Positionner la variable LDFLAGS à la valeur –all–static impose l'édition de liens statique de binutils.

`tooldir=$LFS/usr:` Normalement, le répertoire tooldir (le répertoire où se retrouvent les exécutables de binutils) est positionné à $(exec_prefix)/$(target_alias) ce qui vaut, par exemple /usr/i686–pc–linux–gnu. Étant donné que nous ne créons ces exécutables que pour notre système, nous n'avons pas besoin de ce répertoire spécifique à une cible dans $LFS/usr. Vous utiliseriez cette option si vous utilisiez votre système pour des compilations croisées (par exemple, vous compileriez un paquetage sur votre machine Intel qui générerait du code exécutable sur une machine Apple PowerPC)

Contenu

Le paquet Binutils contient les programmes ld, as, ar, nm, objcopy, objdump, ranlib, size, strings, strip, c++filt, addr2line et nlmconv.

Description

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.

---

as

as a d'abord été prévu pour assembler la sortie du compilateur GNU C gcc, pour pouvoir être utilisée par le lieur ld.

---

ar

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

---

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 dans 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 que leur programme compile et fonctionne.

---

ranlib

ranlib génère un index sur le 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.

---

size

size liste les tailles des sections −− et la taille totale −− de chaque fichier objet objfile dans sa liste d'arguments. Par défaut, une ligne est générée en sortie pour chaque fichier objet ou chaque module dans l'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 sont suivies d'un caractère non imprimable. Par défaut, il affiche seulement les chaînes des parties initialisées et chargées des fichiers objet; pour d'autres types de fichier, il affiche les chaînes de tout le fichier.

strings est essentiellement utilisé pour déterminer le contenu des fichiers non−texte.

---

strip

strip débarrasse les fichiers objet de tous les symboles, ou de certains en particuliers. 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.

---

c++filt

Le langage C++ fournit un mécanisme de surcharge, ce qui signifie que vous pouvez écrire plusieurs fonctions ayant le même nom (pourvu que chacune réclame 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 le lieur puisse prévenir un conflit entre ces fonctions surchargées.

---

addr2line

addr2line traduit les adresses d'un programme en noms de fichier 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.

---

nlmconv

nlmconv convertit des fichiers objet re−localisables en fichiers de Module Installables NetWare, pouvant aussi lire les fichiers d'entête pour les informations d'entête NLM.

Installer Bzip2

Installation de Bzip2

Installez Bzip2 en exécutant les commandes suivantes :

```
sed \
   s/"\$(CC) \$(CFLAGS) -o"/"\$(CC) \$(CFLAGS)  \$(LDFLAGS)
-o"/ \
   Makefile | make -f -  LDFLAGS=-static &&
make PREFIX=$LFS/usr install &&
cd $LFS/usr/bin &&
mv bzcat bunzip2 bzip2 bzip2recover $LFS/bin
```

Explication des commandes

`sed:` Ici, la commande sed recherche les chaînes de caractères "$(CC) $(CFLAGS) −o" et les remplace par "$(CC) $(CFLAGS) $(LDFLAGS) −o" dans le fichier Makefile. Nous faisons cette modification pour qu'il soit plus aisé de faire l'édition de lien statique de bzip2.

`...Makefile | make −f −:` Makefile est le dernier paramètre de la commande sed, et indique le fichier dans lequel on recherche et remplace. Normalement, sed envoie le fichier modifié vers stdout (la sortie standard) qui serait votre console. Avec la construction utilisée, les sorties de sed seront envoyées au programme make. Normalement, quand make est démarré, il cherche un certain nombre de fichiers comme Makefile. Mais nous avons modifié le fichier Makefile, aussi nous ne voulons pas que make l'utilise. Le paramètre "−f −" indique à make de lire son entrée d'un autre fichier, ou de stdin (l'entrée standard) ce que le paramètre tiret (−) implique. Ceci n'est pas la seule façon de procéder. Une autre manière aurait été de commander à sed d'écrire sa sortie dans un fichier différent, et d'indiquer à make, à l'aide du paramètre −f, de lire ce fichier alternatif.

`LDFLAGS=−static:` Ceci est la seconde manière de procéder à une édition de liens statique. C'est aussi la façon la plus commune. Comme vous le remarquerez, la valeur −all−static n'est utilisée qu'avec le paquetage binutils, et ne sera plus utilisé dans la suite de ce livre.

Contenu

Le package Bzip2 contient les programmes bzip2, bunzip2, bzcat et bzip2recover.

Description

bzip2

bzip2 compresse les fichiers avec l'algorithme de compression de texte par tri de blocs de Burrows−Wheeler, et le codage de Huffman. La compression est en général bien meilleure que celle obtenue par les compresseurs plus conventionnels basés sur LZ77/LZ78, et s'approche des performances de la famille PPM des compresseurs statistiques.

---

bunzip2

bunzip2 décompresse les fichiers qui ont été compressés avec bzip2.

---

bzcat

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

---

bzip2recover

bzip2recover restaure les données des fichiers bzip2 endommagés.

Installer Diffutils

Installation de Diffutils

Installez Diffutils en exécutant les commandes suivantes :

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

Explication des commandes

`CPPFLAGS=-Dre_max_failures=re_max_failures2:` La variable CPPFLAGS est lue par le programme cpp (C PreProcessor). La valeur de cette variable indique au préprocesseur de remplacer chaque instance de re_max_failures en re_max_failures2 avant de passer le fichier source au compilateur lui−même pour la compilation. Ce package a des problèmes quand on en 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

Le package Diffutils contient les programmes cmp, diff, diff3 et sdiff.

Description

cmp et diff

cmp et diff comparent tous deux deux fichiers et montrent leurs différences. Ces deux programmes ont des options pour comparer des fichiers dans différentes situations.

---

diff3

La différence entre diff et diff3 est que diff compare deux fichiers, et diff3 en compare trois.

---

sdiff

sdiff fusionne deux fichiers et affiche interactivement le résultat.

Installer Fileutils

Installation de Fileutils

Installez Fileutils en exécutant les commandes suivantes :

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

Explication de la commande

`--libexecdir=$LFS/bin:` Cette option de configuration va définir $LFS/bin comme étant le répertoire des exécutables. Normalement, ces exécutables sont dans /usr/libexec mais rien n'est mis dedans. Le changer empêche que ce répertoire soit créé pour rien.

Contenu

Le package Fileutils contient les programmes chgrp, chmod, chown, cp, dd, df, dir, dircolors, du, install, ln, ls, mkdir, mkfifo, mknod, mv, rm, rmdir, sync, touch et vdir.

Description

chgrp

chgrp modifie le groupe d'appartenance de chaque fichier donné en lui affectant le groupe spécifié, qui peut être un nom de groupe ou un ID de groupe numérique.

---

chmod

chmod modifie les permissions de chaque fichier spécifié selon le mode donné, qui peut être soit une représentation symbolique des changements à effectuer, ou un nombre octal représentant le schéma de bits pour les nouvelles permissions.

---

chown

chown modifie l'utilisateur et/ou le groupe d'appartenance de chaque fichier spécifié.

---

cp

cp copie les fichiers d'un endroit dans un autre.

---

dd

dd copie un fichier (depuis l'entrée standard vers la sortie standard, par défaut) avec une taille de bloc choisie par l'utilisateur, en pouvant éventuellement y faire des conversions.

---

df

df affiche la quantité d'espace disque disponible sur le système de fichiers contenant chaque nom de fichier en argument. Si aucun nom de fichier n'est donné, l'espace disponible sur tous les systèmes de fichier actuellement montés est affiché.

---

ls, dir et vdir

dir et vdir sont des versions de ls avec différents formats de sortie par défaut. Ces programmes listent chaque fichier ou répertoire spécifié. Le contenu des répertoires est trié alphabétiquement. Pour ls, les fichiers sont listés par défaut en colonne, triés verticalement, si la sortie standard est un terminal; autrement, ils sont listés un par ligne. Pour dir, les fichiers sont listés par défaut en colonne, triés verticalement. Pour vdir, les fichiers sont listés par défaut en format long.

---

dircolors

dircolors affiche les commandes pour positionner la variable d'environnement LS_COLOR. La variable LS_COLOR est utilisée pour modifier le schéma de couleurs par défaut utilisé par ls et les utilitaires associés.

---

du

du affiche la quantité d'espace disque utilisée par chaque argument, et pour chaque sous−répertoire du répertoire en argument.

---

install

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

ln

ln crée des liens physiques ou logiques (symboliques) entre fichiers.

mkdir

mkdir crée des répertoires avec un nom spécifié.

mkfifo

mkfifo crée un FIFO avec chaque nom spécifié.

mknod

mknod crée un FIFO, fichier spécial de type caractère ou bloc, avec le nom de fichier spécifié.

mv

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

rm

rm supprime des fichiers ou des répertoires.

rmdir

rmdir supprime des répertoires, s'ils sont vides.

sync

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

touch

touch positionne les heures d'accès et de modification de chaque fichier spécifié à l'heure courant. Les fichiers qui n'existent pas sont créés vides.

Installer GCC sur le système normal si nécessaire

Installation de GCC sur le système normal si nécessaire

Pour pouvoir par la suite compiler Glibc−2.1.3, vous avez besoin d'avoir installé gcc−2.95.2. Bien que toute version de GCC postérieure à 2.8 convienne, 2.95.2 est la version fortement recommandée à utiliser. egcs−2.91.x est aussi connu pour fonctionner. Si vous n'avez ni gcc−2.95.x ni egcs−2.91.x, vous devez installer gcc−2.95.2 sur votre système normal avant de pouvoir compiler Glibc, plus loin dans ce chapitre.

Pour savoir de quelle version de compilateur dispose votre système, exécuter la commande suivante :

```
gcc −−version
```

Si votre système Linux normal ne possède pas gcc−2.95.x ou egcs−2.91.x, vous devez l'installer maintenant. Au lieu de remplacer le compilateur présent dans votre système, nous allons installer gcc dans un répertoire séparé (/usr/local/gcc2952). De cette façon, aucun exécutable ou fichier d'en−tête ne sera remplacé.

Après avoir désarchivé gcc−2.95.2, n'entrez pas dans le répertoire gcc−2.95.2 qui s'est créé, mais restez dans le répertoire $LFS/usr/src. installez GCC en exécutant les commandes suivantes :

```
mkdir $LFS/usr/src/gcc−build &&
cd $LFS/usr/src/gcc−build &&
../gcc−2.95.2/configure \
   −−prefix=/usr/local/gcc2952 \
   −−with−local−prefix=/usr/local/gcc2952  \
     −−with−gxx−include−dir=/usr/local/gcc2952/include/g++ \
   −−enable−shared −−enable−languages=c,c++  &&
make bootstrap &&
make install
```

Explication des commandes

`−−with−local−prefix:` GCC installe un certain nombre de fichiers dans /usr/local même quand −−prefix est positionné à une autre valeur. Dans ce cas, nous ne voulons pas que cela se produise et c'est pourquoi nous utilisons l'option −−with−local−prefix pour changer ce chemin.

`−−with−gxx−include−dir:` GCC installe les en−têtes C++ dans /usr/include/g++ par défaut. De nouveau, dans ce cas nous ne voulons pas que cela se produise, nous voulons que cette version de GCC soit installée complètement sous /usr/local/gcc2952.

`make bootstrap:` Compile gcc en le "bootstrappant". Ici, cela veut dire que le compilateur sera reconstruit trois fois au total. Premièrement, il est compilé avec le compilateur par défaut de votre système (qui est en général un compilateur gcc ou egcs). Cela produit le compilateur Etape 1. Puis GCC se recompilera lui−même, mais au lieu d'utiliser le compilateur de votre système, il utilisera le compilateur

Étape 1. Cela produit le compilateur Étape 2. Puis il se recompilera de nouveau en utilisant le compilateur Étape 2. Enfin il comparera le résultat de la deuxième et de la troisième compilation. S'ils sont identiques, la compilation a été un succès.

Contenu

Le package GCC contient les compilateurs, les préprocesseurs et la bibliothèque GNU C++.

Description

Compilateur

Un compilateur traduit le code source sous format texte en un format que l'ordinateur peut comprendre. Après qu'un fichier de code source a été compilé en un fichier objet, un lieur crée un fichier exécutable à partir d'un ou plusieurs de ces fichiers objet générés par le compilateur.

Préprocesseur

Un préprocesseur pré−traduit un fichier source, faisant par exemple l'inclusion des contenus des fichiers d'en−tête dans le fichier source. Vous ne faites pas cela vous−même, en général, pour gagner du temps. Vous insérez simplement une ligne comme #include <nom_fichier>. Le préprocesseur insère le contenu de ce fichier dans le fichier source. C'est une des choses que fait le préprocesseur.

Bibliothèque C++

La bibliothèque C++ est utilisée par les programmes C++. La bibliothèque C++ contient les fonctions qui sont couramment utilisées dans les programmes C++. Ainsi, le programmeur n'a pas à ré−écrire certaines fonctions (comme écrire une chaîne de caractères à l'écran) à chaque fois qu'il crée un programme.

Installer GCC sur le système LFS

Installation de GCC sur le système LFS

Après avoir désarchivé gcc−2.95.2, n'entrez pas dans le répertoire gcc−2.95.2 qui s'est créé, mais restez dans le répertoire $LFS/usr/src. installez GCC en exécutant les commandes suivantes :

```
mkdir $LFS/usr/src/gcc-build &&
cd $LFS/usr/src/gcc-build &&
../gcc-2.95.2/configure --prefix=/usr \
   --with-gxx-include-dir=/usr/include/g++  \
   --enable-languages=c,c++ --disable-nls  &&
make -e LDFLAGS=-static bootstrap &&
make prefix=$LFS/usr local_prefix=$LFS/usr/local  \
   gxx_include_dir=$LFS/usr/include/g++  install &&
        cd $LFS/lib &&
        ln -s ../usr/lib/gcc-lib/*/2.95.2/cpp cpp  &&
        cd $LFS/usr/lib &&
        ln -s gcc-lib/*/2.95.2/cpp cpp  &&
        cd $LFS/usr/bin &&
        ln -s gcc cc
```

Explication des commandes

--enable-languages=c,c++: Ceci indique de ne créer que les compilateurs C et C++, et pas les autres compilateurs disponibles parce qu'ils ne sont en général pas très utilisés. Si vous avez besoin de ces autres compilateurs, n'utilisez pas le paramètre −−enable−languages.

ln -s ../usr/lib/gcc-lib/*/2.95.2/cpp cpp: Ceci crée le lien symbolique $LFS/lib/cpp. Certains packages essaient explicitement de trouver cpp dans /lib.

ln -s ../usr/lib/gcc-lib/*/2.95.2/cpp cpp: Ceci crée le lien symbolique $LFS/usr/lib/cpp, parce qu'il y a des packages qui supposent que cpp est dans /usr/lib.

Contenu

Le package GCC contient les compilateurs, les préprocesseurs et la bibliothèque GNU C++.

Description

Compilateur

Un compilateur traduit le code source sous format texte en un format que l'ordinateur peut comprendre. Après qu'un fichier de code source a été compilé en un fichier objet, un lieur crée un fichier exécutable à partir d'un ou plusieurs de ces fichiers objet générés par le compilateur.

Préprocesseur

Un préprocesseur pré−traduit un fichier source, faisant par exemple l'inclusion des contenus des fichiers d'en−tête dans le fichier source. Vous ne faites pas cela vous−même, en général, pour gagner du temps. Vous insérez simplement une ligne comme #include <nom_fichier>. Le préprocesseur insère le contenu de ce fichier dans le fichier source. C'est une des choses que fait le préprocesseur.

Bibliothèque C++

La bibliothèque C++ est utilisée par les programmes C++. La bibliothèque C++ contient les fonctions qui sont couramment utilisées dans les programmes C++. Ainsi, le programmeur n'a pas à ré−écrire certaines fonctions (comme écrire une chaîne de caractères à l'écran) à chaque fois qu'il crée un programme.

Installer le noyau Linux

Installation du noyau Linux

Nous n'allons pas déjà 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.

Créez les fichiers de configuration du noyau en exécutant la commande suivante :

```
yes "" | make config
```

Ignorez l'alerte *Broken pipe* que vous verrez peut−être à la fin. Maintenant, exécutez les commandes suivantes pour créer les dépendances correctement :

```
make dep
```

Maintenant que c'est fait, nous devons créer les liens symboliques `$LFS/usr/include/linux` et `$LFS/usr/include/asm`. Créez les en exécutant les commandes suivantes :

```
cd $LFS/usr/include &&
ln −s ../src/linux/include/linux linux &&
ln −s ../src/linux/include/asm asm
```

Explication des commandes

`yes "" | make config:` Ceci exécute make config et répond "Y" à toutes les questions que le script de configuration pose à l'utilisateur. Ici, nous ne configurons pas le véritable noyau, la seule chose dont nous avons besoin c'est d'un fichier de configuration quelconque, pour pouvoir ensuite exécuter make dep qui créera des fichiers, entre autres version.h, dans $LFS/usr/src/linux/include/linux 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, c'est que make dep crée dans $LFS/usr/src/linux/include/linux les fichiers susmentionnés dont nous aurons besoin plus tard.

`ln −s ../src/linux/include/linux linux` et **`ln −s ../src/linux/include/asm asm:`** Ces commandes créent les liens symboliques linux et asm dans le répertoire $LFS/usr/include et les font pointer vers les répertoires corrects dans l'arborescence des sources de Linux. Parmi les packages nécessitant les en−têtes du noyau, il y a ceux comportant des lignes comme #include <linux/errno.h>. Ces chemins sont relatifs au répertoire /usr/include, donc le lien symbolique /usr/include/linux pointe vers le

répertoire contenant les fichiers d'en−tête du noyau linux. Il en va de même pour le lien symbolique asm.

Contenu

Le package Noyau Linux contient le noyau de Linux.

Description

Le noyau de Linux est au cur 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.

Installer Glibc

Installation de Glibc

Désarchivez glibc−crypt et glibc−linuxthreads dans le répertoire glibc−2.1.3, pas dans le répertoire $LFS/usr/src. N'entrez pas dans les répertoires créés. Désarchivez et laissez les choses comme elles sont.

Il faut changer un petit nombre de paramètres de Glibc, comme le répertoire où les bibliothèques dynamiques sont supposées être installées et le répertoire qui contient les fichiers de configuration système. Pour le faire, vous devez créer le répertoire `$LFS/usr/src/glibc−build` et entrer dans ce répertoire avec :

```
mkdir $LFS/usr/src/glibc−build &&
cd $LFS/usr/src/glibc−build
```

dans ce répertoire vous créez le nouveau fichier `configparms` en exécutant ce qui suit :

```
cat > configparms << "EOF"
# Begin configparms

slibdir=/lib
sysconfdir=/etc

# End configparms
EOF
```

Avant d'installer effectivement Glibc vous devez désarchiver le fichier patch de Glibc

Notez que le script de configuration de Glibc peut protester en disant que certains fichiers du répertoire /usr/include sont trop anciens et seront remplacés, ou que certains liens symboliques ne sont plus supposés exister (comme le lien symbolique /usr/include/scsi qui est présent sur les vieux systèmes Linux). S'il vous demande de déplacer un lien symbolique comme scsi, faites le. S'il vous dit qu'il va remplacer des fichiers anciens par des fichiers Glibc plus récents, vous pouvez ne pas vous en préoccuper. Glibc ne sait pas qu'il finira dans $LFS quand le fichier de configuration aura été exécuté.

Si une version correcte de GCC est installée sur votre système, allez dans le répertoire `$LFS/usr/src/glibc−build` et installez Glibc en exécutant les commandes suivantes :

```
cd ../glibc−2.1.3 &&
patch −Np1 −i ../glibc−2.1.3.patch &&
cd $LFS/usr/src/glibc−build &&
../glibc−2.1.3/configure \
   −−prefix=/usr −−enable−add−ons  \
           −−with−headers=$LFS/usr/include  \
           −−libexecdir=/usr/bin &&
```

```
make &&
make install_root=$LFS install &&
make install_root=$LFS localedata/install-locales
```

Si aucune version correcte de GCC n'était installée sur votre système, allez dans le répertoire `$LFS/usr/src/glibc-build` et installez Glibc, en utilisant gcc-2.95.2 que vous venez d'installer, en exécutant les commandes suivantes :

```
cd ../glibc-2.1.3 &&
patch -Np1 -i ../glibc-2.1.3.patch &&
cd $LFS/usr/src/glibc-build &&
CC=/usr/local/gcc2952/bin/gcc \
   ../glibc-2.1.3/configure --prefix=/usr  --enable-add-ons \
           --with-headers=$LFS/usr/include \
           --libexecdir=/usr/bin &&
make &&
make install_root=$LFS install &&
make install_root=$LFS localedata/install-locales
```

Copier les anciennes bibliothèques NSS

Si votre système Linux normal utilise glibc-2.0, vous devez copier les bibliothèques NSS dans la partition LFS. Certains programmes liés statiquement dépendent des bibliothèques NSS, en particulier les programmes qui doivent interpréter les noms d'utilisateurs (username) , les identificateurs utilisateurs (userid) et les identificateurs de groupe (groupid). Vous pouvez vérifier quelle version de bibliothèque C est utilisée par votre système Linux normal en exécutant :

```
strings /lib/libc* | grep "release version"
```

Le résultat de cette commande devrait vous dire quelque chose comme :

GNU C Library stable release version 2.1.3, by Roland McGrath et al.

Si Glibc-2.0.x est installé sur votre distribution de départ, copiez les bibliothèques NSS en exécutant :

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

Explication des commandes

`patch -Np1 -i ../glibc-2.1.3.patch`: Ceci applique un patch qui fixe un défaut mineur dans

Glibc. Glibc définit quelques noms de variables comportant des caractères interdits. Bash−2.03, ainsi que les versions plus anciennes, ne protestent pas à l'inverse de Bash−2.04 qui ne compile pas Glibc correctement.

`--enable-add-ons:` Ceci inclut les ajouts que nous installons avec Glibc : linuxthreads et crypt.

`--with-headers=$LFS/usr/include:` Ceci indique à Glibc d'utiliser les fichiers en−têtes du noyau de notre système LFS, et pas ceux de la distribution de départ, qui peuvent être obsolètes ou modifiés.

`make install_root=$LFS:` Ceci est la façon de spécifier à Glibc l'équivalent de −−prefix=.

Contenu

Le package Glibc contient la bibliothèque GNU C.

Description

La bibliothèque C est un ensemble de fonctions couramment utilisées dans les programmes. Ainsi, un programmeur n'a pas besoin de créer ses propres fonctions pour chaque tâche de base. Les opérations les plus courantes comme écrire une ligne à l'écran sont déjà présentes et à la disposition du programmeur.

La bibliothèque C (comme, actuellement, presque toutes les bibliothèques) est fournie en deux versions: dynamique et statique. En bref, quand un programme utilise une bibliothèque C statique, le code de la bibliothèque C est copié dans le fichier exécutable. Quand un programme utilise une bibliothèque dynamique, cet exécutable ne contiendra pas le code de la bibliothèque C, mais simplement une routine qui chargera les fonctions depuis la bibliothèque au moment de l'exécution du programme. Cela signifie une réduction significative de la taille du fichier du programme. Si vous ne comprenez pas ce concept, vous feriez mieux de lire la documentation fournie avec la bibliothèque C, car cela est trop compliqué à expliquer ici en une ou deux lignes.

Installer Grep

Installation de Grep

Installez Grep en exécutant les commandes suivantes :

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

Contenu

Le package grep contient les programmes egrep, fgrep et grep.

Description

egrep

egrep affiche les lignes des fichiers qui vérifient une expression régulière étendue.

---

fgrep

fgrep affiche les lignes des fichiers qui vérifient une liste de chaînes fixes, séparées par des caractères de fin de ligne, l'une quelconque de ces chaînes devant correspondre.

---

grep

grep affiche les lignes des fichiers qui vérifient une expression régulière basique.

Installer Gzip

Installation de Gzip

Avant d'installer Gzip, il vous faut décompresser le fichier patch de gzip.

```
patch -Np1 -i ../gzip-1.2.4a.patch &&
./configure --prefix=$LFS/usr --disable-nls  &&
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
```

Contenu

Le package Gzip contient les programmes compress, gunzip, gzexe, gzip, uncompress, zcat, zcmp, zdiff, zforece, zgrep, zmore et znew.

Description

gunzip

gunzip décompresse les fichiers qui ont été compressés avec gzip.

---

gzexe

gzexe vous permet de compresser des exécutables en les remplaçant, et de faire en sorte qu'ils se décompressent automatiquement et s'exécutent lorsque vous les lancez (avec une dégradation des performances).

---

gzip

gzip réduit la taille des fichiers nommés en utilisant le codage de Lempel–Zip (LZ77).

---

zcat

zcat décompresse soit une liste de fichiers fournie en ligne de commande ou sur son entrée standard, et écrit les données décompressées sur la sortie standard.

---

zcmp

zcmp invoque le programme cmp sur des fichiers compressés.

---

zdiff

zdiff invoque le programme diff sur des fichiers compressés.

---

zforce

zforce force l'extension .gz sur tous les fichiers gzip, pour que gzip ne les compresse pas deux fois. Ceci peut être utile pour des fichiers dont les noms ont été tronqués après un transfert.

---

zgrep

zgrep invoque le programme grep sur des fichiers compressés.

---

zmore

zmore est un filtre qui permet d'examiner des fichiers texte simples ou compressés, écran par écran, sur un terminal (de la même manière que more).

---

znew

znew recompresse des fichiers du format .Z (compress) en format .gz (gzip).

Installer Make

Installation de Make

Installez Make en exécutant les commandes suivantes :

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

Contenu

Le package Make contient le programme make.

Description

make détermine automatiquement quels morceaux d'un gros programme ont besoin d'être recompilés, et exécute les commandes pour les recompiler.

Installer Sed

Installation de Sed

Installez Sed en exécutant les commandes suivantes :

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

Contenu

Le package Sed contient le programme sed.

Description

sed est un éditeur de flux. Un éditeur de flux est utilisé pour opérer des transformations de texte basiques sur un flux d'entrée (un fichier ou une entrée depuis un pipe).

Installer Shellutils

Installation de Sh–utils

Installez Shellutils en exécutant les commandes suivantes :

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

Contenu

Le package Shellutils contient les 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.

Description

basename

basename supprime le répertoire et les suffixes des noms de fichier.

---

chroot

chroot lance une commande ou un shell interactif avec un répertoire racine particulier.

---

date

date affiche la date et l'heure courantes dans un format spécifié, ou positionne la date du système.

---

dirname

dirname supprime les suffixes autres que le répertoire du nom de fichier.

---

echo

echo affiche une ligne de texte.

---

env

env lance un programme dans un environnement modifié.

---

expr

expr évalue les expressions.

---

factor

factor affiche les facteurs premiers de tous les nombres entiers spécifiés.

---

false

false se termine toujours avec un code de statut indiquant une erreur.

---

groups

groups affiche les groupes auxquels appartient un utilisateur.

---

hostid

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

---

hostname

hostname positionne ou affiche le nom de l'hôte courant du système.

---

id

id affiche les GID et UID réels et effectifs d'un utilisateur ou de l'utilisateur courant.

---

logname

logname affiche le nom de connexion de l'utilisateur courant.

nice

nice lance un programme avec une priorité de cadencement modifiée.

nohup

nohup lance une commande insensible aux interruptions, avec une sortie autre qu'un terminal.

pathchk

pathchk vérifie si des noms de fichier sont valides ou portables.

pinky

pinky est un utilitaire léger qui recherche des informations sur un utilisateur particulier.

printenv

printenv affiche tout ou partie de l'environnement.

printf

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

pwd

pwd affiche le nom du répertoire courant de travail.

seq

seq affiche des nombres dans une certaine plage, avec un incrément spécifié.

sleep

sleep attend pendant un laps de temps spécifié.

---

stty

stty modifie et affiche les paramètres de ligne du terminal.

---

su

su lance un shell avec des ID d'utilisateur et de groupe substitués.

---

tee

tee lit depuis l'entrée standard et écrit sur la sortie standard et dans des fichiers.

---

test

test vérifie les types de fichier et compare les valeurs.

---

true

true se termine 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 sur le système.

---

uptime

uptime indique depuis combien de temps le système tourne.

---

users

users affiche le nom des utilisateurs actuellement connectés sur l'hôte courant.

---

who

who montre qui est connecté.

---

whoami

whoami affiche votre id utilisateur effectif.

---

yes

yes sort une chaîne à répétition, jusqu'à ce qu'il soit tué.

Installer Tar

Installation de Tar

Si vous voulez pouvoir utiliser directement bzip2 avec tar, utilisez le patch disponible sur le site FTP de LFS. Ce patch ajoute l'option –y à tar qui fonctionne de la même manière que l'option –z, utilisée pour les fichiers compressés avec gzip.

Appliquez le patch en exécutant les commandes suivantes :

```
cd src &&
patch -i ../../gnutarpatch.txt &&
cd ..
```

Installez Tar en exécutant les commandes suivantes :

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

Contenu

Le package tar contient les programmes tar et rmt.

Description

tar

tar est un programme d'archivage conçu pour stocker et extraire des fichiers d'une archive au format d'un fichier tar.

---

rmt

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

Installer Textutils

Installation de Textutils

Installez Textutils en exécutant les commandes suivantes :

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

Contenu

Le paquet Textutils contient les programmes cat, cksum, comm, split, cut, expand, fmt, fold, head, join, md5sum, nl, od, paste, pr, ptx, sort, split, sum, tac, tail, tr, tsort, unexpand, uniq et wc.

Description

cat

cat concatène le(s) fichier(s) ou l'entrée standard vers la sortie standard.

---

cksum

cksum affiche la somme de contrôle CRC et le nombre d'octets de chaque fichier spécifié.

---

comm

comm compare deux fichiers triés ligne à ligne.

---

csplit

cplit découpe un fichier en morceaux séparés par un(des) motif(s) dans des fichiers xx01, xx02, ..., et affiche le nombre d'octets de chaque morceau sur la sortie standard.

---

cut

cut écrit les parties sélectionnées des lignes des fichiers spécifiés sur la sortie standard.

expand

expand convertit les tabulations dans les fichiers en espaces, en écrivant sur la sortie standard.

fmt

fmt reformate chaque paragraphe dans le(s) fichier(s) spécifié(s), en écrivant sur la sortie standard.

fold

fold emballe les lignes d'entrée de chaque fichier spécifié (entrée standard par défaut), en écrivant sur la sortie standard.

head

head affiche les xx (10 par défaut) premières lignes de chaque fichier spécifié sur la sortie standard.

join

join fusionne les lignes identiques de deux fichiers dans un fichier commun.

md5sum

md5sum affiche ou vérifie les sommes de contrôle MD5.

nl

nl écrit chaque fichier spécifié sur la sortie standard, en y ajoutant les numéros de ligne.

od

od écrit une représentation non−ambiguë, en octal par défaut, d'un fichier spécifié, sur la sortie standard.

paste

paste écrit des lignes, d'une manière consistant à prendre séquentiellement les lignes correspondantes de chaque fichier spécifié, séparées par des tabulations, sur la sortie standard.

---

pr

pr met en pages ou en colonnes des fichiers pour impression.

---

ptx

ptx produit un index de permutation du contenu d'un fichier.

---

sort

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

---

split

split extrait des morceaux de taille fixe d'un fichier d'entrée dans PREFIXaa, PREFIXab, ...

---

sum

sum écrit la somme de contrôle et le nombre de blocs de chaque fichier spécifié.

---

tac

tac écrit chaque fichier spécifié sur la sortie standard, la dernière ligne en premier.

---

tail

tail écrit les xx (10 par défaut) dernières lignes de chaque fichier spécifié sur la sortie standard.

---

tr

tr convertit, répète et/ou supprime des caractères de l'entrée standard, en écrivant sur la sortie standard.

---

tsort

tsort écrit des listes totalement ordonnées en accord avec l'ordonnancement partiel dans les fichiers spécifiés.

---

unexpand

unexpand convertit les espaces dans chaque fichier en tabulations, en écrivant sur la sortie standard.

---

uniq

uniq supprime toutes les lignes identiques successives sauf une depuis des fichiers ou l'entrée standard, et écrit dans des fichiers ou la sortie standard.

---

wc

wc affiche le nombre de lignes, mots et octets de chaque fichier spécifié, et une ligne de totalisation si plus d'un fichier est spécifié.

Créer les fichiers passwd et group

Pour que l'utilisateur et le groupe root soient reconnus et puissent se connecter, il faut une entrée dans les fichiers /etc/passwd et /etc/group. D'autre part, le groupe root ainsi que quelques autres sont recommandés et nécessaires à certains packages. Les groupes et leur GID ci–dessous ne font parties d'aucun standard. Le LSB recommande uniquement, à coté d'un groupe root, un groupe bin avec un GID de 1. Vous pouvez choisir vous même les noms et les GID des autres groupes. Les packages bien écrits ne dépendent pas de la valeur numérique du GID mais seulement du nom du groupe, le GID d'un groupe n'a donc pas une grande importance. Comme il n'y a aucun standard pour les groupes, je ne suivrai pas les conventions utilisées par Debian, RedHat ou par d'autres. Les groupes ajoutés ici sont ceux mentionnés dans le script MAKEDEV (le script qui crée les fichiers device dans le répertoire /dev).

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

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

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

```
cat > $LFS/etc/group << "EOF"
root:x:0:
bin:x:1:
sys:x:2:
kmem:x:3:
tty:x:4:
uucp:x:5:
daemon:x:6:
floppy:x:7:
disk:x:8:
EOF
```

Monter le système de fichiers $LFS/proc

Certains programmes, pour fonctionner correctement, ont besoin que le système de fichiers proc soit également monté et accessible depuis l'environnement chroot. Il n'y a pas de problèmes à monter le système de fichiers proc deux fois ou plus, parce que c'est un système de fichiers virtuel maintenu par le noyau.

Montez le système de fichiers proc sous $LFS/proc en exécutant la commande suivante :

```
mount proc $LFS/proc -t proc
```

Chapitre 6. Installer les logiciels du système de base

Introduction

L'installation de l'ensemble des logiciels est assez simple et vous allez penser qu'il serait bien plus facile et rapide de donner des instructions d'installation génériques pour tous les packages, et d'expliquer comment installer quelque chose uniquement si un package particulier requiert une méthode d'installation différente. Bien que je sois d'accord avec vous sur ce point, j'ai cependant choisi de fournir l'ensemble des instructions pour chaque package. Ceci dans le simple but d'éviter toute confusion possible et des erreurs en découlant.

A propos des symboles de déboguage

La plupart des programmes et des bibliothèques sont compilés par défaut avec les symboles de déboguage et un niveau d'optimisation de 2 (options de gcc −g et −O2), et sont prévus pour un CPU spécifique. Sur les plateformes Intel, les logiciels sont compilés pour les processeurs i386 par défaut. Si vous ne désirez pas faire fonctionner les logiciels sur d'autres machines que la vôtre, vous pouvez envisager de modifier les options par défaut du compilateur pour qu'il effectue les compilations avec un meilleur niveau d'optimisation, sans symboles de déboguage, et génère du code pour votre architecture spécifiquement. Expliquons d'abord ce sont les symboles de déboguage.

Un programme compilé avec des symboles de déboguage signifie que vous pouvez exécuter un programme ou une bibliothèque via un débogueur, et que les résultats du débogueur seront facilement utilisables. Ces symboles de déboguage augmentent également la taille du programme ou de la bibliothèque de façon significative.

Pour supprimer les symboles de déboguage d'un fichier binaire (qui doit être un binaire de type a.out ou ELF), lancez **`strip --strip-debug filename`**. Vous pouvez utiliser des jokers si vous devez supprimer les symboles de déboguage de plusieurs fichiers (utilisez quelque chose comme strip −−strip−debug $LFS/usr/bin/*). Une autre possibilité, plus facile, est de compiler les programmes sans les symboles de déboguage. La plupart des gens n'utiliseront sans doute jamais un débogueur sur leurs logiciels, donc enlever ces symboles vous fera gagner beaucoup de place disque.

Avant que vous vous demandiez si ces symboles de déboguage feront une grande différence, voici quelques statistiques:

- Un binaire Bash dynamique avec symboles de déboguage: 1.2Mo.
- Un binaire Bash dynamique sans symboles de déboguage: 478ko.
- /lib et /usr/lib (fichiers glibc et gcc) avec symboles de déboguage: 87Mo.
- /lib et /usr/lib (fichiers glibc et gcc) sans symboles de déboguage: 16Mo.

Les tailles peuvent différer selon le compilateur utilisé et la version de la bibliothèque C utilisée pour lier les programmes dynamiques, mais vos résultats seront similaires si vous comparez les programmes avec et sans symboles de déboguage. Après avoir terminé ce chapitre et supprimé tous les symboles de déboguage de tous les binaires et bibliothèques LFS, je regagné quelques 102Mo d'espace disque. Belle différence!

Créer $LFS/root/.bash_profile

Une fois entrés dans l'environnement chroot dans le chapitre suivant, nous voulons que soient exportées quelques variables d'environnement dans ce shell, comme PS1, PATH et d'autres variables que vous auriez positionnées. Pour cela, 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 lançant ce qui suit.

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

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

export PS1 PATH

# Fin /root/.bash_profile
EOF
```

Vous pouvez ajouter d'autres variables d'environnement, des alias et tout ce que vous voulez ou avez besoin, à votre discrétion, autant que vous l'estimez nécessaire.

Entrer dans l'environnement chroot

Il est temps d'entrer dans notre environnement chroot pour installer le reste des logiciels dont nous avons besoin.

Tapez la commande suivante pour entrer dans l'environnement chroot. A partir de maintenant, il n'est plus nécessaire d'utiliser la variable $LFS, parce que tout ce que vous ferez sera restreint à la partition LFS (puisque / est maintenant /mnt/lfs, mais le shell ne le sait pas).

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

Maintenant que nous sommes dans l'environnement chroot, nous pouvons continuer à installer tout les logiciels du système de base. Assurez−vous de bien exécuter toutes les commandes suivantes de ce chapitre à partir de l'environnement chroot.

Créer les fichiers de périphérique

Installation de MAKEDEV

Installez MAKEDEV en lançant les commandes suivantes:

```
sed "s/# 9/9/" MAKEDEV >/dev/MAKEDEV &&
chmod 754 /dev/MAKEDEV
```

Créer les entrées de /dev

Créez les fichiers de périphérique en lançant les commandes suivantes:

```
cd /dev &&
./MAKEDEV -v generic
```

Le paramètre "generic" passé au script MAKEDEV ne crée pas tous les périphériques dont vous aurez besoin, tels que les périphériques audio, hdc, hdd et autres. S'il vous semble qu'il manque quelque chose, demandez à MAKEDEV de le créer. Pour créer hdc, remplacez generic par hdc. Vous pouvez aussi ajouter hdc à generic, ce qui vous fera exécuter **`./MAKEDEV -v generic hdc`** pour créer l'ensemble des périphériques génériques, plus les fichiers dont vous aurez besoin pour accéder à hdc (et hdc1, hdc2, etc.).

Notez bien que ce script date de 1997 et peut donc être périmé, ne supportant plus les matériels les plus récents. Si vous avez besoin de fichiers de périphérique qui ne sont pas connus de ce script, lisez bien le fichier Documentation/devices.txt dans l'arborescence source de Linux. Ce fichier liste tous les numéros majeur et mineur pour tous les fichiers de périphériques que le noyau connaît. Avec cette liste, vous pouvez créer par vous-mêmes ces fichiers de périphérique. Reportez-vous à la page d'aide de mknod pour plus d'information sur la manière de construire les fichiers de périphériques par vous-mêmes.

Explication des commandes

`sed "s/# 9/9/" MAKEDEV >/dev/MAKEDEV:` Par défaut, le script Makefile crée seulement les périphériques hda1 à hda8 et hdb1 à hdb8. En remplaçant "# 9" par "9" dans le script MAKEDEV, il créera hda1 à hda20, hfb1 à hdb20 et éventuellement d'autres (comme hdc et hdd).

`chmod 754 /dev/MAKEDEV:` Ceci positionne les permissions du script MAKEDEV à 754, ce qui le rend exécutable uniquement pour le propriétaire et son groupe, et lisible par tous.

Contenu

Le paquet MAKEDEV contient le script MAKEDEV.

Description

MAKEDEV est un script qui peut vous aider à créer les fichiers de périphérique statiques nécessaires, qui sont d'habitude dans le répertoire /dev.

Installer Man−pages

Installation de Man−pages

Installez Man−pages en lançant les commandes suivantes:

```
make install
```

Contenu

Le paquet Man−pages contient différentes pages de manuel qui ne sont pas fournies avec les paquets.

Description

Des exemples de pages de manuel fournies sont les pages de manuel décrivant toutes les fonctions C et C++, quelques fichiers /dev/ importants, etc.

Installer Ed

Installation de Ed

Installez Ed en lançant les commandes suivantes:

```
./configure --prefix=/usr &&
make &&
make install &&
mv /usr/bin/ed /usr/bin/red /bin
```

Contenu

Le package Ed contient le programme ed.

Description

ed est un éditeur de texte par ligne. Il est utilisé pour créer, afficher, modifier et réaliser d'autres manipulations sur les fichiers texte.

Installer Patch

Installation de Patch

Installez Patch en lançant les commandes suivantes:

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

Contenu

Le package Patch contient le programme patch.

Description

Le programme patch modifie un fichier en accord avec un fichier patch. Un fichier patch est, en général, une liste créée par le programme diff, qui contient des instructions sur la manière dont un fichier original doit être modifié. patch est très utilisé pour les corrections de code source, car il fait économiser du temps et de la place. Imaginez que vous avez un pacakge d'une taille de 1Mo. Dans la version suivante de ce package ne sont modifiés que deux des fichiers de la première version. Vous pouvez mettre à disposition un package entièrement nouveau de 1Mo, ou fournir un fichier patch de 1ko qui mettra à jour la première version pour la rendre identique à la seconde. Ainsi, si vous avez déjà téléchargé la première version, un fichier patch peut vous éviter un second téléchargement important.

Installer Findutils

Installer Findutils

Avant d'installer Findutils, vous devez déballer le fichier patch de findutils.

Installez Findutils en lançant les commandes suivantes:

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

Contenu

Le package Findutils contient les programmes find, locate, updatedb et xargs.

Description

find

Le programme find recherche les fichiers, dans une hiérarchie de répertoires, qui correspondent à un certain critère. Si aucun critère n'est fourni, il liste tous les fichiers dans le répertoire courant et ses sous–répertoires.

---

locate

locate scrute une base de données qui contient tous les fichiers et répertoires d'un système de fichiers. Ce programme liste les fichiers et répertoires de cette base de données qui vérifient un certain critère. Si vous recherchez un fichier, ce programme va scruter la base de données et vous dire exactement où sont situés les fichiers que vous avez demandé. Cela n'a de sens que si la base de données de localisation est correctement mise à jour, autrement ce programme vous fournira des informations périmées.

---

updatedb

Le programme updatedb met à jour la base de données de localisation. Il scrute tout le système de fichiers (y compris les autres systèmes de fichiers actuellement montés, sauf si vous lui spécifiez de ne pas le faire), et enregistre chaque répertoire et fichier qu'il trouve dans la base de données qui sera utilisée par le programme locate, qui retrouve ces informations. C'est une bonne habitude que de mettre à jour cette base de données quotidiennement, pour vous assurer que la base de données est à jour.

---

xargs

La commande xargs applique une commande à une liste de fichiers. Si vous devez exécuter la même commande sur de multiples fichiers, vous pouvez créer un fichier qui contient tous ces fichiers (un par ligne) et utiliser xargs pour lancer une commande sur la liste.

Installer Mawk

Installation de Mawk

Installez Mawk en lançant les commandes suivantes:

```
./configure &&
make &&
make BINDIR=/usr/bin \
   MANDIR=/usr/share/man/man1 install  &&
cd /usr/bin &&
ln -s mawk awk
```

Contenu

Le paquet Mawk contient le programme mawk.

Description

mawk

mawk est un interpréteur pour le langage de programmation AWK. Le langage AWK est utile pour manipuler des fichiers de données, retrouver et travailler sur du texte, et pour prototyper et expérimenter des algorithmes.

Installer Ncurses

Installation de Ncurses

Installez Ncurses en lançant les commandes suivantes:

```
./configure --prefix=/usr --libdir=/lib \
   --with-shared  --disable-termcap &&
make &&
make install &&
cd /lib &&
ln -s libncurses.a libcurses.a
```

Explication des commandes

`--with-shared:` Ceci permet la construction des fichiers partagés de la bibliothèque ncurses.

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

`ln -s libncurses.a libcurses.a:` Ceci crée le lien symbolique /lib/libcurses.a qui, pour une raison inconnue, n'est pas créé pendant l'installation de libncurses.

Contenu

Le package Ncurses contient les bibliothèques ncurses, panel, menu et form. Il contient également les programmes tic, infocmp, clear, tput, toe et tset.

Description

Les bibliothèques

Les bibliothèques qui constituent la bibliothèque Ncurses sont utilisées pour afficher du texte (souvent de manière fantaisiste) sur votre écran. Un exemple d'utilisation de ncurses se trouve dans le processus "make menuconfig" du noyau. Les bibliothèques contiennent les routines pour créer des panneaux, des menus, des formulaires, et des routines générales d'affichage de texte.

---

tic

tic est le compilateur de description des entrées de terminfo. Le programme traduit un fichier terminfo du format source en format binaire utilisable avec les routines de la bibliothèque ncurses. Les fichiers terminfo contiennent des informations sur les capacités de votre terminal.

infocmp

Le programme infocmp peut être utilisé pour comparer une entrée binaire de terminfo avec d'autres, ré−écrire une description terminfo pour tenir compte du champ use= de terminfo, ou afficher une description terminfo du fichier binaire (term) sous différents formats (à l'opposé de ce que fait tic).

clear

Le programme clear efface votre écran si cela est possible. Il regarde dans l'environnement du type de terminal, puis dans la base de données terminfo, pour trouver comment effacer l'écran.

tput

Le programme tput utilise la base de données terminfo pour construire les valeurs des capacités dépendant du terminal et les informations disponibles pour le shell, pour initialiser ou réinitialiser le terminal, ou renvoyer le nom long du type de terminal requis.

toe

Le programme toe liste tous les types de terminaux disponibles par leur nom primaire, avec leur description.

tset

Le programme tset initialise les terminaux pour qu'ils puissent être utilisés, mais il n'est plus beaucoup utilisé désormais. Il est fourni pour la compatibilité avec BSD 4.4.

Installer Vim

Installation de Vim

Si vous ne voulez pas que vim soit installé comme éditeur sur votre système LFS, vous pouvez télécharger et installer un autre éditeur que vous préférez. Il existe quelques conseils sur la manière d'installer différents éditeurs disponibles à http://cvs.linuxfromscratch.org/index.cgi/hints/editors/.

Vous avez besoin de déballer à la fois les packages vim−rt et vim−src pour installer Vim. Les deux packages déballeront leurs fichiers dans le répertoire vim−5.7. Les fichiers des deux packages ne s'écrasent pas entre eux, donc l'ordre de déballage importe peu. Installez Vim en lançant les commandes suivantes:

```
./configure --prefix=/usr &&
make &&
make install &&
cd /usr/bin &&
ln -s vim vi
```

Si vous prévoyez d'installer le système X Window sur votre système LFS, vous pourrez recompiler Vim après l'installation de X. Vim est fourni avec une très jolie version GUI de l'éditeur, qui nécessite que X et quelques autres bibliothèques soient installés. Pour plus d'information, lisez la documentation de Vim.

Contenu

Le paquet Vim contient les programmes ctags, etags, ex, gview, gvim, rgview, rgvim, rview, rvim, view, vim, vimtutor et xxd.

Description

ctags

ctags génère des fichiers de marquage pour le code source.

---

etags

etags fait la même chose que ctags, mais il peut générer des fichiers de références croisées qui listent des informations sur les différents objets source trouvés dans un ensemble de fichiers de langage.

---

ex

ex démarre vim en mode Ex.

gview

gview est la version GUI de view.

gvim

gvim est la version GUI de vim.

rgview

rgview est la version GUI de rview.

rgvim

rgvim est la version GUI de rvim.

rview

rview est une version restreinte de view. Aucune commande shell ne peut être lancée et Vim ne peut pas être interrompu.

rvim

rvim est la version restreinte de vim. Aucune commande shell ne peut être lancée et Vim ne peut pas être interrompu.

view

view lance vim en mode lecture seule.

vim

vim lance vim de la manière normale, par défaut.

vimtutor

vimtutor lance le tutorial de Vim.

xxd

xxd crée un dump hexadécimal, ou fait l'inverse.

Installer GCC

Installation de GCC

Après avoir déballé l'archive gcc−2.95.2, ne rentrez pas dans le répertoire nouvellement créé gcc−2.95.2, mais restez dans le répertoire /usr/src. Installez GCC en lançant les commandes suivantes:

```
mkdir /usr/src/gcc-build &&
cd /usr/src/gcc-build &&
../gcc-2.95.2/configure --prefix=/usr \
   --with-gxx-include-dir=/usr/include/g++  \
   --enable-shared  --enable-languages=c,c++ &&
make bootstrap &&
make install
```

Contenu

Le package GCC contient les compilateurs, les préprocesseurs et la bibliothèque GNU C++.

Description

Compilateur

Un compilateur traduit le code source sous format texte en un format que l'ordinateur peut comprendre. Après qu'un fichier de code source a été compilé en un fichier objet, un lieur crée un fichier exécutable à partir d'un ou plusieurs de ces fichiers objet générés par le compilateur.

Préprocesseur

Un préprocesseur pré−traduit un fichier source, faisant par exemple l'inclusion des contenus des fichiers d'en−tête dans le fichier source. Vous ne faites pas cela vous−même, en général, pour gagner du temps. Vous insérez simplement une ligne comme #include <nom_fichier>. Le préprocesseur insère le contenu de ce fichier dans le fichier source. C'est une des choses que fait le préprocesseur.

Bibliothèque C++

La bibliothèque C++ est utilisée par les programmes C++. La bibliothèque C++ contient les fonctions qui sont couramment utilisées dans les programmes C++. Ainsi, le programmeur n'a pas à ré−écrire certaines fonctions (comme écrire une chaîne de caractères à l'écran) à chaque fois qu'il crée un programme.

Installer Bison

Installation de Bison

Installez Bison en lançant les commandes suivantes:

```
./configure --prefix=/usr \
   --datadir=/usr/share/bison  &&
make &&
make install
```

Certains programmes ignorent bison et tentent de trouver le programme yacc (bison est une −− meilleure −− alternative à yacc). Donc, pour satisfaire ces quelques programmes, nous allons créer un script yacc qui appelle bison et émule les conventions suivies par yacc pour dénommer ses fichiers de sortie.

Créez un nouveau fichier `/usr/bin/yacc` en lançant ce qui suit:

```
cat > /usr/bin/yacc << "EOF"
#!/bin/sh
# Debut /usr/bin/yacc

/usr/bin/bison −y $*

# Fin /usr/bin/yacc
EOF
chmod 755 /usr/bin/yacc
```

Explication des commandes

`--datadir=/usr/share/bison:` Ceci installe les fichiers de grammaire de bison dans /usr/share/bison plutôt que dans /usr/share.

Contenu

Le package Bison contient le programme bison.

Description

Bison est un générateur de parseur, en remplacement de YACC. YACC signifie Encore un Autre Compilateur de Compilateurs. Qu'est−ce donc que Bison? C'est un programme qui génère un programme qui analyse la

structure d'un fichier texte. Au lieu d'écrire le programme, vous spécifiez comment les choses doivent être reliées et, avec ces règles, un programme est construit pour analyser les fichiers texte.

Il y a de nombreux exemples où une structuration est nécessaire et l'un d'eux est la calculatrice.

Soit la chaîne:

```
1 + 2 * 3
```

Vous pouvez facilement arriver au résultat de 7. Pourquoi? A cause de sa structure. Vous savez 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
```

Vous commencez en bas de l'arbre, et vous arrivez aux nombres 2 et 3, qui sont liés par un symbole de multiplication; l'ordinateur multiplie donc 2 et 3. Le résultat de cette multiplication est mémorisé et la prochaine chose 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 uniquement utilisé pour les calculatrices.

Installer Less

Installation de Less

Installez Less en lançant les commandes suivantes:

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

Contenu

Le package Less contient le programme less.

Description

Le programme less est un paginateur de fichier (ou un visualisateur de texte). Il affiche le contenu d'un fichier avec la possibilité de défilement. Less est une évolution du paginateur habituel appelé "more". Less permet 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 quand vous lisez de gros fichiers.

Installer Groff

Installation de Groff

Installez Groff en lançant les commandes suivantes:

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

Contenu

Le package Groff contient les programmes addftinfo, afmtodit, eqn, grodvi, groff, grog, grohtml, grolj4, grops, grotty, hpftodit, indxbib, lkbib, lookbib, neqn, nroff, pfbtops, pic, psbb, refer, soelim, tbl, tfmtodit et troff.

Description

addftinfo

addftinfo lit un fichier de fonte troff et y ajoute des informations supplémentaires de métrique de fonte, afin qu'il soit utilisable par le système groff.

---

afmtodit

afmtodit crée un fichier de fonte pour être utilisé par groff et grops.

---

eqn

eqn compile les descriptions d'équations intégrées dans des fichiers d'entrée troff en commandes pouvant être comprises par troff.

---

grodvi

grodvi est un pilote pour groff qui produit un format dvi TeX.

---

groff

groff est un frontal pour le système de formatage de document groff. Normalement, il lance le programme troff et un postprocesseur approprié au périphérique choisi.

grog

grog lit des fichiers et tente de trouver lesquelles des options groff −e, −man, −me, −mn, −ms, −p, −s et −t sont nécessaires pour imprimer les fichiers, et affiche la commande groff incluant ces options sur la sortie standard.

grohtml

grohtml traduit la sortie de troff GNU en html.

grolj4

grolj4 est un pilote pour groff qui produit une sortie en format PCL5 utilisable par une imprimante HP Laserjet 4.

grops

grops traduit la sortie de troff GNU en PostScript.

grotty

grotty traduit la sortie de troff GNU en une forme convenable pour les périphériques de type machine à écrire.

hpftodit

hpftodit crée un fichier de fonte destiné à être utilisé avec groff −Tlj4 à partir d'un fichier de métrique de fonte spécifique HP.

indxbib

indxbib crée un index de permutation pour les bases de données bibliographiques à partie du fichier spécifié, destiné à être utilisé par refer, lookbib et lkbib.

lkbib

lkbib recherche dans les bases de données bibliographiques les références contenant les clés spécifiées, et affiche les références trouvées sur la sortie standard.

---

lookbib

lookbib affiche une invite sur la sortie d'erreur standard (sauf si l'entrée standard n'est pas un terminal), lit une ligne depuis l'entrée standard contenant un ensemble de mots−clés, recherche dans les bases de données bibliographiques dans le fichier spécifié les références contenant ces mots−clés, affiche les références trouvées sur la sortie standard, et répète le processus jusqu'à la fin de l'entrée.

---

neqn

Nous ne savons actuellement pas ce qu'est neqn, ni ce qu'il fait.

---

nroff

Le script nroff émule la commande nroff en utilisant groff.

---

pfbtops

pfbtops traduit une fonte PostScript de format .pfb en ASCII.

---

pic

pic compile les descriptions d'images incluses dans des fichiers d'entrée troff ou TeX en commandes qui peuvent être comprises par TeX ou troff.

---

psbb

psbb lit un fichier qui doit être un document PostScript conforme aux conventions de Structuration des Documents, et cherche un commentaire %%BoundingBox.

---

refer

refer copie le contenu d'un fichier sur la sortie standard, sauf que les lignes entre .[et .] sont interprétées comme des citations, et que les lignes entre .R1 et .R2 sont interprétées comme des commandes sur la manière de traiter les citations.

---

soelim

soelim lit les fichiers et remplace les lignes de la forme *fichier .so* par le contenu de *fichier*.

tbl

tbl compile les descriptions de tables intégrées dans les fichiers d'entrée troff en commandes qui peuvent être comprises par troff.

tfmtodit

tfmtodit crée un fichier de fonte à utiliser avec **`groff -Tdvi`**.

troff

troff est en grande partie compatible avec le troff Unix. En général, il doit être invoqué avec la commande groff, qui lancera aussi les préprocesseurs et les postprocesseurs dans l'ordre approprié, et avec les options qui conviennent.

Installer Man

Installation de Man

Installez Man en lançant les commandes suivantes:

```
./configure -default &&
make &&
make install &&
sed s/AWK=/"AWK=\/usr\/bin\/mawk"/ /usr/sbin/makewhatis  >
makewhatis-new &&
mv makewhatis-new /usr/sbin/makewhatis &&
chmod 755 /usr/sbin/makewhatis
```

Explication des commandes

`-default:` Ceci configure le package man avec les choix par défaut.

`sed s/AWK=/"AWK=\/usr\/bin\/mawk"/ /usr/sbin/makewhatis > makewhatis-new:` Ceci modifie la variable AWK de /usr/bin/makewhatis et remplit l'emplacement du programme mawk.

`chmod 755 /usr/sbin/makewhatis:` Ceci rend le script makewhatis de nouveau exécutable.

Contenu

Le paquet Man contient les programmes man, apropos, whatis et makewhatis.

Description

man

man formate et affiche les pages de manuel en ligne.

---

apropos

apropos recherche dans un ensemble de fichiers de base de données contenant des descriptions brèves des commandes système, en fonction de mots−clés, et affiche le résultat sur la sortie standard.

---

whatis

whatis recherche dans un ensemble de fichiers de base de données contenant des descriptions brèves de commandes système, en fonction de mots−clés, et affiche le résultat sur la sortie standard. Seules les correspondances de mots complets sont affichées.

---

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 avec le contenu de la section NAME de la page du manuel.

Installer Perl

Installation de Perl

Installez Perl en lançant les commandes suivantes:

```
./Configure -Dprefix=/usr &&
make &&
make install
```

Si vous ne désirez pas répondre à toutes les questions que Perl vous pose, vous pouvez ajouter l'option –d au script de configuration, et Perl utilisera tous les paramètres par défaut. Pour éviter que le script Configure vous questionne après la création du fichier config.sh, vous pouvez passer l'option –e à perl. Les commandes avec ces paramètres ressemblent à:

```
./Configure -Dprefix=/usr -d -e &&
make &&
make install
```

Contenu

Le package Perl contient Perl – Extraction et Rapport Pratiques

---

Description

Perl combine les capacités et les fonctionnalités de C, awk, sed et sh dans un puissant langage de programmation.

Installer M4

Installation de M4

Installez M4 en lançant les commandes suivantes:

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

Si votre système de base tourne avec un noyau 2.0 et que votre version de Glibc est la 2.1, alors il est plus que probable que vous aurez des difficultés à exécuter M4 dans l'environnement chroot, à cause d'incompatibilités entre le programme M4, Glibc−2.1 et le noyau en exécution 2.0. Si vous avez des problèmes pour exécuter le programme m4 dans l'environnement chroot (par exemple en installant les packages Autoconf et Automake), vous devrez sortir de l'environnement chroot et compiler M4 en statique. De cette manière, le binaire sera lié avec Glibc 2.0 (si vous tournez avec un noyau 2.0, votre version de Glibc est au mieux 2.0 sur un système correct. Le noyau 2.0 et Glibc−2.1 ne vont pas très bien ensemble) et ne vous posera plus aucun problème.

Pour créer une version de M4 liée statiquement, exécutez les commandes suivantes:

```
logout
cd $LFS/usr/src/m4-1.4
./configure --prefix=/usr --disable-nls
make LDFLAGS=-static
make prefix=$LFS/usr install
```

Maintenant, vous pouvez de nouveau entrer dans l'environnement chroot et continuer avec le pacakge suivant. Si vous voulez recompiler M4 dynamiquement, vous pourrez le faire après avoir redémarré dans le système LFS au lieu d'y être via chroot.

```
chroot $LFS env -i HOME=/root bash --login
```

Contenu

Le package M4 contient le processeur M4.

Description

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 les fichiers nommés, lancer des commandes UNIX, faire de l'arithmétique d'entiers, manipuler du texte de diverses façons, faire de la récursivité, etc. M4 peut être utilisé soit comme frontal d'un compilateur, ou comme processeur de macros à part.

Installer Texinfo

Installation de Texinfo

Installez Texinfo en lançant les commandes suivantes:

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

Contenu

Le paquet Texinfo contient les programmes info, install–info, makeinfo, texi2dvi et texindex.

Description

info

Le programme info lit les documents Info, en général situés dans votre répertoire /usr/doc/info. Les documents Info sont comme des pages de man(uel), mais ont tendance à aller plus en profondeur que la simple explication des options d'un programme.

---

install–info

Le programme install–info met à jour les entrées d'info. Quand vous exécutez le programme info, une liste de sujets disponibles (c'est–à–dire les documents info disponibles) est affichée. Le programme install–info est utilisé pour tenir cette liste de sujets disponibles à jour. Si vous décidez de supprimer des fichiers d'info manuellement, vous devrez également supprimer le sujet dans le fichier d'index. Ce programme est utilisé dans ce but. Il fonctionne également, dans l'autre sens, quand vous ajoutez des documents d'info.

---

makeinfo

Le programme makeinfo traduit des documents source Texinfo en différents 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.

Installer Autoconf

Installation d'Autoconf

Installez Autoconf en lançant les commandes suivantes:

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

Contenu

Le paquet Autoconf contient les programmes autoconf, autoheader, autoreconf, autoscan, autoupdate et ifnames.

Description

autoconf

autoconf est un outil pour produire des scripts shell qui configureront automatiquement les paquets de code source logiciel pour les adapter à de nombreux systèmes UNIX. Les scripts de configuration produits par Autoconf sont indépendants d'Autoconf quand ils sont lancés; leurs utilisateurs n'ont donc pas besoin d'avoir Autoconf.

---

autoheader

Le programme autoheader peut créer un fichier modèle de commandes C #define à utiliser pour la configuration.

---

autoreconf

Si vous avez beaucoup de scripts générés par autoconf, le programme autoreconf peut vous faire gagner du temps. Il lance autoconf (et autoheader, si besoin) de façon répétée pour reconstruire les scripts de configuration Autoconf et les modèles d'entête de configuration dans l'arborescence rattachée au répertoire courant.

---

autoscan

Le programme autoscan peut vous aider à créer un fichier configure.in pour un paquet logiciel. autoscan

scrute les fichiers source dans une arborescence rattachée à un répertoire donné en argument de ligne de commande, ou le répertoire courant si aucun n'est spécifié. Il recherche les fichiers source pour des problèmes habituels de portabilité et crée un fichier configure.scan qui est un préliminaire à configure.in pour ce paquet.

autoupdate

Le programme autoupdate met à jour un fichier configure.in qui appelle les macros Autoconf par leurs anciens noms, afin qu'il utilise les noms de macro actuels.

ifnames

ifnames peut servir lors de l'écriture d'un configure.in pour un paquet logiciel. Il affiche les identifiants que le paquet utilise déjà dans les conditions de préprocesseur C. Si un paquet a déjà été mis à jour pour une portabilité quelconque, ce programme peut vous aider à trouver ce que sa configuration doit vérifier. Il peut aider à remplir quelques trous dans un configure.in généré par autoscan.

Installer Automake

Installation d'Automake

Installez Automake en lançant les commandes suivantes:

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

Contenu

Le paquet Automake contient les programmes aclocal et automake.

Description

aclocal

Automake inclut un certain nombre de macros Autoconf qui peuvent être utilisées dans votre paquet; certaines d'entre elles sont requises par Automake, dans certains cas. Ces macros doivent être définies dans votre aclocal.m4; autrement, elles ne seront pas vues par autoconf.

Le programme aclocal va générer automatiquement les fichiers aclocal.m4, en fonction du contenu de configure.in. Ceci fournit un bon moyen d'obtenir des macros fournies par Automake, sans avoir à rechercher ici ou là. De plus, le mécanisme aclocal peut être étendu pour pouvoir être utilisé pour d'autres paquets.

---

automake

Pour créer tous les fichiers Makefile.in d'un paquet, lancez le programme automake dans le répertoire racine, sans arguments. automake va chercher automatiquement chaque Makefile.am approprié (en parcourant configure.in) et générer le Makefile.in correspondant.

Installer Bash

Installation de Bash

Installez Bash en lançant les commandes suivantes:

```
./configure --prefix=/usr --with-curses  &&
make &&
make install &&
logout
```

Remplacez le bash statique par le bash dynamique et entrez de nouveau dans l'environnement chroot en lançant:

```
mv $LFS/usr/bin/bash $LFS/usr/bin/bashbug  $LFS/bin &&
chroot $LFS /usr/bin/env -i HOME=/root /bin/bash  --login
```

Contenu

Le paquet Bash contient le programme bash.

Description

bash est le SHell Bourne−Again, qui est un interpréteur de commandes couramment utilisé sur les systèmes Unix. bash est un programme qui lit l'entrée standard, c'est−à−dire le clavier. Vous tapez quelque chose et le programme évalue ce que vous avez entré, et en fait quelque chose, comme lancer un programme.

Installer Flex

Installation de Flex

Installez Flex en lançant les commandes suivantes:

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

Contenu

Le package Flex contient le programme flex.

Description

flex est un outil pour générer des programmes qui reconnaissent des motifs dans le texte. La reconnaissance de motifs est très utile dans de nombreuses applications. Vous définissez les règles de recherche, et flex crée un programme qui recherche ces motifs. La raison pour laquelle les gens utilisent flex est qu'il est plus facile de définir des règles sur ce qu'il faut rechercher, que d'écrire un programme qui scrute le texte pour cela.

Installer File

Installation de File

Installez File en lançant les commandes suivantes:

```
./configure --prefix=/usr  --datadir=/usr/share/misc &&
make &&
make install
```

Contenu

Le pacakge File contient le programme file.

Description

file teste chaque fichier spécifié pour essayer de le classer. Il y a trois sortes de tests, réalisés dans cet ordre: tests du système de fichiers, test des nombres magiques et test de langage. Le premier test qui réussit provoque l'affichage du type de fichier.

Installer Libtool

Installation de Libtool

Installez Libtool en lançant les commandes suivantes:

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

Contenu

Le package Libtool contient les programmes libtool et libtoolize. Il contient aussi la bibliothèque ltdl.

Description

libtool

libtool fournit des services généraux de support de construction des bibliothèques.

---

libtoolize

libtoolize fournit un moyen standard d'ajouter un support libtool à votre package.

Bibliothèque ltdl

Libtool fournit une petite bibliothèque, appelée `libltdl', qui a pour but de masquer les différentes difficultés des bibliothèques dlopening aux programmeurs.

Installer Bin86

Installation de Bin86

Installez Bin86 en lançant les commandes suivantes:

```
make &&
make PREFIX=/usr install
```

Contenu

Le paquet Bin86 contient les programmes as86, as86_encap, ld86, objdump86, nm86 et size86.

Description

as86

as86 est un assembleur pour les processeurs 8086...80386.

---

as86_encap

as86_encap est un script shell pour appeler as86 et convertir le binaire créé en fichier C prog.v à inclure ou à lier avec des programmes comme les installeurs de bloc de démarrage.

---

ld86

ld86 comprend uniquement les fichiers objet construit par l'assembleur as86; il peut soit les lier en un exécutable I&D séparé ou impur.

---

objdump86

Pas de description disponible.

---

nm86

Pas de description disponible.

---

size86

Pas de description disponible.

Installer Binutils

Installation de Binutils

Installez Binutils en lançant les commandes suivantes:

```
./configure --prefix=/usr --enable-shared &&
make -e tooldir=/usr &&
make -e tooldir=/usr install
```

Contenu

Le paquet Binutils contient les programmes ld, as, ar, nm, objcopy, objdump, ranlib, size, strings, strip, c++filt, addr2line et nlmconv.

Description

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.

---

as

as a d'abord été prévu pour assembler la sortie du compilateur GNU C gcc, pour pouvoir être utilisée par le lieur ld.

---

ar

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

---

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 dans 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 que leur programme compile et fonctionne.

---

ranlib

ranlib génère un index sur le 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.

---

size

size liste les tailles des sections −− et la taille totale −− de chaque fichier objet objfile dans sa liste d'arguments. Par défaut, une ligne est générée en sortie pour chaque fichier objet ou chaque module dans l'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 sont suivies d'un caractère non imprimable. Par défaut, il affiche seulement les chaînes des parties initialisées et chargées des fichiers objet; pour d'autres types de fichier, il affiche les chaînes de tout le fichier.

strings est essentiellement utilisé pour déterminer le contenu des fichiers non−texte.

---

strip

strip débarrasse les fichiers objet de tous les symboles, ou de certains en particuliers. 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.

---

c++filt

Le langage C++ fournit un mécanisme de surcharge, ce qui signifie que vous pouvez écrire plusieurs fonctions ayant le même nom (pourvu que chacune réclame 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 le lieur puisse prévenir un conflit entre ces fonctions surchargées.

---

addr2line

addr2line traduit les adresses d'un programme en noms de fichier 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.

---

nlmconv

nlmconv convertit des fichiers objet re−localisables en fichiers de Module Installables NetWare, pouvant aussi lire les fichiers d'entête pour les informations d'entête NLM.

Installer Bzip2

Installation de Bzip2

Installez Bzip2 en lançant les commandes suivantes:

```
make -f Makefile-libbz2_so &&
make bzip2recover libbz2.a &&
cp bzip2-shared /bin/bzip2 &&
cp bzip2recover /bin &&
cp bzip2.1 /usr/share/man/man1 &&
cp bzlib.h /usr/include &&
cp -a libbz2.so* libbz2.a /lib &&
rm /usr/lib/libbz2.a &&
cd /bin &&
rm bunzip2 && ln -s bzip2 bunzip2 &&
rm bzcat && ln -s bzip2 bzcat &&
cd /usr/share/man/man1 &&
ln -s bzip2.1 bunzip2.1 &&
ln -s bzip2.1 bzcat.1 &&
ln -s bzip2.1 bzip2recover.1
```

Bien que ce ne soit pas, à proprement parler, une partie du système LFS de base, il vaut le coup de mentionner que vous pouvez télécharger un patch pour Tar qui permet au programme tar de compresser et décompresser facilement en utilisant bzip2/bunzip2. Avec un tar normal, vous devez utiliser des enchaînements tels que bzcat fichier.tar.bz | tar xv ou tar --use-compress-prog=bunzip2 -xvf fichier.tar.bz2 pour utiliser bzip2 et bunzip2 avec tar. Ce patch vous fournit l'option -y pour décompacter des archives Bzip2 avec tar xvfy fichier.tar.bz2. L'application de ce patch sera décrite plus tard quand vous réinstallerez le paquetage Tar.

Explication des commandes

`make -f Makefile-libbz2_so:` Ceci lance la construction de bzip2 avec un fichier Makefile différent, en l'occurrence Makefile-libbz2_so, qui crée une bibliothèque dynamique libbz2.so et lie les utilitaires bzip2 avec.

Contenu

Le package Bzip2 contient les programmes bzip2, bunzip2, bzcat et bzip2recover.

Description

bzip2

bzip2 compresse les fichiers avec l'algorithme de compression de texte par tri de blocs de Burrows−Wheeler, et le codage de Huffman. La compression est en général bien meilleure que celle obtenue par les compresseurs plus conventionnels basés sur LZ77/LZ78, et s'approche des performances de la famille PPM des compresseurs statistiques.

bunzip2

bunzip2 décompresse les fichiers qui ont été compressés avec bzip2.

bzcat

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

bzip2recover

bzip2recover restaure les données des fichiers bzip2 endommagés.

Installer Gettext

Installation de Gettext

Installez Gettext en lançant les commandes suivantes:

```
./configure --prefix=/usr &&
make &&
make install &&
mv /po-mode.el /usr/share/gettext
```

Contenu

Le pacakge gettext contient les programmes gettext, gettextize, msgcmp, msgcomm, msgfmt, msgmerge, msgunfmt et xgettext.

Description

gettext

Le pacakge gettext est utilisé pour l'internationalisation (connue aussi sous le terme de i18n) et pour la localisation (connue aussi sous le terme de l10n). Les programmes peuvent être compilés avec le Support du Langage Natal (NLS) qui leur permet d'afficher les messages dans votre langue natale plutôt qu'en anglais, par défaut.

Installer Console−tools

Installation de Console−tools

Avant de commencer à installer Console−tools, vous devez déballer le fichier console−tools−0.2.3.patch.

Installez Console−tools en lançant les commandes suivantes:

```
patch −Np1 −i ../console−tools−0.2.3.patch  &&
./configure −−prefix=/usr &&
make &&
make install &&
cd doc/man &&
mv consolechars.8.in consolechars.8  &&
cp *.1 /usr/share/man/man1 &&
cp *.4 /usr/share/man/man4 &&
cp *.5 /usr/share/man/man5 &&
cp *.8 /usr/share/man/man8
```

Contenu

Le package Console−tools contient les programmes charset, chvt, consolechars, deallocvt, dumpkeys, fgconsole, fix_bs_and_del, font2psf, getkeycodes, kbd_mode, loadkeys, loadunimap, mapscrn, mk_modmap, openvt, psfaddtable, psfgettable, psfstriptable, resizecons, saveunimap, screendump, setfont, setkeycodes, setleds, setmetamode, setvesablank, showcfont, showkey, splitfont, unicode_start, unicode_stop, vcstime, vt−is−URF8 et writevt.

Description

charset

charset positionne un ACM à utiliser dans un des emplacements de jeu de caractères G0/G1.

---

chvt

chvt modifie le terminal virtuel en avant−plan.

---

codepage

Pas de description disponible.

---

consolechars

consolechars charge les fontes d'écran pour console EGA/VGA, les cartes de fonte d'écran et/ou les cartes de jeu de caractères d'application.

---

deallocvt

deallocvt désalloue les terminaux virtuels inutilisés.

---

dumpkeys

dumkeys renvoie les tables de traduction du clavier.

---

fgconsole

fgconsole affiche le nombre de terminaux virtuels actifs.

---

fix_bs_and_del

Pas de description disponible.

---

font2psf

Pas de description disponible.

---

getkeycodes

getkeycodes affiche la table de correspondance du noyau: "code saisi vers code touche".

---

kbd_mode

kbd_mode renvoie ou positionne le mode du clavier.

---

loadkeys

loadkeys charge les tables de traduction du clavier.

---

loadunimap

Pas de description disponible.

mapscrn

Pas de description disponible.

mk_modmap

Pas de description disponible.

openvt

openvt lance un programme sur un nouveau terminal virtuel.

psfaddtable

psfaddtable ajoute une table de caractères Unicode à une fonte de console.

psfgettable

psfgettable extrait une table de caractères Unicode contenue dans une fonte de console.

psfstriptable

psfstriptable supprime la table de caractères Unicode contenue dans une fonte de console.

resizecons

resizecons modifie l'image qu'a le noyau de la taille de la console.

saveunimap

Pas de description disponible.

screendump

Pas de description disponible.

setfont

Pas de description disponible.

setkeycodes

setkeycodes charge les entrées de la table de correspondance du noyau: "code saisi vers code touche".

setleds

setleds positionne les diodes du clavier.

setmetamode

setmetamode définit la gestion des méta−touches du clavier.

setvesablank

Pas de description disponible.

showcfont

showcfont affiche tous les caractères dans la fonte d'écran courante.

showkey

showkey examine les codes saisis et les codes touche envoyés par le clavier.

splitfont

Pas de description disponible.

unicode_start

unicode_start met la console en mode Unicode.

---

unicode_stop

Pas de description disponible.

---

vcstime

Pas de description disponible.

---

vt−is−UTF8

vt−is−UTF8 vérifie si le terminal virtuel courant est en mode UTF8 ou octet.

---

writevt

Pas de description disponible.

Installer Console−data

Installation de Console−data

Remplacez <chemin−fichier−kmap> ci−dessous par le chemin correct vers le fichier kmap.gz voulu. Un exemple peut être i386/qwerty/us.kmap.gz.

Installez Console−data en lançant les commandes suivantes:

```
./configure --prefix=/usr &&
make &&
make install &&
cd /usr/share/keymaps &&
ln -s <chemin-fichier-kmap> defkeymap.kmap.gz
```

Contenu

Le paquet console−data contient les fichiers de données utilisés et réclamés par le package console−tools.

Installer Diffutils

Installation de Diffutils

Installez Diffutils en lançant les commandes suivantes:

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

Contenu

Le package Diffutils contient les programmes cmp, diff, diff3 et sdiff.

Description

cmp et diff

cmp et diff comparent tous deux deux fichiers et montrent leurs différences. Ces deux programmes ont des options pour comparer des fichiers dans différentes situations.

---

diff3

La différence entre diff et diff3 est que diff compare deux fichiers, et diff3 en compare trois.

---

sdiff

sdiff fusionne deux fichiers et affiche interactivement le résultat.

Installer E2fsprogs

Installation de E2fsprogs

Installez E2fsprogs en lançant les commandes suivantes:

Notez bien l'option vide −−with−root−prefix= ci−dessous. Elle doit bien rester telle quelle. Je n'ai pas oublié de fournir une valeur ici.

```
./configure --prefix=/usr --with-root-prefix= \
   --enable-elf-shlibs &&
make &&
make install &&
make install-libs
```

Contenu

Le package e2fsprogs contient les programmes chattr, lsattr, uuidgen, badblocks, debugfs, dumpe2fs, e2fsck, e2label, fsck, fsck.ext2, mke2fs, mkfs.ext2, mklost+found et tune2fs.

Description

chattr

chattr modifie les attributs d'un fichier sur un système de fichier Linux ext2.

---

lsattr

lsattr liste les attributs d'un fichier sur un système de fichiers 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.

---

badblocks

badblocks est utilisé pour rechercher les secteurs défectueux sur un périphérique (en général une partition de

disque).

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.

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.

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 répertoire 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.

tune2fs

tune2fs ajuste les paramètres d'optimisation de système de fichiers sur un système de fichiers Linux ext2.

Installer Fileutils

Installation de Fileutils

Installez Fileutils en lançant les commandes suivantes:

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

Contenu

Le package Fileutils contient les programmes chgrp, chmod, chown, cp, dd, df, dir, dircolors, du, install, ln, ls, mkdir, mkfifo, mknod, mv, rm, rmdir, sync, touch et vdir.

Description

chgrp

chgrp modifie le groupe d'appartenance de chaque fichier donné en lui affectant le groupe spécifié, qui peut être un nom de groupe ou un ID de groupe numérique.

---

chmod

chmod modifie les permissions de chaque fichier spécifié selon le mode donné, qui peut être soit une représentation symbolique des changements à effectuer, ou un nombre octal représentant le schéma de bits pour les nouvelles permissions.

---

chown

chown modifie l'utilisateur et/ou le groupe d'appartenance de chaque fichier spécifié.

---

cp

cp copie les fichiers d'un endroit dans un autre.

---

dd

dd copie un fichier (depuis l'entrée standard vers la sortie standard, par défaut) avec une taille de bloc choisie par l'utilisateur, en pouvant éventuellement y faire des conversions.

---

df

df affiche la quantité d'espace disque disponible sur le système de fichiers contenant chaque nom de fichier en argument. Si aucun nom de fichier n'est donné, l'espace disponible sur tous les systèmes de fichier actuellement montés est affiché.

---

ls, dir et vdir

dir et vdir sont des versions de ls avec différents formats de sortie par défaut. Ces programmes listent chaque fichier ou répertoire spécifié. Le contenu des répertoires est trié alphabétiquement. Pour ls, les fichiers sont listés par défaut en colonne, triés verticalement, si la sortie standard est un terminal; autrement, ils sont listés un par ligne. Pour dir, les fichiers sont listés par défaut en colonne, triés verticalement. Pour vdir, les fichiers sont listés par défaut en format long.

---

dircolors

dircolors affiche les commandes pour positionner la variable d'environnement LS_COLOR. La variable LS_COLOR est utilisée pour modifier le schéma de couleurs par défaut utilisé par ls et les utilitaires associés.

---

du

du affiche la quantité d'espace disque utilisée par chaque argument, et pour chaque sous−répertoire du répertoire en argument.

---

install

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

---

ln

ln crée des liens physiques ou logiques (symboliques) entre fichiers.

---

mkdir

mkdir crée des répertoires avec un nom spécifié.

mkfifo

mkfifo crée un FIFO avec chaque nom spécifié.

mknod

mknod crée un FIFO, fichier spécial de type caractère ou bloc, avec le nom de fichier spécifié.

mv

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

rm

rm supprime des fichiers ou des répertoires.

rmdir

rmdir supprime des répertoires, s'ils sont vides.

sync

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

touch

touch positionne les heures d'accès et de modification de chaque fichier spécifié à l'heure courant. Les fichiers qui n'existent pas sont créés vides.

Installer Grep

Installation de Grep

Installez Grep en lançant les commandes suivantes:

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

Contenu

Le package grep contient les programmes egrep, fgrep et grep.

Description

egrep

egrep affiche les lignes des fichiers qui vérifient une expression régulière étendue.

---

fgrep

fgrep affiche les lignes des fichiers qui vérifient une liste de chaînes fixes, séparées par des caractères de fin de ligne, l'une quelconque de ces chaînes devant correspondre.

---

grep

grep affiche les lignes des fichiers qui vérifient une expression régulière basique.

Installer Gzip

Installation de Gzip

Installez Gzip en lançant les commandes suivantes:

```
./configure --prefix=/usr &&
make &&
make install &&
cd /usr/bin &&
mv gzip /bin &&
rm gunzip /bin/gunzip &&
cd /bin &&
ln -s gzip gunzip &&
ln -s gzip compress &&
ln -s gunzip uncompress
```

Contenu

Le package Gzip contient les programmes compress, gunzip, gzexe, gzip, uncompress, zcat, zcmp, zdiff, zforece, zgrep, zmore et znew.

Description

gunzip

gunzip décompresse les fichiers qui ont été compressés avec gzip.

---

gzexe

gzexe vous permet de compresser des exécutables en les remplaçant, et de faire en sorte qu'ils se décompressent automatiquement et s'exécutent lorsque vous les lancez (avec une dégradation des performances).

---

gzip

gzip réduit la taille des fichiers nommés en utilisant le codage de Lempel–Zip (LZ77).

---

zcat

zcat décompresse soit une liste de fichiers fournie en ligne de commande ou sur son entrée standard, et écrit les données décompressées sur la sortie standard.

---

zcmp

zcmp invoque le programme cmp sur des fichiers compressés.

---

zdiff

zdiff invoque le programme diff sur des fichiers compressés.

---

zforce

zforce force l'extension .gz sur tous les fichiers gzip, pour que gzip ne les compresse pas deux fois. Ceci peut être utile pour des fichiers dont les noms ont été tronqués après un transfert.

---

zgrep

zgrep invoque le programme grep sur des fichiers compressés.

---

zmore

zmore est un filtre qui permet d'examiner des fichiers texte simples ou compressés, écran par écran, sur un terminal (de la même manière que more).

---

znew

znew recompresse des fichiers du format .Z (compress) en format .gz (gzip).

Installer Ld.so

Installation de Ld.so

Installez Ld.so en lançant les commandes suivantes:

```
cd util &&
make ldd ldconfig &&
cp ldd /bin &&
cp ldconfig /sbin &&
cd ../man &&
cp ldd.1 /usr/share/man/man1 &&
cp *.8 /usr/share/man/man8 &&
rm /usr/bin/ldd &&
hash -r
```

La commande "hash −r" est là pour forcer bash à oublier la localisation des commandes précédemment exécutées. Si vous avez lancé ldd auparavant, bash s'attend à le trouver dans /usr/bin. Comme nous l'avons déplacé dans /bin, le cache doit être nettoyé pour que bash puisse le trouver dans /bin quand vous voudrez l'exécuter de nouveau.

Vous aurez peut−être remarqué que nous n'utilisons pas les optimisations du compilateur pour ce pacakge. La raison en est que surcharger la variable CFLAGS entraîne des problèmes de compilation. Vous pouvez éditer le fichier Config.mk et ajouter les valeurs appropriées à la variable CFLAGS, puis compiler le package. Si vous voulez le faire, c'est votre choix. Je ne pense pas que cela en vaille la peine. Les programmes ld et ldd sont en général rarement utilisés.

Contenu

Du package Ld.so, nous utiliserons les programmes ldconfig et ldd.

Description

ldconfig

ldconfig crée les liens et la mémoire cache nécessaires (pour une utilisation par le lieur dynamique, ld.so) pour les bibliothèques partagées les plus récentes trouvées dans les répertoires spécifiés sur la ligne de commande, dans le fichier /etc/ld.so.conf, et dans les répertoires par défaut (/usr/lib et /lib). ldconfig vérifie l'en−tête et le nom de fichier des bibliothèques trouvées en déterminant quelles versions doivent voir leurs liens mis à jour.

---

ldd

ldd affiche les bibliothèques partagées requises par chaque programme ou la bibliothèque partagée spécifiée sur la ligne de commande.

Installer Lilo

Installation de Lilo

Installez Lilo en lançant les commandes suivantes:

```
make &&
make install
```

Il semble que la compilation de ce pacakge échoue sur certaines machines quand l'option du compilateur –g est utilisée. Si vous ne parvenez pas à compiler Lilo ainsi, supprimez la valeur –g de la variable CFLAGS dans le fichier `Makefile`.

A la fin de l'installation, le processus « make install » écrira un message indiquant que vous devez exécuter /sbin/lilo pour terminer la mise à jour. Ne le faites pas, car cela ne sert à rien. Le fichier /etc/lilo.conf n'est pas encore présent. Nous complèterons l'installation de Lilo dans le chapitre 8.

Contenu

Le package Lilo contient le programme lilo.

Description

lilo installe le chargeur de démarrage de Linux, qui est utilisé pour lancer un système Linux.

Installer Make

Installation de Make

Installez Make en lançant les commandes suivantes:

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

Contenu

Le package Make contient le programme make.

Description

make détermine automatiquement quels morceaux d'un gros programme ont besoin d'être recompilés, et exécute les commandes pour les recompiler.

Installer Modutils

Installation de Modutils

Installez Modutils en lançant les commandes suivantes:

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

Contenu

Le package Modutils contient les programmes depmod, genksyms, insmod, insmod_ksymoops_clean, kerneld, kernelversion, ksyms, lsmod, modinfo, modprobe et rmmod.

Description

depmod

depmod gère les descriptions de dépendance pour les modules du noyau qui peuvent être chargés.

---

genksyms

genksyms lit (sur l'entrée standard) la sortie de gcc –E source.c et génère un fichier contenant des informations de version.

---

insmod

insmod installe dans le noyau utilisé un module qui peut être chargé.

---

insmod_ksymoops_clean

insmod_ksymoops_clean supprime les ksym et les modules sauvés qui n'ont pas été accédés depuis deux jours.

---

kerneld

kerneld exécute une action du noyau dans l'espace utilisateur (comme le chargement sur demande de modules).

---

kernelversion

kernelversion renvoie la version majeure du noyau utilisé.

---

ksyms

ksyms affiche les symboles exportés du noyau.

---

lsmod

lsmod fournit les informations sur tous les modules chargés.

---

modinfo

modinfo examine un fichier objet associé à un module du noyau et affiche toutes les informations qu'il peut y trouver.

---

modprobe

modprobe utilise un fichier de dépendance ressemblant à Makefile, créé par depmod, pour charger automatiquement le(s) module(s) pertinent(s) de l'ensemble des modules disponibles, dans l'arborescence des répertoires prédéfinie.

---

rmmod

rmmod décharge des modules chargés du noyau utilisé.

Installer Procinfo

Installation de Procinfo

Installez Procinfo en lançant les commandes suivantes:

```
sed "s/-ltermcap/-lncurses/" Makefile | make -f -  &&
make install
```

Explication des commandes

`sed "s/XConsole/#XConsole/" Makefile | make -f -`: Ceci mettra en commentaire la variable XConsole dans le fichier Makefile et enverra la sortie de sed (le Makefile modifié) directement dans le programme make. Ceci est une alternative et un moyen plus efficace de diriger la sortie dans un fichier puis de dire à make d'utiliser ce fichier. La construction de XConsole est désactivée parce qu'elle ne peut pas encore être construite, puisque nous n'avons pas encore installé X.

Contenu

Le package Procinfo contient le programme procinfo.

Description

procinfo recueille des données système depuis le répertoire /proc et les affiche de façon correctement formatée sur le périphérique de sortie standard.

Installer Procps

Installation de Procps

Installez Procps en lançant les commandes suivantes:

```
sed "s/XConsole/#XConsole/" Makefile | make -f -  &&
sed "s/XConsole/#XConsole/" Makefile | make -f - install  &&
mv /usr/bin/kill /bin
```

Contenu

Le package Procps contient les programmes free, kill, oldps, ps, skill, snice, sysctl, tload, top, uptime, vmstat, w et watch.

Description

free

free affiche la quantité totale de mémoire, libre et utilisée, physique ou d'échange, dans le système, ainsi que la mémoire partagée et les mémoires tampon utilisées par le noyau.

---

kill

kills envoie des signaux aux processus.

---

oldps and ps

ps fournit un instantané des processus en cours.

---

skill

skill envoie des signaux aux processus répondant à un critère donné.

---

snice

snice modifie la priorité d'ordonnancement des processus répondant à un critère donné.

---

sysctl

sysctl modifie les paramètres du noyau en exécution.

---

tload

tload affiche un graphe de la charge moyenne du système courant sur le terminal spécifié (ou le terminal du processus tload si aucun autre n'est spécifié).

---

top

top fournit une vue de l'activité en cours du processeur en temps réel.

---

uptime

uptime affiche une ligne contenant les informations suivantes: l'heure courante, la durée de fonctionnement du système, le nombre d'utilisateurs actuellement connectés, et les moyennes de charge du système pour les 1, 5 et 15 dernières minutes.

---

vmstat

vmstat donne des informations sur les processus, la mémoire, la pagination, les entrées/sorties de blocs, les trappes et l'activité du cpu.

---

w

w affiche des informations sur les utilisateurs actuellement connectés à la machine, et leurs processus.

---

watch

watch lance une commande de façon répétitive, affichant sa sortie (le premier écran complet).

Installer Psmisc

Installation de Psmisc

Installez Psmisc en lançant les commandes suivantes:

```
sed "s/-ltermcap/-lncurses/" Makefile | make -f -  &&
make install
```

Contenu

Le package Psmisc contient les programmes fuser, killall et pstree.

Description

fuser

fuser affiche le PID des processus utilisant les fichiers ou les systèmes de fichiers spécifiés.

killall

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

pstree

pstree montre les processus en cours sous la forme d'un arbre.

Installer Sed

Installation de Sed

Installez Sed en lançant les commandes suivantes:

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

Contenu

Le package Sed contient le programme sed.

Description

sed est un éditeur de flux. Un éditeur de flux est utilisé pour opérer des transformations de texte basiques sur un flux d'entrée (un fichier ou une entrée depuis un pipe).

Installer Shellutils

Installation de Shellutils

Installez Shellutils en lançant les commandes suivantes:

```
./configure --prefix=/usr &&
make &&
make install &&
cd /usr/bin &&
mv date echo false pwd stty /bin &&
mv su true uname hostname /bin
```

Contenu

Le package Shellutils contient les 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.

Description

basename

basename supprime le répertoire et les suffixes des noms de fichier.

---

chroot

chroot lance une commande ou un shell interactif avec un répertoire racine particulier.

---

date

date affiche la date et l'heure courantes dans un format spécifié, ou positionne la date du système.

---

dirname

dirname supprime les suffixes autres que le répertoire du nom de fichier.

---

echo

echo affiche une ligne de texte.

---

env

env lance un programme dans un environnement modifié.

---

expr

expr évalue les expressions.

---

factor

factor affiche les facteurs premiers de tous les nombres entiers spécifiés.

---

false

false se termine toujours avec un code de statut indiquant une erreur.

---

groups

groups affiche les groupes auxquels appartient un utilisateur.

---

hostid

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

---

hostname

hostname positionne ou affiche le nom de l'hôte courant du système.

---

id

id affiche les GID et UID réels et effectifs d'un utilisateur ou de l'utilisateur courant.

---

logname

logname affiche le nom de connexion de l'utilisateur courant.

nice

nice lance un programme avec une priorité de cadencement modifiée.

nohup

nohup lance une commande insensible aux interruptions, avec une sortie autre qu'un terminal.

pathchk

pathchk vérifie si des noms de fichier sont valides ou portables.

pinky

pinky est un utilitaire léger qui recherche des informations sur un utilisateur particulier.

printenv

printenv affiche tout ou partie de l'environnement.

printf

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

pwd

pwd affiche le nom du répertoire courant de travail.

seq

seq affiche des nombres dans une certaine plage, avec un incrément spécifié.

sleep

sleep attend pendant un laps de temps spécifié.

---

stty

stty modifie et affiche les paramètres de ligne du terminal.

---

su

su lance un shell avec des ID d'utilisateur et de groupe substitués.

---

tee

tee lit depuis l'entrée standard et écrit sur la sortie standard et dans des fichiers.

---

test

test vérifie les types de fichier et compare les valeurs.

---

true

true se termine 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 sur le système.

---

uptime

uptime indique depuis combien de temps le système tourne.

---

users

users affiche le nom des utilisateurs actuellement connectés sur l'hôte courant.

---

who

who montre qui est connecté.

---

whoami

whoami affiche votre id utilisateur effectif.

---

yes

yes sort une chaîne à répétition, jusqu'à ce qu'il soit tué.

Installer Shadowpwd

Installation de la Suite de Masquage des mots de passe

Installez la Suite de Masquage des mots de passe en lançant les commandes suivantes:

```
./configure --prefix=/usr &&
make &&
make install &&
cd etc &&
cp limits login.access \
   login.defs.linux shells suauth  /etc &&
mv /etc/login.defs.linux /etc/login.defs
```

Explication des commandes

`cp limits login.access and others:` Ces fichiers ne sont pas installés pendant l'installation du package ; nous devons donc les copier manuellement, car ces fichiers sont utilisés pour configurer les détails d'authentification sur votre système.

Contenu

La Suite de Masquage des mots de passe contient les programmes chage, chfn, chsh, expiry, faillog, gpasswd, lastlog, login, newgrp, passwd, sg, su, chpasswd, dpasswd, groupadd, groupdel, groupmod, grpck, grpconv, grpunconv, logoutd, mkpasswd, newusers, pwck, pwconv, pwunconv, useradd, userdel, usermod et vipw.

Description

chage

chage modifie le nombre de jours entre deux changements de mot de passe et la date du dernier changement de mot de passe.

---

chfn

chfn modifie le nom complet de l'utilisateur, le numéro de travail, le poste de travail et le numéro de téléphone du domicile pour un compte utilisateur.

---

chsh

chsh modifie le shell de connexion de l'utilisateur.

---

expiry

Ce que fait ce programme est actuellement inconnu.

---

faillog

faillog formate le contenu du journal d'échec /var/log/faillog, et conserve le nombre et les limites des échecs.

---

gpasswd

gpasswd est utilisé pour administrer le fichier /etc/group.

---

lastlog

lastlog formate et affiche le contenu du journal de dernière connexion, /var/log/lastlog. Le nom de connexion, le port et l'heure de dernière connexion seront affichés.

---

login

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

---

newgrp

newgrp est utilisé pour modifier l'ID du groupe courant pendant une session de connexion.

---

passwd

passwd modifie les mots de passe des comptes utilisateur et groupe.

---

sg

sg exécute une commande avec un ID de groupe différent.

---

su

su modifie les ID effectifs de groupe et d'utilisateur en ceux d'un utilisateur. Ceci remplace le programme su installé avec le paquet Shellutils.

---

chpasswd

chpasswd lit un fichier contenant des paires nom d'utilisateur et mot de passe depuis l'entrée standard, et utilise ces informations pour mettre à jour un groupe d'utilisateurs existants.

---

dpasswd

dpasswd ajoute, supprime et met à jour les mots de passe d'appel pour les shells de connexion utilisateur.

---

groupadd

La commande groupadd crée un nouveau compte de groupe avec les valeurs spécifiées en ligne de commande et les valeurs par défaut du système.

---

groupdel

La commande groupdel modifie les fichiers de compte du système, supprimant toutes les entrées faisant référence au groupe.

---

groupmod

La commande groupmod modifie les systèmes de compte du système pour prendre en compte les changements spécifiés sur la ligne de commande.

---

grpck

grpck vérifie l'intégrité des informations d'authentification du système.

---

grpconv

grpconv convertit en fichiers de groupe masqués des fichiers de groupe normaux.

---

grpunconv

grpunconv convertit en fichiers de groupe normaux des fichiers de groupe masqués.

logoutd

logoutd applique les restrictions de temps de connexion et de port spécifiées dans /etc/porttime.

mkpasswd

mkpasswd lit un fichier dans le format donné par les options et le convertit dans le formet de fichier de base de données correspondant.

newusers

newusers lit un fichier contenant des paires nom d'utilisateur et mot de passe en clair, et utilise ces informations pour mettre à jour un groupe d'utilisateurs existants ou pour créer de nouveaux utilisateurs.

pwck

pwck vérifie l'intégrité des informations d'authentification du système.

pwconv

pwconv convertit en fichiers de mot de passe masqués des fichiers de mot de passe normaux.

pwunconv

pwunconv convertit en fichiers de mots de passe normaux des fichiers de mot de passe masqués.

useradd

useradd crée un nouvel utilisateur ou met à jour les informations par défaut du nouvel utilisateur.

userdel

userdel modifie les fichiers de compte du système, supprimant toutes les entrées faisant référence au nom de connexion spécifié.

usermod

usermod modifie les fichiers de compte du système pour refléter les modifications spécifiées sur la ligne de commande.

---

vipw et vigr

vipw et vigr éditent respectivement les fichiers /etc/passwd et /etc/group. Avec l'option –s, ils éditent les versions masquées de ces fichiers, respectivement /etc/shadow et /etc/gshadow.

Installer Sysklogd

Installation de Sysklogd

Installez Sysklogd en lançant les commandes suivantes:

```
patch -Np1 -i ../sysklogd-1.4.patch  &&
make &&
make install
```

Contenu

Le package Sysklogd contient les programmes klogd et syslogd.

Description

klogd

klogd est un démon système qui intercepte et journalise les messages du noyau de Linux.

---

syslogd

syslogd fournit une sorte de journalisation que de nombreux programmes modernes utilisent. Chaque message journalisé contient au moins une heure et un champ nom d'hôte, normalement un champ nom de programme également, mais cela dépend de la fidélité du programme qui journalise.

Installer Sysvinit

Installation de Sysvinit

Quand vous modifiez les niveaux d'exécution (par exemple quand vous allez arrêter votre système), le programme init envoie les signaux TERM et KILL à tous les processus qu'init a lancés. Mais init affiche un message à l'écran disant "sending all processes the TERM signal" (envoi du signal TERM à tous les processus), et la même chose pour le signal KILL. Ce message signifie qu'init envoie ce signal à tous les processus actuellement en train de tourner, ce qui n'est en fait pas le cas. Pour éviter cette confusion, vous pouvez appliquer le patch sysvinit trouvé sur le site FTP de LFS à sysvinit, qui modifie le message dans le fichier shutdown.c et fait afficher "sending all processes started by init the TERM signal" (envoi du signal TERM à tous les processus lancés par init).

Appliquez le patch en lançant la commande suivante:

```
patch -Np1 -i ../sysvinit-2.78.patch
```

Installez Sysvinit en lançant les commandes suivantes:

```
cd src &&
make &&
make install
```

Contenu

Le package Sysvinit contient les programmes pidof, last, lastb, mesg, utmpdump, wall, halt, init, killall5, poweroff, reboot, runlevel, shutdown, sulogin et telinit.

Description

pidof

pidof trouve les id des processus (pid) des programmes spécifiés et affiche ces id sur la sortie standard.

---

last

last recherche en arrière dans le fichier /var/log/wtmp (ou le fichier désigné par l'option –f) et affiche une liste de tous les utilisateurs s'étant connectés (et déconnectés) depuis la création du fichier.

---

lastb

lastb est identique à last, sauf que, par défaut, il affiche un journal du fichier /var/log/btmp, qui contient toutes les tentatives de connexion ayant échouées.

---

mesg

mesg contrôle l'accès à votre terminal par d'autres. Typiquement, ce programme est utilisé pour autoriser ou interdire à d'autres utilisateurs d'écrire sur votre terminal.

---

utmpdump

utmpdumps affiche le contenu d'un fichier (en général /var/run/utmp) sur la sortie standard dans un format compréhensible par l'utilisateur.

---

wall

wall envoie un message à toutes les personnes connectées ayant leur permission mesg positionnée à yes.

---

halt

halt note que le système est en train de s'arrêter dans le fichier /var/log/wtmp, puis dit au noyau soit de s'arrêter, soit de redémarrer, ou d'éteindre le système. Si halt ou reboot est appelé quand le système n'est pas dans un niveau d'exécution de 0 ou 6, shutdown sera appelé à la place (avec une option –h ou –r).

---

init

init est le père de tous les processus. Son rôle principal est de créer les processus à partir d'un script enregistré dans le fichier /etc/inittab. Ce fichier a en général des entrées sur chaque ligne indiquant à init d'engendrer des getty, pour que l'utilisateur puisse se connecter. Il contrôle aussi les processus autonomes requis par un système particulier.

---

killall5

killall5 est la commande killall de SystemV. Il envoie un signal à tous les processus, sauf les processus dans sa propre session, de manière à ne pas tuer le shell qui fait tourner le script qui l'a appelé.

---

poweroff

poweroff est équivalent à shutdown –h –p now. Il arrête l'ordinateur et l'éteint (si un BIOS compatible APM

est utilisé, et qu'APM est activé dans le noyau).

reboot

reboot est équivalent à shutdown −r now. Il redémarre l'ordinateur.

runlevel

runlevel lit le fichier utmp du système (typiquement, /var/run/utmp) pour localiser l'enregistrement du niveau d'exécution, puis écrit les niveaux d'exécution du système, le précédent et l'actuel, sur la sortie standard, séparés par un seul espace.

shutdown

shutdown arrête le système de façon sécurisée. Tous les utilisateurs connectés sont avertis que le système va s'arrêter, et la connexion est bloquée.

sulogin

sulogin est invoqué par init quand le système passe en mode mono−utilisateur (ceci est réalisé par une entrée dans /etc/inittab). init tente également d'exécuter sulogin quand l'option −b lui est passée par le gestionnaire de démarrage (comme LILO).

telinit

telinit envoie les signaux appropriés à init, lui indiquant le niveau d'exécution sous lequel passer.

Installer Tar

Installation de Tar

Si vous voulez pouvoir utiliser directement des fichiers bzip2 avec tar, utilisez le patch tar disponible sur le site FTP de LFS. Ce patch ajoutera l'option –y à tar, qui fonctionne de la même manière que l'option –z pour tar (que vous pouvez utiliser pour les fichiers gzip).

Appliquez le patch en lançant la commande suivante:

```
cd src &&
patch –i ../../gnutarpatch.txt &&
cd ..
```

Installez Tar en lançant les commandes suivantes depuis le répertoire racine:

```
./configure --prefix=/usr --libexecdir=/usr/bin  &&
make &&
make install &&
mv /usr/bin/tar /bin
```

Contenu

Le package tar contient les programmes tar et rmt.

Description

tar

tar est un programme d'archivage conçu pour stocker et extraire des fichiers d'une archive au format d'un fichier tar.

---

rmt

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

Installer Textutils

Installation de Textutils

Installez Textutils en lançant les commandes suivantes:

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

Contenu

Le paquet Textutils contient les programmes cat, cksum, comm, split, cut, expand, fmt, fold, head, join, md5sum, nl, od, paste, pr, ptx, sort, split, sum, tac, tail, tr, tsort, unexpand, uniq et wc.

Description

cat

cat concatène le(s) fichier(s) ou l'entrée standard vers la sortie standard.

---

cksum

cksum affiche la somme de contrôle CRC et le nombre d'octets de chaque fichier spécifié.

---

comm

comm compare deux fichiers triés ligne à ligne.

---

csplit

cplit découpe un fichier en morceaux séparés par un(des) motif(s) dans des fichiers xx01, xx02, ..., et affiche le nombre d'octets de chaque morceau sur la sortie standard.

---

cut

cut écrit les parties sélectionnées des lignes des fichiers spécifiés sur la sortie standard.

expand

expand convertit les tabulations dans les fichiers en espaces, en écrivant sur la sortie standard.

fmt

fmt reformate chaque paragraphe dans le(s) fichier(s) spécifié(s), en écrivant sur la sortie standard.

fold

fold emballe les lignes d'entrée de chaque fichier spécifié (entrée standard par défaut), en écrivant sur la sortie standard.

head

head affiche les xx (10 par défaut) premières lignes de chaque fichier spécifié sur la sortie standard.

join

join fusionne les lignes identiques de deux fichiers dans un fichier commun.

md5sum

md5sum affiche ou vérifie les sommes de contrôle MD5.

nl

nl écrit chaque fichier spécifié sur la sortie standard, en y ajoutant les numéros de ligne.

od

od écrit une représentation non−ambiguë, en octal par défaut, d'un fichier spécifié, sur la sortie standard.

paste

paste écrit des lignes, d'une manière consistant à prendre séquentiellement les lignes correspondantes de chaque fichier spécifié, séparées par des tabulations, sur la sortie standard.

---

pr

pr met en pages ou en colonnes des fichiers pour impression.

---

ptx

ptx produit un index de permutation du contenu d'un fichier.

---

sort

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

---

split

split extrait des morceaux de taille fixe d'un fichier d'entrée dans PREFIXaa, PREFIXab, ...

---

sum

sum écrit la somme de contrôle et le nombre de blocs de chaque fichier spécifié.

---

tac

tac écrit chaque fichier spécifié sur la sortie standard, la dernière ligne en premier.

---

tail

tail écrit les xx (10 par défaut) dernières lignes de chaque fichier spécifié sur la sortie standard.

---

tr

tr convertit, répète et/ou supprime des caractères de l'entrée standard, en écrivant sur la sortie standard.

---

tsort

tsort écrit des listes totalement ordonnées en accord avec l'ordonnancement partiel dans les fichiers spécifiés.

---

unexpand

unexpand convertit les espaces dans chaque fichier en tabulations, en écrivant sur la sortie standard.

---

uniq

uniq supprime toutes les lignes identiques successives sauf une depuis des fichiers ou l'entrée standard, et écrit dans des fichiers ou la sortie standard.

---

wc

wc affiche le nombre de lignes, mots et octets de chaque fichier spécifié, et une ligne de totalisation si plus d'un fichier est spécifié.

Installer Utillinux

Installation de Util–Linux

Installez Util–Linux en lançant les commandes suivantes:

```
sed –e s/HAVE_SLN=no/HAVE_SLN=yes/ \
   –e s/HAVE_TSORT=no/HAVE_TSORT=yes/  \
   MCONFIG > MCONFIG~  &&
mv MCONFIG~ MCONFIG &&
./configure &&
make &&
make install
```

Explication des commandes

`HAVE_SLN=yes:` Nous ne construisons pas ce programme parce qu'il a déjà été installé par Glibc.

`HAVE_TSORT=yes:` Nous ne construisons pas ce programme non plus car il a déjà été installé par Textutils.

Contenu

Le paquet Util–linux contient les programmes arch, dmesg, kill, more, mount, umount, agetty, blockdev, cfdisk, ctrlaltdel, elvtune, fdisk, fsck.minix, hwclock, kbdrate, losetup, mkfs, mkfs.bfs, mkfs.minix, mkswap, sfdisk, swapoff, swapon, cal, chkdupexe, col, colcrt, colrm, column, cytune, ddate, fdformat, getopt, hexdump, ipcrm, ipcs, logger, look, mcookie, namei, rename, renice, rev, script, setfdprm, setsid, setterm, ul, whereis, write, ramsize, rdev, readprofile, rootflags, swapdev, tunelp et vidmode.

Description

arch

arch affiche l'architecture de la machine.

---

dmesg

dmesg est utilisé pour examiner ou vérifier la mémoire tampon circulaire du noyau (messages de démarrage du noyau).

---

kill

kill envoie un signal donné au processus spécifié.

---

more

more est un filtre pour paginer le texte écran par écran.

---

mount

mount monte un système de fichiers depuis un périphérique dans un répertoire (point de montage).

---

umount

umount démonte un système de fichiers monté.

---

agetty

agetty ouvre un port tty, demande un nom de connexion et appelle la commande /bin/login.

---

blockdev

blockdev vous permet d'appeler ioctls sur un périphérique de type bloc depuis la ligne de commande.

---

cfdisk

cfdisk est un manipulateur de table de partitions de disque basé sur libncurses.

---

ctrlaltdel

ctrlaltdel positionne la fonction associée à la combinaison de touche CTRL+ALT+DEL (remise à zéro logicielle ou matérielle).

---

elvtune

elvtune permet d'optimiser l'augmentation d'entrées/sorties par file de base de périphérique bloc.

---

fdisk

fdisk est un manipulateur de table de partitions de disque.

fsck.minix

fsck.minix effectue une vérification de cohérence pour le système de fichiers Linux MINIX.

hwclock

hwclock interroge et positionne l'horloge matérielle (appelée aussi horloge RTC ou BIOS).

kbdrate

kbdrate réinitialise le taux de répétition du clavier et son délai.

losetup

losetup configure et contrôle les périphériques de boucle.

mkfs

mkfs construit un système de fichiers Linux sur un périphérique, en général une partition de disque dur.

mkfs.bfs

mkfs.bfs crée un système de fichiers SCO bfs sur un périphérique, en général une partition de disque dur.

mkfs.minix

mkfs.minix crée un système de fichiers Linux MINIX sur un périphérique, en général une partition de disque dur.

mkswap

mkswap met en place une zone d'échange Linux sur un périphérique ou dans un fichier.

sfdisk

sfdisk est un manipulateur de table de partitions de disque.

---

swapoff

swapoff désactive les échanges de page mémoire pour les périphériques et les fichiers.

---

swapon

swapon active les échanges de page mémoire pour les périphériques et les fichiers.

---

cal

cal affiche un calendrier simple.

---

chkdupexe

chkdupexe trouve les exécutables en double.

---

col

col filtre les fins de ligne inversées de son entrée.

---

colcrt

colcrt filtre les sorties de nroff pour une prévisualisation via CRT.

---

colrm

colrm supprime les colonnes d'un fichier.

---

column

column met des listes en colonne.

---

cytune

cytune interroge et modifie le seuil d'interruption pour le pilote Cyclades.

ddate

ddate convertit les dates en système Grégorien en dates en système Discordien.

fdformat

fdformat réalise le formatage de bas niveau d'une disquette.

getopt

getopt sépare les options de commande, de la même manière que la commande C getopt.

hexdump

hexdump affiche les fichiers spécifiés, ou l'entrée standard, dans un format spécifié par l'utilisateur (ascii, décimal, hexadécimal, octal).

ipcrm

ipcrm supprime une ressource spécifiée.

ipcs

ipcs fournit des informations sur les utilitaires ipc.

logger

logger crée des entrées dans le journal système.

look

look affiche les lignes commençant par une chaîne donnée.

mcookie

mcookie génère des nombres magiques pour xauth.

---

namei

namei suit un chemin jusqu'à trouver un point terminal.

---

rename

rename renomme les fichiers.

---

renice

renice modifie la priorité des processus en exécution.

---

rev

rev inverse les lignes d'un fichier

---

script

script enregistre une session de terminal dans un fichier texte.

---

setfdprm

setfdprm positionne les paramètres fournis par l'utilisateur pour la disquette.

---

setsid

setsid lance les programmes dans une nouvelle session.

---

setterm

setterm positionne les attributs d'un terminal.

---

ul

ul lit un fichier et traduit les occurrences de blancs soulignés en séquence qui indiquent le soulignement sur le terminal utilisé.

---

whereis

whereis localise un binaire, un fichier source et une page de manuel pour une commande.

---

write

write envoie un message à un autre utilisateur.

---

ramsize

ramsize interroge et positionne la taille du disque mémoire.

---

rdev

rdev interroge et positionne le périphérique racine image, le périphérique d'échange, la taille du disque mémoire ou le mode vidéo.

---

readprofile

readprofile lit les informations de profil du noyau.

---

rootflags

rootflags interroge et positionne des informations supplémentaires utilisées lors du montage de la racine.

---

swapdev

swapdev interroge et positionne le périphérique d'échange.

---

tunelp

tunelp positionne différents paramètres pour le périphérique lp.

---

vidmode

vidmode interroge et positionne le mode vidéo.

Suppression des anciens fichiers de la bibliothèque NSS

Si vous avez copié les fichiers de la bibliothèque NSS depuis votre système Linux habituel dans le système LFS (parce que votre système habituel fonctionne avec glibc−2.0), il est temps de les supprimer maintenant, en lançant:

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

Configurer les logiciels essentiels

Maintenant que tous les logiciels sont installés, tout ce qui nous reste à faire pour que certains programmes fonctionnent correctement est de créer leurs fichiers de configuration.

Configurer Vim

Par défaut, Vim fonctionne en mode compatible avec vi. Certains peuvent l'apprécier, mais je préfère nettement faire tourner vim en mode vim (autrement, je n'aurais pas inclu Vim dans ce livre, mais le Vi original). Créez le fichier `/root/.vimrc` en lançant ce qui suit:

```
cat > /root/.vimrc << "EOF"
" Debut /root/.vimrc

set nocompatible
set bs=2

" Fin /root/.vimrc
EOF
```

Configurer Glibc

Nous avons besoin de créer le fichier /etc/nsswitch. Bien que glibc fournisse des valeurs par défaut quand ce fichier est absent ou corrompu, ces valeurs ne fonctionnent pas correctement avec le réseau dont nous traiterons dans le prochain chapitre. De plus, notre zone horaire doit être configurée.

Créez un nouveau fichier `/etc/nsswitch.conf` en lançant ce qui suit:

```
cat > /etc/nsswitch.conf << "EOF"
# Debut /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 /etc/nsswitch.conf
EOF

Exécutez le script **tzselect** et répondez aux questions à propos de votre zone horaire. Une fois ceci fait, le script vous fournira l'emplacement du fichier de zone horaire dont vous aurez besoin.

Créez le lien symbolique `/etc/localtime` en lançant:

```
cd /etc &&
rm localtime &&
ln -s ../usr/share/zoneinfo/<resultat de tzselect>  localtime
```

La sortie de tzselect peut ressembler à *EST5EDT* ou *Canada/Eastern*.

Le lien symbolique que vous allez créer avec cette information sera:

```
ln -s ../usr/share/zoneinfo/EST5EDT localtime
```

Ou:

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

Configurer le Chargeur Dynamique

Par défaut, le chargeur dynamique recherche les bibliothèques dynamiques dans quelques répertoires par défaut; normalement, le fichier `/etc/ld.so.conf` est donc inutile, à moins que vous n'ayez d'autres répertoires dans lesquels vous voudriez que le système fasse ses recherches. Le répertoire `/usr/local/lib` n'est, par défaut, pas scruté pour rechercher des bibliothèques dynamiques; nous vous proposons de créer ce fichier de configuration pour permettre à ldso de scruter ce répertoire. Cela vous évitera des déconvenues après l'installation de nouveaux logiciels dans l'arborescence /usr/local.

Créez un nouveau fichier `/etc/ld.so.conf` en lançant ce qui suit:

```
cat > /etc/ld.so.conf << "EOF"
# Debut /etc/ld.so.conf

/lib
```

```
/usr/lib
/usr/local/lib

# Fin /etc/ld.so.conf
EOF
```

Bien qu'il ne soit pas nécessaire d'ajouter les répertoires `/lib` et `/usr/lib`, cela ne fait pas de mal. De cette manière, vous voyez vraiment ce qui est recherché, et n'avez pas à vous remémorer les chemins de recherche par défaut, si vous ne le voulez pas.

Configurer Lilo

Nous n'allons pas créer un fichier de configuration de lilo à partir de rien, mais nous utiliserons le fichier de votre système Linux habituel. Ce fichier est différent pour chaque machine, et je ne peux donc pas le créer ici. Comme vous voudrez avoir les mêmes options pour lilo que ce que vous avez en utilisant votre système Linux habituel, vous devrez créer le fichier exactement comme sur le système habituel.

Copiez le fichier de configuration de Lilo et les images du noyau utilisées par Lilo en lançant les commandes suivantes depuis un shell sur votre système Linux habituel. Ne lancez pas ces commandes depuis votre shell chroot.

```
cp /etc/lilo.conf $LFS/etc
cp /boot/<images noyau> $LFS/boot
```

Avant d'exécuter la seconde commande, vous devrez connaître les noms des images du noyau. Vous ne pouvez pas simplement copier tous les fichiers du répertoire /boot. Le fichier /etc/lilo.conf contient les noms des images du noyau que vous utilisez. Ouvrez le fichier et regardez les lignes ressemblant à ceci:

```
image=/boot/vmlinuz
```

Recherchez toutes les variables *image*, car leur valeur représentent le nom et la localisation des fichiers image. Ces fichiers sont en général sous /boot, mais il peuvent également être dans d'autres répertoires, selon les conventions de votre distribution.

Configurer Sysklogd

Créez un nouveau fichier `/etc/syslog.conf` en lançant ce qui suit:

```
cat > /etc/syslog.conf << "EOF"
# Debut /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 /etc/syslog.conf
EOF
```

Configurer la Suite Masquage des Mots de passe (Shadow passwords)

Ce paquet contient les utilitaires pour modifier les mots de passe des utilisateurs, ajouter de nouveaux groupes/utilisateurs, détruire des groupes/utilisateurs, etc. Je ne vais pas vous expliquer ce que signifie « masquage des mots de passe » (password shadowing). Vous pouvez lire tout ce qui concerne cela dans le fichier doc/HOWTO dans l'arborescence source de la suite masquage des mots de passe déballée. Il y a une chose que vous devez garder à l'esprit, si vous décidez d'utiliser le support des mots de passe masqués, c'est que les programmes qui nécessitent la vérification des mots de passe (par exemple, xdm, les démons ftp, les démons pop3, etc.) doivent être « compatibles avec le masquage » (shadow–compliant), c'est–à–dire qu'ils doivent pouvoir travailler avec des mots de passe masqués.

Les mots de passe masqués ne sont pas activés par défaut. Se contenter d'installer la suite masquage des mots de passe n'active pas les mots de passe masqués.

C'est maintenant une bonne occasion de lire le chapitre 5 du fichier doc/HOWTO. Vous pouvez y lire comment activer les mots de passe masqués, comment tester si le masquage fonctionne et, sinon, comment le désactiver de nouveau.

La documentation fait allusion à la création de npasswd et nshadow après que vous ayiez exécuté pwconv. Ceci est une erreur dans la documentation. Ces deux fichiers seront créés. Après avoir exécuté pwconv, /etc/passwd ne contiendra plus de mots de passe, mais /etc/shadow oui. Vous n'avez pas besoin de renommer les fichiers npasswd et nshadow vous–mêmes.

Configurer Sysvinit

Créez un nouveau fichier `/etc/inittab` en lançant ce qui suit:

```
cat > /etc/inittab << "EOF"
# Debut /etc/inittab

id:3:initdefault:

si::sysinit:/etc/init.d/rcS
```

```
l0:0:wait:/etc/init.d/rc 0
l1:S1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6

ft:06:respawn:/sbin/sulogin

ca:12345:ctrlaltdel:/sbin/shutdown –t1 –a –r now

su:S1: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

# Fin /etc/inittab
EOF
```

Créer les fichiers /var/run/utmp, /var/log/wtmp et /var/log/btmp

Les programmes comme login, shutdown, uptime et d'autres cherchent à écrire dans /var/run/utmp, /var/log/btmp et /var/log/wtmp. Ces fichiers contiennent des informations sur qui est actuellement connecté. Ils contiennent également des informations sur l'heure des derniers démarrage et arrêt de l'ordinateur, et un enregistrement des tentatives de connexion ayant échoué.

Créez ces fichiers avec leurs permissions adéquates en lançant les commandes suivantes:

```
touch /var/run/utmp /var/log/wtmp /var/log/btmp
/var/log/lastlog &&
chmod 644 /var/run/utmp /var/log/wtmp /var/log/btmp
/var/log/lastlog
```

Créer le mot de passe de root

Choisissez un mot de passe pour l'utilisateur root et créez–le en lançant la commande suivante:

```
passwd root
```

Chapitre 7. Création des scripts de démarrage du système

Introduction

Ce chapitre vous permettra de créer les scripts nécessaires lors du démarrage du système. Ces scripts permettent notamment de remonter en mode lecture–écriture le système de fichiers racine qui avait été monté en mode lecture seule par le noyau, d'activer la (les) partition(s) de swap, de lancer un contrôle du système de fichiers racine pour s'assurer qu'il est intact et de lancer les démons utilisés par le système.

Création des répertoires

Nous avons besoin pour commencer de créer quelques répertoires supplémentaires qui seront utilisés par les scripts de boot. Créer ces répertoires avec les commandes suivantes:

```
cd /etc &&
mkdir sysconfig rc0.d rc1.d rc2.d rc3.d &&
mkdir rc4.d rc5.d rc6.d init.d rcS.d &&
cd init.d
```

Création du script rc

Le premier script principal de démarrage est le script `/etc/init.d/rc`. Créez un nouveau fichier `/etc/init.d/rc` contenant ce qui suit:

```
cat > rc << "EOF"
#!/bin/sh
# Begin /etc/init.d/rc
#
# par Jason Pearce – jason.pearce@linux.org
# Modifie par Gerard Beekmans – gerard@linuxfromscratch.org
# print_error_msg base sur une idee de Simon Perreault – nomis80@yahoo.com

#
# Inclusion des fonctions declarees dans le fichier /etc/init.d/functions
#

source /etc/init.d/functions

#
# La fonction print_error_msg affiche un message d'erreur lorsqu'une
# erreur imprevue se produit et n'a pas ete interceptee pour une quelconque
# raison par un appel a evaluate_retval ou un quelconque controle d'erreur
#

print_error_msg()
{

      echo
      $FAILURE
echo –n "Vous ne devriez pas lire ce message d'erreur. Il "
echo "signifie qu'une erreur imprevue s'est produite et que "
echo –n "le sous–script $i s'est termine avec la valeur de "
echo "retour $error_value pour une raison inconnue. Si vous "
echo –n "etes capable de trouver la cause de cette erreur "
echo "dans l'un des fichiers fournis dans ce livre, merci "
echo –n "de nous en informer a lfs–discuss@linuxfromscratch.org"
      $NORMAL
      echo
      echo
      echo "Appuyer sur une touche pour continuer..."
      read
}

#
# Si vous decommentez la variable debug ci–dessous, aucun des scripts
# ne sera execute, seul le nom du script et ses parametres seront
# affiches a l'ecran afin de voir comment les scripts sont appeles par rc."
#
```

```
# Decommenter la ligne suivante pour debogage
# debug=echo

#
# Demarrage de script ou programme
#
startup() {

$debug $*

}

#
# Ignorer CTRL−C seulement dans ce shell, de telle sorte que nous
# puissions interrompre les sous−processus.
#

trap ":" INT QUIT TSTP

#
# Maintenant, determinons les niveaux d'execution actuel et precedent.
# La variable $RUNLEVEL est renseignee par init pour tous ses fils.
# Ce script s'execute en tant que fils d'init.
#

runlevel=$RUNLEVEL

#
# Recuperation du premier argument. Positionner le nouveau niveau
# d'execution en accord avec cet argument. Si aucun niveau d'execution
# n'a ete passe a ce script, nous ne changeons pas les niveaux d'execution.
#

[ "$1" != "" ] && runlevel=$1
if [ "$runlevel" = "" ]
then
        echo "Usage: $0 <runlevel>" >&2
        exit 1
fi

#
# La meme chose pour $PREVLEVEL (cf. ci−dessus $RUNLEVEL). previous sera
# renseigne avec le niveau d'execution precedent. Si $PREVLEVEL n'est pas
# positionne, cela signifie qu'il n'y avait pas de niveau d'execution
# precedent et nous mettrons previous à N.
#

previous=$PREVLEVEL
[ "$previous" = "" ] && previous=N
```

```
export runlevel previous

#
# Y a−t'il un repertoire rc pour le nouveau niveau d'execution ?
#

if [ −d /etc/rc$runlevel.d ]

then

#
# Si rc existe pour le nouveau niveau d'execution , recuperer d'abord
# tous les scripts K* de ce niveau d'execution.
#

      if [ $previous != N ]
      then
            for i in /etc/rc$runlevel.d/K*
            do
            [ ! −f $i ] && continue

#
# La variable suffix contiendra le nom du script sans le Kxxx initial
#

                  suffix=${i#/etc/rc$runlevel.d/K[0−9][0−9][0−9]}
#
# S'il y a un script de demarrage pour ce script K dans le niveau d'execution
# precedent, determinons son chemin d'acces complet
#
         previous_start=/etc/rc$previous.d/S[0−9][0−9][0−9]$suffix
#
# S'il n'y a pas de niveau d'execution precedent, il se peut que quelque chose
# avait ete demarre dans rcS.d (niveau sysinit) aussi nous determinerons aussi
# le chemin pour cette possibilite.
#

                  sysinit_start=/etc/rcS.d/S[0−9][0−9][0−9]$suffix

#
# Arreter le service s'il y a un script de demarrage dans le niveau
# precedent ou dans le niveau sysinit. Si previous_start ou sysinit_start
# n'existe pas, la commande 'continue' est lancee, ce qui conduit le
# script a interrompre cette iteration de la boucle for et a poursuivre
# avec l'iteration suivante.
# Cela revient a ce qu'il n'execute pas les commandes apres les deux
# lignes suivantes et recommence au debut de cette boucle for. Cf.
# man bash pour plus d'information a ce sujet.
#

                  [ ! −f $previous_start ] &&
```

```
                [ ! −f $sysinit_start ] && continue

#
# Si nous trouvons previous_start ou sysinit_start, lancer le
# script K
#

                startup $i stop
                error_value=$?
#
# Si la valeur de retour du script n'est pas 0, quelque chose est alle
# de travers avec le controle d'erreur inclus dans le script. La fonction
# print_error_msg sera appelee et le message plus la valeur de retour du
# script K seront affiches a l'ecran.
#

                if [ $error_value != 0 ]
                then
                        print_error_msg
                fi

        done
    fi

#
# Maintenant nous lançons les scripts START pour ce niveau d'execution.
#

    for i in /etc/rc$runlevel.d/S*
    do
        [ ! −f $i ] && continue

        if [ $previous != N ]
    then
#
# Trouver le script de demarrage dans le niveau d'execution precédant
# et stopper le script dans ce niveau d'execution.
#

                suffix=${i#/etc/rc$runlevel.d/S[0−9][0−9][0−9]}
                stop=/etc/rc$runlevel.d/K[0−9][0−9][0−9]$suffix
        previous_start=/etc/rc$previous.d/S[0−9][0−9][0−9]$suffix
#
# S'il y a un script de demarrage dans le niveau precedent et pas de script
# d'arret dans ce niveau, nous n'avons pas a relancer le service; interrompre
# cette iteration et demarrer la suivante.
#

                [ −f $previous_start ] && [ ! −f $stop ] &&
                continue
        fi
```

```
            case "$runlevel" in
                0|6)

#
# les niveaux 0 et 6 sont les niveaux d'arret (halt) et de redemarrage (reboot).
# Nous ne demarrons reellement rien si bien que nous faisons l'appel avec le
# parametre 'stop'.
#

                        startup $i stop
                        error_value=$?
#
# Si la valeur de retour du script n'est pas 0, quelque chose est alle
# de travers dans le controle d'erreur du script. La fonction
# print_error_msg sera appelee et le message plus la valeur de retour du
# script K seront affiches a l'ecran.
#

                        if [ $error_value != 0 ]
                        then
                              print_error_msg
                        fi
                        ;;
                *)
                        startup $i start
                        error_value=$?
#
# Si la valeur de retour du script n'est pas 0, quelque chose est alle
# de travers dans le controle d'erreur du script. La fonction
# print_error_msg sera appelee et le message plus la valeur de retour du
# script K seront affiches a l'ecran.
#

                        if [ $error_value != 0 ]
                        then
                              print_error_msg
                        fi
                        ;;
            esac
        done
fi

# End /etc/init.d/rc
EOF
```

Création du script rcS

Le second script principal de démarrage est le script `rcS`. Créez le nouveau fichier `/etc/init.d/rcS` contenant ce qui suit:

```
cat > rcS << "EOF"
#!/bin/sh
# Begin /etc/init.d/rcS

#
# Voir le script rc pour des commentaires plus detailles sur les
# constructions utilisees ici.
#

runlevel=S
prevlevel=N
umask 022
export runlevel prevlevel

trap ":" INT QUIT TSTP

#
# Recupere tous les scripts S de /etc/rcS.d et les execute dans le meme
# ordre
#

for i in /etc/rcS.d/S??*
do
      [ ! –f  "$i" ] && continue;
            $i start
done

# End /etc/init.d/rcS
EOF
```

Création du script functions

Créer un nouveau fichier `/etc/init.d/functions` contenant ce qui suit:

```
cat > functions << "EOF"
#!/bin/sh
# Begin /etc/init.d/functions

#
# Definition de quelques variables qui influencent la facon dont
# le texte sera affiche a l'ecran. La variable SET_COL fait demarrer
# le texte en colonne 70 (comme specifie par la variable COL).
# NORMAL affiche le texte en mode normal. SUCCESS affichera le texte en
# vert et FAILURE en rouge. Voir print_status pour plus de détails.
#

COL=70
SET_COL="echo –en \\033[${COL}G"
NORMAL="echo –en \\033[0;39m"
SUCCESS="echo –en \\033[1;32m"
FAILURE="echo –en \\033[1;31m"

#
# la fonction evaluate_retval evalue la valeur de retour du processus
# qui etait execute juste avant que la fonction ne soit appelee. Si cette
# valeur de retour valait 0, indiquant le succes, la fonction print_status
# est appelee avec le parametre 'success'. Dans les autres cas, la fonction
# print_status est appelee avec le parametre correspondant a l'erreur.
#

evaluate_retval()
{
      if [ $? = 0 ]
      then
            print_status success
      else
            print_status failure
      fi
}

#
# La fonction print_status affiche [ OK ] ou [FAILED] a l'ecran. OK
# utilisera la couleur definie par la variable SUCCESS et FAILED celle
# definie par la variable FAILURE. Les deux seront affichees a partir
# de la colonne definie par la variable COL.
#

print_status()
{
```

```
#
# Si aucun parametre n'est passe a la fonction print_status, afficher
# le mode d'emploi.
#

      if [ $# = 0 ]
      then
            echo "Usage: print_status {success|failure}"
            return 1
      fi

      case "$1" in
            success)
                  $SET_COL
                  echo −n "[  "
                  $SUCCESS
                  echo −n "OK"
                  $NORMAL
                  echo "  ]"
                  ;;
            failure)
                  $SET_COL
                  echo −n "["
                  $FAILURE
                  echo −n "FAILED"
                  $NORMAL
                  echo "]"
                  ;;
      esac

}

#
# la fonction loadproc lance un processus (souvent un demon)
# avec les controles d'erreur appropries
#

loadproc()
{

#
# Si aucun parametre n'est passe a la fonction print_status, afficher
# le mode d'emploi.
#

      if [ $# = 0 ]
      then
            echo "Usage: loadproc {program}"
            exit 1
      fi
```

```
#
# trouver le nom de base ('basename') du premier parametre (le nom du demon sans
# le chemin d'acces qui a ete donne : ainsi /usr/sbin/syslogd devient
# 'syslogd' une fois que basename a ete lance)
#

      base=$(/usr/bin/basename $1)

#
# la variable pidlist contiendra le resultat de la commande pidof.
# pidof essaiera de trouver le PID correspondant a une chaine de
# caracteres ($base dans le cas present).
#

      pidlist=$(/bin/pidof -o $$ -o $PPID -o %PPID -x $base)

      pid=""

      for apid in $pidlist
      do
            if [ -d /proc/$apid ]
            then
                  pid="$pid $apid"
            fi
      done
#
# Si la variable $pid contient quelque chose (provenant de la boucle loop
# precedente), cela signifie que le demon a deja ete lance.
#

      if [ ! -n "$pid" ]
      then
#
# Si $pid est vide, le demon n'a pas ete lance donc nous executons $*
# (tous les parametres du script sont passes a la fonction) puis nous
# controlons la valeur de retour
#
            $*
            evaluate_retval
      else
#
# La variable $pid n'etait pas vide, signifiant que le demon tournait
# deja. Nous affichons donc maintenant [FAILED]
#

            print_status failure
      fi

}
```

```
#
# La fonction killproc tue un processus en effectuant le controle
# d'erreur approprie
#

killproc()
{

#
# Si aucun parametre n'est passe a la fonction print_status, afficher
# le mode d'emploi.
#

        if [ $# = 0 ]
        then
                echo "Usage: killproc {program} [signal]"
                exit 1
        fi

#
# trouver le nom de base ('basename') du premier parametre (le nom du demon sans
# le chemin d'acces qui a ete donne : ainsi /usr/sbin/syslogd devient
# 'syslogd' une fois que basename a ete lance)
#

        base=$(/usr/bin/basename $1)

#
# Controle si nous avons passe un signal pour la commande kill (tel que
# -HUP, -TERM, -KILL, etc...) a la fonction (le second parametre). Si
# le second parametre n'a pas ete fourni, la variable nolevel est positionnee.
# Dans le cas contraire, la variable killlevel recoit la valeur de $2
# (le second parametre).
#

        if [ "$2" != "" ]
        then
                killlevel=-$2
        else
                nolevel=1
        fi

#
# la variable pidlist contiendra le resultat de la commande pidof.
# pidof essaiera de trouver le PID correspondant a une chaine de
# caracteres ($base dans le cas present).
#

        pidlist=$(/bin/pidof -o $$ -o $PPID -o %PPID -x $base)

        pid=""
```

```
        for apid in $pidlist
        do
                if [ −d /proc/$apid ]
                then
                        pid="$pid $apid"
                fi
        done

#
# Si $pid contient quelque chose provenant de la boucle precedente, cela
# signifie qu'on a trouve un ou plusieurs PID appartenant au processus
# qu'il faut 'killer'.
#
        if [ −n "$pid" ]
        then
#
# Si aucun niveau de kill n'a ete indique, on va d'abord essayer −TERM
# puis on va attendre 2 secondes pour laisser le temps à kill de finir
# son travail.
#
                if [ "$nolevel" = 1 ]
                then
                        /bin/kill −TERM $pid
/usr/bin/sleep 2
#
# Si apres un coup de −TERM le PID existe toujours, on va essayer de
# le 'killer' avec un −KILL, puis on va encore attendre 2 secondes
# pour laisser le temps a kill de finir son travail.
#

                        if ps h $pid >/dev/null 2>&1
                        then
                                /bin/kill −KILL $pid
/usr/bin/sleep 2
                        fi
                        /bin/ps h $pid >/dev/null 2>&1
                        if [ $? = 0 ]
                        then
#
# S'il existe encore apres le −KILL, cela veut dire qu'il existe une raison
# pour laquelle on ne pourra pas tuer le processus : on va donc afficher
# [FAILED]
#
                                print_status failure
                        else
#
# S'il a ete tue, on supprime un eventuel fichier PID qui subsisterait
# dans /var/run puis on affiche [  OK  ]
#
                                /bin/rm −f /var/run/$base.pid
```

```
                        print_status success
                    fi
                else
#
# Un niveau de kill a ete fourni. On execute donc kill avec le signal
# en question puis on attend 2 secondes pour permettre a kill de finir
# son boulot.
#
                    /bin/kill $killlevel $pid
/usr/bin/sleep 2
                    /bin/ps h $pid >/dev/null 2>&1
                    if [ $? = 0 ]
                    then
#
# Si la valeur de retour de ps vaut 0, cela signifie que cette commande
# s'est executee avec succes et par consequent que le PID existe toujours.
# Donc le processus n'a pas ete tue avec le signal fourni. Afficher [FAILED]
#
                        print_status failure
                    else
#
# Si la valeur de retour valait 1 ou plus, cela signifie que le PID
# n'existe plus desormais, donc qu'il a ete tue avec succes. Supprimer
# un eventuel fichier PID residuel et afficher [  OK  ]
#
                        /bin/rm -f /var/run/$base.pid
                        print_status success
                    fi
                fi
        else
#
# Le fichier PID n'existe pas, il n'est donc pas necessaire d'essayer
# de le tuer. Afficher [FAILED]
#
                print_status failure
        fi
}

#
# La fonction reloadproc envoie un signal au demon en lui disant de
# recharger son fichier de configuration. Elle est presque identique
# a la fonction killproc si ce n'est qu'elle n'essaie pas de tuer le
# processus avec un signal -KILL (c'est-a-dire -9).
#

reloadproc()
{

#
# Si aucun parametre n'est passe a la fonction print_status, afficher
# le mode d'emploi.
```

```
#

      if [ $# = 0 ]
      then
            echo "Usage: reloadproc {program} [signal]"
            exit 1
      fi

#
# trouver le nom de base ('basename') du premier parametre (le nom du demon sans
# le chemin d'acces qui a ete donne : ainsi /usr/sbin/syslogd devient
# 'syslogd' une fois que basename a ete lance)
#

      base=$(/usr/bin/basename $1)

#
# Controle si nous avons passe un signal pour la commande kill (tel que
# −HUP, −TERM, −KILL, etc...) a la fonction (le second parametre). Si
# le second parametre n'a pas ete fourni, la variable nolevel est positionnee.
# Dans le cas contraire, la variable killlevel recoit la valeur de $2
# (le second parametre).
#


      if [ −n "$2" ]
      then
            killlevel=−$2
      else
            nolevel=1
      fi

#
# la variable pidlist contiendra le resultat de la commande pidof.
# pidof essaiera de trouver le PID correspondant à une chaine de
# caracteres ($base dans le cas present).
#

      pidlist=$(/bin/pidof −o $$ −o $PPID −o %PPID −x $base)

      pid=""

      for apid in $pidlist
      do
            if [ −d /proc/$apid ]
            then
                  pid="$pid $apid"
            fi
      done

#
```

```
# Si $pid contient quelque chose provenant de la boucle precedente, cela
# signifie qu'on a trouve un ou plusieurs PID appartenant au processus
# qu'il faut 'killer'.
#

      if [ –n "$pid" ]
      then

#
# Si aucun niveau de kill n'a ete indique, on va utiliser le signal de
# rechargement par defaut : SIGHUP
#

            if [ "$nolevel" = 1 ]
            then
                  /bin/kill –SIGHUP $pid
                  evaluate_retval
            else
#
# Sinon, on utilisera le signal fourni
#

                  /bin/kill $killlevel $pid
                  evaluate_retval
            fi
      else
#
# Si $pid est vide, aucun PID appartenant au processus n'a ete trouve
# et on affiche [FAILED]
#

            print_status failure
      fi
}

#
# La fonction statusproc essaiera de savoir si un processus s'execute
# ou non.
#

statusproc()
{

#
# Si aucun parametre n'a ete passe a la fonction print_status, afficher
# le mode d'emploi
#

      if [ $# = 0 ]
      then
            echo "Usage: status {program}"
```

```
            return 1
        fi

#
# $pid contiendra la liste des PIDs appartenant à un processus
#

        pid=$(/bin/pidof -o $$ -o $PPID -o %PPID -x $1)
        if [ -n "$pid" ]
        then
#
# Si $pid contient quelque chose, le processus est en cours d'execution :
# on affiche le contenu de la variable $pid
#
            echo "$1 en cours d'execution, sous le PID n° $pid"
            return 0
        fi

#
# Si $pid ne contient rien, on regarde s'il existe un fichier PID et
# on en informe eventuellement l'utilisateur
#

        if [ -f /var/run/$1.pid ]
        then
            pid=$(/usr/bin/head -1 /var/run/$1.pid)
            if [ -n "$pid" ]
            then
                echo "$1 n'est plus en cours d'execution mais /var/run/$1.pid existe"
                return 1
            fi
        else
            echo "$1 n'est plus en cours d'execution"
        fi

}

# End /etc/init.d/functions
EOF
```

Création du script checkfs

Créer un nouveau fichier `/etc/init.d/checkfs` contenant ce qui suit:

```
cat > checkfs << "EOF"
#!/bin/sh
# Begin /etc/init.d/checkfs

#
# Inclure les fonctions declarees dans le fichier /etc/init.d/functions
#

source /etc/init.d/functions

#
# Activer toutes les partitions de swap declarees dans le fichier /etc/fstab
#

echo −n "Activation du swap..."
/sbin/swapon −a
evaluate_retval

#
# Si le fichier /fastboot existe, on ne lance pas le contrôle des partitions
#

if [ −f /fastboot ]
then
      echo "Boot accelere, pas de controle du systeme de fichiers"
else

#
# Montage de la partition racine en lecture seule (juste au cas ou le
# noyau la monterait en lecture−ecriture alors que nous ne voulons pas
# lancer fsck sur une partition montee en lecture−ecriture).
#

      /bin/mount −n −o remount,ro /
      if [ $? = 0 ]
      then

#
# Si le fichier /forcefsck existe, nous voulons forcer un controle de
# partition meme si la partition avait ete demontee proprement la
# derniere fois
#

            if [ −f /forcefsck ]
            then
```

```
                echo −n "/forcefsck existe, controle force"
                echo "controle du systeme de fichiers"
                force="−f"
        else
                force=""
        fi

#
# Controle tous les systemes de fichiers mentionnes dans /etc/fstab
# qui possedent un champ fs_passno positionne a 1 ou 2 (le 6eme champ.
# cf. man fstab pour plus d'informations)
#

        echo "controle des systemes de fichiers..."
        /sbin/fsck $force −a −A −C −T

#
# Si quelque chose se passe mal lors du controle de l'une des
# partitions, fsck se terminera avec une valeur de retour superieure
# a 1. Si c'est le cas, on lance sulogin de façon a ce que vous
# puissiez reparer manuellement les degats.
#

        if [ $? −gt 1 ]
        then
                $FAILURE
                echo
                echo −n "echec de fsck . Reparez vos systemes "
                echo "de fichiers en lancant manuellement /sbin/fsck "
                echo "sans l'option −a"
                echo
                echo −n "Notez bien que le systeme de fichiers "
echo "racine est actuellement monte en lecture seule."
                echo
                echo −n "Je vais maintenant lancer sulogin. Lorsque  "
                echo "vous vous deloguerez, je redemarrerai votre systeme."
                echo
                $NORMAL
                /sbin/sulogin
                /sbin/reboot −f
        else
                print_status success
        fi

    else

#
# Si le remontage en mode lecture seule ne fonctionne pas, on interrompt
# fsck et on affiche une erreur.
#
```

```
            echo –n "Impossible de controler le systeme de fichiers racine "
echo "car il n'a pas pu etre monte en mode lecture seule."
        fi
fi

# End /etc/init.d/checkfs
EOF
```

Création du script halt

Créez un nouveau fichier `/etc/init.d/halt` contenant ce qui suit:

```
cat > halt << "EOF"
#!/bin/sh
# Begin /etc/init.d/halt

#
# Appel de halt. Cf. man halt pour la signification des
# parametres
#

/sbin/halt –d –f –i –p

# End /etc/init.d/halt
EOF
```

Création du script loadkeys

Créez ce fichier seulement si vous ne possédez pas un clavier standard US à 101 touches. Dans ce cas, créez un nouveau fichier `/etc/init.d/loadkeys` contenant ce qui suit:

```
cat > loadkeys << "EOF"
#!/bin/sh
# Begin /etc/init.d/loadkeys

#
# Inclure les fonctions declarees dans le fichier /etc/init.d/functions
#

source /etc/init.d/functions

#
# Charger le fichier keymap
#

echo –n "Chargement de la keymap..."
/usr/bin/loadkeys –d >/dev/null
evaluate_retval

# End /etc/init.d/loadkeys
EOF
```

Création du script mountfs

Créez un nouveau fichier `/etc/init.d/mountfs` contenant ce qui suit:

```
cat > mountfs << "EOF"
#!/bin/sh
# Begin /etc/init.d/mountfs

#
# Inclure les fonctions declarees dans le fichier /etc/init.d/functions
#

source /etc/init.d/functions

case "$1" in
   start)

      #
# Remonte la partition racine en mode lecture−ecriture. −n
# indique a mount de ne pas ecrire le fichier /etc/mtab
      # (car il ne pourra pas le faire. La partition racine
# est tres probablement montee en mode lecture seule).
      #

      echo −n "Remontage du systeme de fichiers racine en mode lecture−ecriture..."
      /bin/mount −n −o remount,rw /
      evaluate_retval

      #
# D'abord vider le fichier /etc/mtab. Puis remonter la partition
# racine une nouvelle fois en mode lecture−ecriture mais en passant
# −f a mount. De cette facon, mount fait tout sauf le montage lui−meme.
# Ceci est necessaire pour ecrire le fichier mtab qui contient la
# liste des systemes de fichiers actuellement montes.
#

      echo > /etc/mtab
      /bin/mount −f −o remount,rw /

      #
# Supprimer les eventuels fichiers /fastboot et /forcefsck. Ils
# ne doivent servir qu'au checkfs du reboot suivant qui vient
# juste d'avoir lieu. Si vous souhaitez un boot accelere ou au
# contraire provoquer le controle des systemes de fichiers une
# nouvelle fois, vous devrez recreer l'un de ces fichiers.
      #

      /bin/rm −f /fastboot /forcefsck
```

```
        #
# Balayer le fichier /etc/fstab et monter tous les systemes de
# fichiers qui ne possedent pas l'option noauto positionnee dans
# le champ fs_mntops (le 4eme champ. cf. man fstab pour plus
# d'informations).
        #

        echo −n "Montage des autres systemes de fichiers..."
        /bin/mount −a
        evaluate_retval
        ;;

    stop)

        #
        # Desactiver toutes les partitions de swap
        #

        echo −n "Desactivation du swap..."
        /sbin/swapoff −a
        evaluate_retval

        #
# Et demontage de tous les systemes de fichiers, montage
# du systeme de fichiers racine en lecture seule (tous
# sont demontes mais comme la racine ne peut pas l'etre
# a ce moment, mount la remontera automatiquement en
# lecture seule, ce que est suppose se passer. De cette
# facon, aucune donnee ne pourra plus etre inscrite sur
# le disque).
        #

        echo −n "Demontage des systemes de fichiers..."
        /bin/umount −a −r
        evaluate_retval
        ;;

    *)
        echo "Usage: $0 {start|stop}"
        exit 1
    ;;
esac

# End /etc/init.d/mountfs
EOF
```

Création du script de redémarrage (reboot)

Créez un nouveau fichier `/etc/init.d/reboot` contenant ce qui suit:

```
cat > reboot << "EOF"
#!/bin/sh
# Begin /etc/init.d/reboot

#
# Appelle reboot. Cf. man halt pour la signification des parametres.
#


echo "Redemarrage du systeme en cours..."

/sbin/reboot –d –f –i

# End /etc/init.d/reboot
EOF
```

Création du script sendsignals

Créez un nouveau fichier `/etc/init.d/sendsignals` contenant ce qui suit:

```
cat > sendsignals << "EOF"
#!/bin/sh
# Begin /etc/init.d/sendsignals

#
# Inclusion des fonctions declarees dans le fichier /etc/init.d/functions
#

source /etc/init.d/functions

#
# Envoyer a tous les processus restant le signal TERM
#

echo −n "Envoi du signal TERM a tous les processus..."
/sbin/killall5 −15
evaluate_retval

#
# Envoyer a tous les processus restant (une fois que le signal TERM
# leur a ete envoye) le signal KILL.
#

echo −n "Envoi du signal KILL a tous les processus..."
/sbin/killall5 −9
evaluate_retval

# End /etc/init.d/sendsignals
EOF
```

Création du script setclock

Le script suivant n'a de réelle utilité que lorsque votre horloge matérielle (également appelée horloge BIOS ou CMOS) n'est pas réglée sur le temps GMT. Le réglage recommandé consiste à configurer votre horloge matérielle en GMT et de convertir l'heure en temps local en utilisant le lien symbolique /etc/localtime. Mais si vous utilisez un système d'exploitation qui ne comprend pas une horloge réglée en GMT (les plus répandus sont ceux de Micro$oft), vous pourriez vouloir régler votre horloge en temps local de telle façon que le temps soit correctement affiché sur ces systèmes d'exploitation. Ce script réinitialisera le temps du noyau sur l'horloge matérielle sans le convertir par l'utilisation du lien symbolique /etc/localtime.

Si vous voulez utiliser ce script pour votre système même si votre horloge matérielle est réglée en GMT, modifiez la variable UTC ci−dessous pour lui donner la valeur *1*.

```
cat > setclock << "EOF"
#!/bin/sh
# Begin /etc/init.d/setclock

#
# Inclusion des fonctions declarees dans le fichier /etc/init.d/functions
# et inclusion des variables du fichier /etc/sysconfig/clock.
#

source /etc/init.d/functions
source /etc/sysconfig/clock

#
# Des maintenant, nous voulons regler l'horloge systeme sur l'horloge
# materielle, aussi nous utilisons le parametre −hctosys.
#

CLOCKPARAMS="−−hctosys"

#
# Si la variable UTC est initialisee dans le fichier /etc/sysconfig/clock,
# ajoutez le parametre −u de facon a indiquer a hwclock que l'horloge
# materielle est reglee sur le temps UTC plutot qu'en temps local.
#

case "$UTC" in
      yes|true|1)
            CLOCKPARAMS="$CLOCKPARAMS −u"
            ;;
esac

echo −n "Reglage de l'horloge..."
/sbin/hwclock $CLOCKPARAMS
evaluate_retval

# End /etc/init.d/setclock
```

```
EOF
```

Création du fichier /etc/sysconfig/clock

Créez un nouveau fichier `/etc/sysconfig/clock` en exécutant ce qui suit:

```
cat > /etc/sysconfig/clock << "EOF"
# Begin /etc/sysconfig/clock

UTC=1

# End /etc/sysconfig/clock
EOF
```

Si votre horloge matérielle (désignée également sous le nom d'horloge BIOS ou CMOS) n'est pas réglée sur le temps GMT, donnez à la variable UTC du fichier /etc/sysconfig/clock la valeur *0* (zéro).

Création du script sysklogd

Créez un nouveau fichier `/etc/init.d/sysklogd` contenant ce qui suit:

```
cat > sysklogd << "EOF"
#!/bin/sh
# Begin /etc/init.d/sysklogd

#
# Inclusion des fonctions declarees dans le fichier /etc/init.d/functions
#

source /etc/init.d/functions

case "$1" in
      start)
            echo −n "Demarrage du demon de journalisation systeme..."
            loadproc /usr/sbin/syslogd −m 0

            echo −n "Demarrage du demon de journalisation du noyau..."
            loadproc /usr/sbin/klogd
            ;;

      stop)
            echo −n "Arret du demon de journalisation du noyau..."
            killproc klogd

            echo −n "Arret du demon de journalisation systeme..."
            killproc syslogd
            ;;

      reload)
      echo −n "Rechargement du fichier de configuration du demon de journalisation systeme..."
            reloadproc syslogd 1
            ;;

      restart)
            $0 stop
            /usr/bin/sleep 1
            $0 start
            ;;

      status)
            statusproc /usr/sbin/syslogd
            statusproc /usr/sbin/klogd
            ;;

      *)
            echo "Usage: $0 {start|stop|reload|restart|status}"
```

```
            exit 1
        ;;

esac

# End /etc/init.d/sysklogd
EOF
```

Création du script modèle (template)

Créez un nouveau fichier `/etc/init.d/template` contenant ce qui suit:

```
cat > template << "EOF"
#!/bin/sh
# Begin /etc/init.d/

#
# Inclusion des fonctions declarees dans le fichier /etc/init.d/functions
#

source /etc/init.d/functions

case "$1" in
        start)
                echo –n "Demarrage d..."
                loadproc
                ;;

        stop)
                echo –n "Arret d..."
                killproc
                ;;

        reload)
                echo –n "Rechargement d..."
                reloadproc
                ;;

        restart)
                $0 stop
                /usr/bin/sleep 1
                $0 start
                ;;

        status)
                statusproc
                ;;

        *)
                echo "Usage: $0 {start|stop|reload|restart|status}"
                exit 1
        ;;

esac

# End /etc/init.d/
EOF
```

Création des liens symboliques et réglage des permissions

Donnez à ces fichiers les permissions appropriées et créez les liens symboliques nécessaires en exécutant les commandes suivantes. Si vous n'avez pas créé les scripts loadkeys et setclocks, assurez−vous de ne pas les taper dans les commandes ci−dessous.

```
cd /etc/init.d &&
chmod 754 rc rcS functions checkfs halt loadkeys mountfs
reboot &&
chmod 754 sendsignals setclock sysklogd template  &&
cd ../rc0.d &&
ln −s ../init.d/sysklogd K900sysklogd &&
ln −s ../init.d/sendsignals S800sendsignals &&
ln −s ../init.d/mountfs S900mountfs &&
ln −s ../init.d/halt S999halt &&
cd ../rc6.d &&
ln −s ../init.d/sysklogd K900sysklogd &&
ln −s ../init.d/sendsignals S800sendsignals &&
ln −s ../init.d/mountfs S900mountfs &&
ln −s ../init.d/reboot S999reboot &&
cd ../rcS.d &&
ln −s ../init.d/checkfs S200checkfs &&
ln −s ../init.d/mountfs S300mountfs &&
ln −s ../init.d/setclock S400setclock &&
ln −s ../init.d/loadkeys S500loadkeys &&
cd ../rc1.d &&
ln −s ../init.d/sysklogd K900sysklogd &&
cd ../rc2.d &&
ln −s ../init.d/sysklogd S100sysklogd &&
cd ../rc3.d &&
ln −s ../init.d/sysklogd S100sysklogd &&
cd ../rc4.d &&
ln −s ../init.d/sysklogd S100sysklogd &&
cd ../rc5.d &&
ln −s ../init.d/sysklogd S100sysklogd
```

Création du fichier/etc/fstab file

On utilise le fichier /etc/fstab pour permettre à certains programmes de savoir où certaines partitions devraient avoir été montées par défaut. Créons donc un nouveau fichier `/etc/fstab` contenant ce qui suit:

```
cat > /etc/fstab << "EOF"
# Begin /etc/fstab

/dev/<désignation de la partition LFS> / ext2 defaults 1 1
/dev/<désignation de la partition de swap> swap swap defaults 0 0
proc /proc proc defaults 0 0

# End /etc/fstab
EOF
```

Remplacez <désignation de la partition LFS> et <désignation de la partition de swap> par les périphériques appropriés (/dev/hda5 and /dev/hda6 dans mon cas).

Chapitre 8. Rendre amorçable le système LFS

Introduction

Ce chapitre va rendre LFS amorçable. Il traite de la construction d'un nouveau noyau pour notre nouveau système LFS, et de l'ajout des entrées adéquates à LILO pour que vous puissiez choisir de démarrer le système LFS à l'invite LILO:.

Installer un noyau

Le noyau est le cur d'un système Linux. Nous pouvons utiliser l'image du noyau de notre système habituel, mais nous pouvons aussi compiler un nouveau noyau à partir des sources disponibles les plus récentes du noyau.

Construire le noyau requiert plusieurs étapes, pour le configurer et le compiler. Il y a quelques méthodes pour configurer le noyau. Si vous n'aimez pas la façon de faire de ce livre, lisez le fichier `README` et cherchez quelles autres options existent. Lancez les commandes suivantes 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
```

Ajouter une entrée à LILO

Pour être en mesure de démarrer depuis cette partition, nous devons mettre à jour notre fichier /etc/lilo.conf. Ajoutez les lignes suivantes à lilo.conf en lançant:

```
cat >> /etc/lilo.conf << "EOF"
image=/boot/lfskernel
      label=lfs
      root=<partition>
      read−only
EOF
```

<partition> doit être remplacé par la désignation de votre partition (qui serait /dev/hda5 en ce qui me concerne).

Maintenant, mettez à jour le chargeur de lancement en lançant:

```
/sbin/lilo
```

Redémarrer le système

Maintenant que tous les logiciels ont été installés et que les scripts de démarrage ont été créés, il est temps de redémarrer votre ordinateur. Arrêtez maintenant votre système par shutdown −r et relancez−le sous LFS. Après le redémarrage, vous aurez une invite de connexion normale, comme celle que vous aviez sur votre système Linux habituel (à moins que vous n'utilisiez XDM ou un autre type de gestionnaire d'affichage (comme KDM − la version KDE de XDM).

Une chose reste à faire, et c'est la configuration du réseau. Après avoir redémarré et terminé le chapitre suivant de ce livre, votre système LFS sera prêt à l'utilisation et vous pourrez en faire ce que vous voudrez.

Chapitre 9. Configurer un réseau de base

Introduction

Ce chapitre va configurer un réseau de base. Bien que vous ne soyiez pas forcément connecté à un réseau, les logiciels Linux utilisent cependant des fonctions de réseau. Nous allons installer au moins le périphérique de boucle local, et une carte réseau si cela vous concerne. Les scripts appropriés seront également créés pour que le réseau soit activé lors du démarrage.

Installer les logiciels de réseau

Installer 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` que peuvent vous intéresser.

Installer Net–tools

Editez le fichier `Makefile` et modifiez la variable CFLAGS si vous voulez ajouter des optimisations du compilateur.

Installez Net–tools en lançant les commandes suivantes:

```
make &&
make install
```

Créer le script de démarrage /etc/init.d/localnet

Créez un nouveau fichier `/etc/init.d/localnet` contenant ce qui suit:

```
cat > /etc/init.d/localnet << "EOF"
#!/bin/sh
# Debut /etc/init.d/localnet

#
# Inclure les fonctions declarees dans le fichier /etc/init.d/functions
# et les variables du fichier /etc.sysconfig/network.
#

source /etc/init.d/functions
source /etc/sysconfig/network

case "$1" in
      start)
            echo –n "Lancement de l'interface de boucle (loopback)..."
            /sbin/ifconfig lo 127.0.0.1
            evaluate_retval

            echo –n "Configuration de hostname..."
            /bin/hostname $HOSTNAME
            evaluate_retval
            ;;

      stop)
            echo –n "Arret de l'interface de boucle (loopback)..."
            /sbin/ifconfig lo down
            evaluate_retval
            ;;

      restart)
            $0 stop
            sleep 1
            $0 start
            ;;
      *)
            echo "Usage: $0: {start|stop|restart}"
            exit 1
            ;;
esac

# Fin /etc/init.d/localnet
EOF
```

Configurer les permissions et le lien symbolique

Positionnez les permissions correctes des fichiers et créez le lien symbolique nécessaire en lançant les commandes suivantes:

```
cd /etc/init.d &&
chmod 754 localnet &&
cd ../rcS.d &&
ln -s ../init.d/localnet S100localnet
```

Créer le fichier /etc/sysconfig/network

Créez un nouveau fichier `/etc/sysconfig/network` et mettez–y le nom de l'hôte en lançant:

```
echo "HOSTNAME=lfs" > /etc/sysconfig/network
```

Remplacez "lfs" par le nom que vous voulez affecter à votre ordinateur. Notez bien que vous ne devez pas entrer le FQDN (Nom de Domaine Entièrement Qualifié) ici. Cette information sera mise dans le fichier `/etc/hosts` ultérieurement.

Créer le fichier /etc/hosts

Si vous voulez configurer une carte réseau, vous devez choisir une adresse IP, son FQDN et d'éventuels alias à utiliser dans le fichier /etc/hosts. Un exemple est:

<mon−IP> hote.domaine.org alias

Assurez−vous que l'adresse IP appartient bien à l'espace d'adressage IP du réseau privé. Les choix valides sont:

Class	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 correcte peut être 192.168.1.1. Un FQDN valide pour cette adresse IP peut être www.linuxfromscratch.org.

Si vous ne comptez pas utiliser une carte réseau, vous devez quand même avoir un FQDN. Ceci est nécessaire pour que des programmes comme Sendmail fonctionnent correctement (en fait, Sendmail ne démarrera pas s'il ne peut pas déterminer le FQDN).

Si vous ne configurez pas de carte réseau, créez un nouveau fichier `/etc/hosts` en lançant:

```
cat > /etc/hosts << "EOF"
# Debut /etc/hosts (version sans carte reseau)

127.0.0.1 www.mondomaine.com <valeur de HOSTNAME> localhost

# Fin /etc/hosts (version sans carte reseau)
EOF
```

Si vous configurez une carte réseau, créez un nouveau fichier `/etc/hosts` contenant:

```
cat > /etc/hosts << "EOF"
# Debut /etc/hosts (version avec carte reseau)

127.0.0.1 localhost.localdomain localhost
192.168.1.1 www.mondomaine.org <valeur de HOSTNAME>

# Fin /etc/hosts (version avec carte reseau)
EOF
```

Bien sûr, modifiez les valeurs 192.168.1.1 et www.mondomaine.org selon votre choix (ou ce qui est imposé si une adresse IP vous est assignée par un administrateur réseau/système et que vous prévoyez de connecter cette machine à ce réseau).

Créer le script /etc/init.d/ethnet

Cette partie vous concerne uniquement dans le cas où vous voulez configurer une carte réseau. Autrement, vous pouvez la sauter.

Créez un nouveau fichier `/etc/init.d/ethnet` contenant ce qui suit:

```
cat > /etc/init.d/ethnet << "EOF"
#!/bin/sh
# Debut /etc/init.d/ethnet
#
# Script principal par Gerard Beekmans – gerard@linuxfromscratch.org
# Verification de GATEWAY par Jean–François Le Ray – jfleray@club–internet.fr
#

#
# Inclus les fonctions declarees dans le fichier /etc/init.d/functions
# et les variables du fichier /etc/sysconfig/network.
#

source /etc/init.d/functions
source /etc/sysconfig/network

case "$1" in
      start)

#
# Parcours de tous les fichiers de configuration de la carte reseau
#

      for interface in $(ls /etc/sysconfig/network–scripts/ifcfg* | \
         grep –v ifcfg–lo)
      do
#
# Chargement des variables depuis ce fichier
#

         source $interface
#
# Si la variable ONBOOT est positionnee a yes, traiter ce fichier et
# lancer l'interface
#

         if [ "$ONBOOT" == yes ]
         then
            echo –n "Lancement de l'interface $DEVICE..."
            /sbin/ifconfig $DEVICE $IP broadcast $BROADCAST \
               netmask $NETMASK
            evaluate_retval
```

```
        fi
    done

#
# Si le fichier /etc/sysconfig/network contient une variable GATEWAY,
# positionne la passerelle.
#

        if [ "$GATEWAY" != "" ]; then
            echo −n "Configuration du routage pour l'interface eth0..."
            /sbin/route add default gw $GATEWAY metric 1
            evaluate_retval
        fi
            ;;

    stop)

#
# Parcours de tous les fichiers de configuration de la carte reseau
#

    for interface in $(ls /etc/sysconfig/network−scripts/ifcfg* | \
        grep −v ifcfg−lo)
    do
#
# Chargement des variables depuis ce fichier
#

        source $interface

#
# Si la variable ONBOOT est positionnee a yes, traiter ce fichier et
# arreter l'interface
#

        if [ $ONBOOT == yes ]
        then
            echo −n "Arret de l'interface $DEVICE..."

            /sbin/ifconfig $DEVICE down
            evaluate_retval
        fi
    done
            ;;

    restart)
            $0 stop
            sleep 1
            $0 start
            ;;
    *)
```

```
        echo "Usage: $0 {start|stop|restart}"
        exit 1
        ;;
esac

# Fin /etc/init.d/ethnet
EOF
```

Ajouter la passerelle par défaut à /etc/sysconfig/network

Si vous avez besoin de configurer une passerelle par défaut, lancez la commande suivante:

```
cat >> /etc/sysconfig/network << "EOF"
GATEWAY=192.168.1.2
EOF
```

Modifiez GATEWAY pour le faire correspondre à votre configuration du réseau.

Créer les fichiers de configuration NIC

Quelles interfaces sont lancées ou arrêtées par le script ethnet dépend des fichiers dans le répertoire /etc/sysconfig/network−scripts. Ce répertoire devrait contenir des fichiers de la forme ifcfg−x où x est un numéro d'identification (ou quelque autre dénomination que voulez).

Créez d'abord le répertoire network−scripts en lançant:

```
mkdir /etc/sysconfig/network-scripts
```

Maintenant, créez de nouveaux fichiers dans ce répertoire, contenant ce qui suit. Des exemples de nom de fichier sont ifcfg−eth0, ifcfg−eth0:3 et ifcfg−eth1:2.

```
ONBOOT=yes
DEVICE=eth0
IP=192.168.1.1
NETMASK=255.255.255.0
BROADCAST=192.168.1.255
```

Bien entendu, modifiez les valeurs de ces quatre variables dans chaque fichier selon votre propre configuration. En général, NETMASK et BROADCAST restent tels quels, et seules les variables DEVICE et IP changent pour chaque interface réseau. Si la variable ONBOOT est positionnée à yes, le script ethnet la

lancera lors du démarrage du système. Si elle est positionnée à une autre valeur que yes, le fichier sera ignoré par le script ethnet, et donc l'interface ne sera pas lancée.

Configurer les permissions et les liens symboliques

Positionnez les permissions correctes des fichiers et créez les liens symboliques nécessaires en lançant les commandes suivantes

```
cd /etc/init.d &&
chmod 754 ethnet &&
cd ../rc3.d &&
ln -s ../init.d/ethnet S200ethnet &&
cd ../rc4.d &&
ln -s ../init.d/ethnet S200ethnet &&
cd ../rc5.d &&
ln -s ../init.d/ethnet S200ethnet
```

III. Troisième Partie – Annexes

Table des matières

A. Description des paquets
B. Ressources
C. Sites de téléchargement officiels

Annexe A. Description des paquets

Introduction

Cet appendice décrit les aspects suivants de chaque package installé dans ce livre:

- Le contenu de chaque package
- La fonction de chaque programme d'un package

Les packages sont listés dans le même ordre que celui de leur installation dans les chapitres 5 (systèmes Intel) ou 11 (systèmes PPC).

La plupart des informations sur ces packages (surtout leur description) provient des pages man de ces packages. Je ne vais pas afficher toute la page de man, mais juste les éléments vitaux pour que vous compreniez ce que fait un programme. Si vous voulez connaître tous les détails sur un programme, je vous conseille de commencer par lire la page de man complètement, en plus de cet appendice.

Vous trouverez aussi que certains packages sont documentés plus en profondeur que d'autres. Ceci est dû au fait que je commence à mieux connaître certains packages que d'autres. Si vous avez quelque chose à ajouter aux descriptions qui suivent, n'hésitez pas à m'envoyer un mail. Cette liste est prévue pour contenir une description poussée de chaque package installé, mais je ne peux faire ceci seul. J'ai reçu de l'aide de plusieurs personnes, mais une aide supplémentaire est encore nécessaire.

Remarquez bien que, actuellement, seul ce que fait un package est décrit, et non pas pourquoi vous avez besoin de l'installer. Ceci sera ajouté ultérieurement.

Glibc

Contenu

Le package Glibc contient la bibliothèque GNU C.

Description

La bibliothèque C est un ensemble de fonctions couramment utilisées dans les programmes. Ainsi, un programmeur n'a pas besoin de créer ses propres fonctions pour chaque tâche de base. Les opérations les plus courantes comme écrire une ligne à l'écran sont déjà présentes et à la disposition du programmeur.

La bibliothèque C (comme, actuellement, presque toutes les bibliothèques) est fournie en deux versions: dynamique et statique. En bref, quand un programme utilise une bibliothèque C statique, le code de la bibliothèque C est copié dans le fichier exécutable. Quand un programme utilise une bibliothèque dynamique, cet exécutable ne contiendra pas le code de la bibliothèque C, mais simplement une routine qui chargera les fonctions depuis la bibliothèque au moment de l'exécution du programme. Cela signifie une réduction significative de la taille du fichier du programme. Si vous ne comprenez pas ce concept, vous feriez mieux de lire la documentation fournie avec la bibliothèque C, car cela est trop compliqué à expliquer ici en une ou deux lignes.

Noyau Linux

Contenu

Le package Noyau Linux contient le noyau de Linux.

Description

Le noyau de Linux est au cur 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.

Ed

Contenu

Le package Ed contient le programme ed.

Description

ed est un éditeur de texte par ligne. Il est utilisé pour créer, afficher, modifier et réaliser d'autres manipulations sur les fichiers texte.

Patch

Contenu

Le package Patch contient le programme patch.

Description

Le programme patch modifie un fichier en accord avec un fichier patch. Un fichier patch est, en général, une liste créée par le programme diff, qui contient des instructions sur la manière dont un fichier original doit être modifié. patch est très utilisé pour les corrections de code source, car il fait économiser du temps et de la place. Imaginez que vous avez un pacakge d'une taille de 1Mo. Dans la version suivante de ce package ne sont modifiés que deux des fichiers de la première version. Vous pouvez mettre à disposition un package entièrement nouveau de 1Mo, ou fournir un fichier patch de 1ko qui mettra à jour la première version pour la rendre identique à la seconde. Ainsi, si vous avez déjà téléchargé la première version, un fichier patch peut vous éviter un second téléchargement important.

GCC

Contenu

Le package GCC contient les compilateurs, les préprocesseurs et la bibliothèque GNU C++.

Description

Compilateur

Un compilateur traduit le code source sous format texte en un format que l'ordinateur peut comprendre. Après qu'un fichier de code source a été compilé en un fichier objet, un lieur crée un fichier exécutable à partir d'un ou plusieurs de ces fichiers objet générés par le compilateur.

Préprocesseur

Un préprocesseur pré−traduit un fichier source, faisant par exemple l'inclusion des contenus des fichiers d'en−tête dans le fichier source. Vous ne faites pas cela vous−même, en général, pour gagner du temps. Vous insérez simplement une ligne comme #include <nom_fichier>. Le préprocesseur insère le contenu de ce fichier dans le fichier source. C'est une des choses que fait le préprocesseur.

Bibliothèque C++

La bibliothèque C++ est utilisée par les programmes C++. La bibliothèque C++ contient les fonctions qui sont couramment utilisées dans les programmes C++. Ainsi, le programmeur n'a pas à ré−écrire certaines fonctions (comme écrire une chaîne de caractères à l'écran) à chaque fois qu'il crée un programme.

Bison

Contenu

Le package Bison contient le programme bison.

Description

Bison est un générateur de parseur, en remplacement de YACC. YACC signifie Encore un Autre Compilateur de Compilateurs. Qu'est−ce donc que Bison? C'est un programme qui génère un programme qui analyse la structure d'un fichier texte. Au lieu d'écrire le programme, vous spécifiez comment les choses doivent être reliées et, avec ces règles, un programme est construit pour analyser les fichiers texte.

Il y a de nombreux exemples où une structuration est nécessaire et l'un d'eux est la calculatrice.

Soit la chaîne:

```
1 + 2 * 3
```

Vous pouvez facilement arriver au résultat de 7. Pourquoi? A cause de sa structure. Vous savez 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
```

Vous commencez en bas de l'arbre, et vous arrivez aux nombres 2 et 3, qui sont liés par un symbole de multiplication; l'ordinateur multiplie donc 2 et 3. Le résultat de cette multiplication est mémorisé et la prochaine chose 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 uniquement utilisé pour les calculatrices.

Mawk

Contenu

Le paquet Mawk contient le programme mawk.

Description

mawk

mawk est un interpréteur pour le langage de programmation AWK. Le langage AWK est utile pour manipuler des fichiers de données, retrouver et travailler sur du texte, et pour prototyper et expérimenter des algorithmes.

Findutils

Contenu

Le package Findutils contient les programmes find, locate, updatedb et xargs.

Description

find

Le programme find recherche les fichiers, dans une hiérarchie de répertoires, qui correspondent à un certain critère. Si aucun critère n'est fourni, il liste tous les fichiers dans le répertoire courant et ses sous–répertoires.

locate

locate scrute une base de données qui contient tous les fichiers et répertoires d'un système de fichiers. Ce programme liste les fichiers et répertoires de cette base de données qui vérifient un certain critère. Si vous recherchez un fichier, ce programme va scruter la base de données et vous dire exactement où sont situés les fichiers que vous avez demandé. Cela n'a de sens que si la base de données de localisation est correctement mise à jour, autrement ce programme vous fournira des informations périmées.

updatedb

Le programme updatedb met à jour la base de données de localisation. Il scrute tout le système de fichiers (y compris les autres systèmes de fichiers actuellement montés, sauf si vous lui spécifiez de ne pas le faire), et enregistre chaque répertoire et fichier qu'il trouve dans la base de données qui sera utilisée par le programme locate, qui retrouve ces informations. C'est une bonne habitude que de mettre à jour cette base de données quotidiennement, pour vous assurer que la base de données est à jour.

xargs

La commande xargs applique une commande à une liste de fichiers. Si vous devez exécuter la même commande sur de multiples fichiers, vous pouvez créer un fichier qui contient tous ces fichiers (un par ligne) et utiliser xargs pour lancer une commande sur la liste.

Ncurses

Contenu

Le package Ncurses contient les bibliothèques ncurses, panel, menu et form. Il contient également les programmes tic, infocmp, clear, tput, toe et tset.

Description

Les bibliothèques

Les bibliothèques qui constituent la bibliothèque Ncurses sont utilisées pour afficher du texte (souvent de manière fantaisiste) sur votre écran. Un exemple d'utilisation de ncurses se trouve dans le processus "make menuconfig" du noyau. Les bibliothèques contiennent les routines pour créer des panneaux, des menus, des formulaires, et des routines générales d'affichage de texte.

---

tic

tic est le compilateur de description des entrées de terminfo. Le programme traduit un fichier terminfo du format source en format binaire utilisable avec les routines de la bibliothèque ncurses. Les fichiers terminfo contiennent des informations sur les capacités de votre terminal.

---

infocmp

Le programme infocmp peut être utilisé pour comparer une entrée binaire de terminfo avec d'autres, ré−écrire une description terminfo pour tenir compte du champ use= de terminfo, ou afficher une description terminfo du fichier binaire (term) sous différents formats (à l'opposé de ce que fait tic).

---

clear

Le programme clear efface votre écran si cela est possible. Il regarde dans l'environnement du type de terminal, puis dans la base de données terminfo, pour trouver comment effacer l'écran.

---

tput

Le programme tput utilise la base de données terminfo pour construire les valeurs des capacités dépendant du terminal et les informations disponibles pour le shell, pour initialiser ou réinitialiser le terminal, ou renvoyer le nom long du type de terminal requis.

---

toe

Le programme toe liste tous les types de terminaux disponibles par leur nom primaire, avec leur description.

---

tset

Le programme tset initialise les terminaux pour qu'ils puissent être utilisés, mais il n'est plus beaucoup utilisé désormais. Il est fourni pour la compatibilité avec BSD 4.4.

Less

Contenu

Le package Less contient le programme less.

Description

Le programme less est un paginateur de fichier (ou un visualisateur de texte). Il affiche le contenu d'un fichier avec la possibilité de défilement. Less est une évolution du paginateur habituel appelé "more". Less permet 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 quand vous lisez de gros fichiers.

Groff

Contenu

Le package Groff contient les programmes addftinfo, afmtodit, eqn, grodvi, groff, grog, grohtml, grolj4, grops, grotty, hpftodit, indxbib, lkbib, lookbib, neqn, nroff, pfbtops, pic, psbb, refer, soelim, tbl, tfmtodit et troff.

Description

addftinfo

addftinfo lit un fichier de fonte troff et y ajoute des informations supplémentaires de métrique de fonte, afin qu'il soit utilisable par le système groff.

---

afmtodit

afmtodit crée un fichier de fonte pour être utilisé par groff et grops.

---

eqn

eqn compile les descriptions d'équations intégrées dans des fichiers d'entrée troff en commandes pouvant être comprises par troff.

---

grodvi

grodvi est un pilote pour groff qui produit un format dvi TeX.

---

groff

groff est un frontal pour le système de formatage de document groff. Normalement, il lance le programme troff et un postprocesseur approprié au périphérique choisi.

---

grog

grog lit des fichiers et tente de trouver lesquelles des options groff −e, −man, −me, −mn, −ms, −p, −s et −t sont nécessaires pour imprimer les fichiers, et affiche la commande groff incluant ces options sur la sortie standard.

---

grohtml

grohtml traduit la sortie de troff GNU en html.

---

grolj4

grolj4 est un pilote pour groff qui produit une sortie en format PCL5 utilisable par une imprimante HP Laserjet 4.

---

grops

grops traduit la sortie de troff GNU en PostScript.

---

grotty

grotty traduit la sortie de troff GNU en une forme convenable pour les périphériques de type machine à écrire.

---

hpftodit

hpftodit crée un fichier de fonte destiné à être utilisé avec groff −Tlj4 à partir d'un fichier de métrique de fonte spécifique HP.

---

indxbib

indxbib crée un index de permutation pour les bases de données bibliographiques à partie du fichier spécifié, destiné à être utilisé par refer, lookbib et lkbib.

---

lkbib

lkbib recherche dans les bases de données bibliographiques les références contenant les clés spécifiées, et affiche les références trouvées sur la sortie standard.

---

lookbib

lookbib affiche une invite sur la sortie d'erreur standard (sauf si l'entrée standard n'est pas un terminal), lit une ligne depuis l'entrée standard contenant un ensemble de mots−clés, recherche dans les bases de données bibliographiques dans le fichier spécifié les références contenant ces mots−clés, affiche les références trouvées sur la sortie standard, et répète le processus jusqu'à la fin de l'entrée.

---

neqn

Nous ne savons actuellement pas ce qu'est neqn, ni ce qu'il fait.

nroff

Le script nroff émule la commande nroff en utilisant groff.

pfbtops

pfbtops traduit une fonte PostScript de format .pfb en ASCII.

pic

pic compile les descriptions d'images incluses dans des fichiers d'entrée troff ou TeX en commandes qui peuvent être comprises par TeX ou troff.

psbb

psbb lit un fichier qui doit être un document PostScript conforme aux conventions de Structuration des Documents, et cherche un commentaire %%BoundingBox.

refer

refer copie le contenu d'un fichier sur la sortie standard, sauf que les lignes entre .[et .] sont interprétées comme des citations, et que les lignes entre .R1 et .R2 sont interprétées comme des commandes sur la manière de traiter les citations.

soelim

soelim lit les fichiers et remplace les lignes de la forme *fichier .so* par le contenu de *fichier*.

tbl

tbl compile les descriptions de tables intégrées dans les fichiers d'entrée troff en commandes qui peuvent être comprises par troff.

tfmtodit

tfmtodit crée un fichier de fonte à utiliser avec **`groff -Tdvi`**.

troff

troff est en grande partie compatible avec le troff Unix. En général, il doit être invoqué avec la commande groff, qui lancera aussi les préprocesseurs et les postprocesseurs dans l'ordre approprié, et avec les options qui conviennent.

Man

Contenu

Le paquet Man contient les programmes man, apropos, whatis et makewhatis.

Description

man

man formate et affiche les pages de manuel en ligne.

apropos

apropos recherche dans un ensemble de fichiers de base de données contenant des descriptions brèves des commandes système, en fonction de mots−clés, et affiche le résultat sur la sortie standard.

whatis

whatis recherche dans un ensemble de fichiers de base de données contenant des descriptions brèves de commandes système, en fonction de mots−clés, et affiche le résultat sur la sortie standard. Seules les correspondances de mots complets sont affichées.

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 avec le contenu de la section NAME de la page du manuel.

Perl

Contenu

Le package Perl contient Perl – Extraction et Rapport Pratiques

Description

Perl combine les capacités et les fonctionnalités de C, awk, sed et sh dans un puissant langage de programmation.

M4

Contenu

Le package M4 contient le processeur M4.

Description

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 les fichiers nommés, lancer des commandes UNIX, faire de l'arithmétique d'entiers, manipuler du texte de diverses façons, faire de la récursivité, etc. M4 peut être utilisé soit comme frontal d'un compilateur, ou comme processeur de macros à part.

Texinfo

Contenu

Le paquet Texinfo contient les programmes info, install−info, makeinfo, texi2dvi et texindex.

Description

info

Le programme info lit les documents Info, en général situés dans votre répertoire /usr/doc/info. Les documents Info sont comme des pages de man(uel), mais ont tendance à aller plus en profondeur que la simple explication des options d'un programme.

install−info

Le programme install−info met à jour les entrées d'info. Quand vous exécutez le programme info, une liste de sujets disponibles (c'est−à−dire les documents info disponibles) est affichée. Le programme install−info est utilisé pour tenir cette liste de sujets disponibles à jour. Si vous décidez de supprimer des fichiers d'info manuellement, vous devrez également supprimer le sujet dans le fichier d'index. Ce programme est utilisé dans ce but. Il fonctionne également, dans l'autre sens, quand vous ajoutez des documents d'info.

makeinfo

Le programme makeinfo traduit des documents source Texinfo en différents 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.

Autoconf

Contenu

Le paquet Autoconf contient les programmes autoconf, autoheader, autoreconf, autoscan, autoupdate et ifnames.

Description

autoconf

autoconf est un outil pour produire des scripts shell qui configureront automatiquement les paquets de code source logiciel pour les adapter à de nombreux systèmes UNIX. Les scripts de configuration produits par Autoconf sont indépendants d'Autoconf quand ils sont lancés; leurs utilisateurs n'ont donc pas besoin d'avoir Autoconf.

---

autoheader

Le programme autoheader peut créer un fichier modèle de commandes C #define à utiliser pour la configuration.

---

autoreconf

Si vous avez beaucoup de scripts générés par autoconf, le programme autoreconf peut vous faire gagner du temps. Il lance autoconf (et autoheader, si besoin) de façon répétée pour reconstruire les scripts de configuration Autoconf et les modèles d'entête de configuration dans l'arborescence rattachée au répertoire courant.

---

autoscan

Le programme autoscan peut vous aider à créer un fichier configure.in pour un paquet logiciel. autoscan scrute les fichiers source dans une arborescence rattachée à un répertoire donné en argument de ligne de commande, ou le répertoire courant si aucun n'est spécifié. Il recherche les fichiers source pour des problèmes habituels de portabilité et crée un fichier configure.scan qui est un préliminaire à configure.in pour ce paquet.

---

autoupdate

Le programme autoupdate met à jour un fichier configure.in qui appelle les macros Autoconf par leurs anciens noms, afin qu'il utilise les noms de macro actuels.

---

ifnames

ifnames peut servir lors de l'écriture d'un configure.in pour un paquet logiciel. Il affiche les identifiants que le paquet utilise déjà dans les conditions de préprocesseur C. Si un paquet a déjà été mis à jour pour une portabilité quelconque, ce programme peut vous aider à trouver ce que sa configuration doit vérifier. Il peut aider à remplir quelques trous dans un configure.in généré par autoscan.

Automake

Contenu

Le paquet Automake contient les programmes aclocal et automake.

Description

aclocal

Automake inclut un certain nombre de macros Autoconf qui peuvent être utilisées dans votre paquet; certaines d'entre elles sont requises par Automake, dans certains cas. Ces macros doivent être définies dans votre aclocal.m4; autrement, elles ne seront pas vues par autoconf.

Le programme aclocal va générer automatiquement les fichiers aclocal.m4, en fonction du contenu de configure.in. Ceci fournit un bon moyen d'obtenir des macros fournies par Automake, sans avoir à rechercher ici ou là. De plus, le mécanisme aclocal peut être étendu pour pouvoir être utilisé pour d'autres paquets.

automake

Pour créer tous les fichiers Makefile.in d'un paquet, lancez le programme automake dans le répertoire racine, sans arguments. automake va chercher automatiquement chaque Makefile.am approprié (en parcourant configure.in) et générer le Makefile.in correspondant.

Bash

Contenu

Le paquet Bash contient le programme bash.

Description

bash est le SHell Bourne−Again, qui est un interpréteur de commandes couramment utilisé sur les systèmes Unix. bash est un programme qui lit l'entrée standard, c'est−à−dire le clavier. Vous tapez quelque chose et le programme évalue ce que vous avez entré, et en fait quelque chose, comme lancer un programme.

Flex

Contenu

Le package Flex contient le programme flex.

Description

flex est un outil pour générer des programmes qui reconnaissent des motifs dans le texte. La reconnaissance de motifs est très utile dans de nombreuses applications. Vous définissez les règles de recherche, et flex crée un programme qui recherche ces motifs. La raison pour laquelle les gens utilisent flex est qu'il est plus facile de définir des règles sur ce qu'il faut rechercher, que d'écrire un programme qui scrute le texte pour cela.

Binutils

Contenu

Le paquet Binutils contient les programmes ld, as, ar, nm, objcopy, objdump, ranlib, size, strings, strip, c++filt, addr2line et nlmconv.

Description

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.

---

as

as a d'abord été prévu pour assembler la sortie du compilateur GNU C gcc, pour pouvoir être utilisée par le lieur ld.

---

ar

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

---

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 dans 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 que leur programme compile et fonctionne.

ranlib

ranlib génère un index sur le 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.

size

size liste les tailles des sections −− et la taille totale −− de chaque fichier objet objfile dans sa liste d'arguments. Par défaut, une ligne est générée en sortie pour chaque fichier objet ou chaque module dans l'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 sont suivies d'un caractère non imprimable. Par défaut, il affiche seulement les chaînes des parties initialisées et chargées des fichiers objet; pour d'autres types de fichier, il affiche les chaînes de tout le fichier.

strings est essentiellement utilisé pour déterminer le contenu des fichiers non−texte.

strip

strip débarrasse les fichiers objet de tous les symboles, ou de certains en particuliers. 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.

c++filt

Le langage C++ fournit un mécanisme de surcharge, ce qui signifie que vous pouvez écrire plusieurs fonctions ayant le même nom (pourvu que chacune réclame 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 le lieur puisse prévenir un conflit entre ces fonctions surchargées.

addr2line

addr2line traduit les adresses d'un programme en noms de fichier 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.

nlmconv

nlmconv convertit des fichiers objet re−localisables en fichiers de Module Installables NetWare, pouvant aussi lire les fichiers d'entête pour les informations d'entête NLM.

Bzip2

Contenu

Le package Bzip2 contient les programmes bzip2, bunzip2, bzcat et bzip2recover.

Description

bzip2

bzip2 compresse les fichiers avec l'algorithme de compression de texte par tri de blocs de Burrows−Wheeler, et le codage de Huffman. La compression est en général bien meilleure que celle obtenue par les compresseurs plus conventionnels basés sur LZ77/LZ78, et s'approche des performances de la famille PPM des compresseurs statistiques.

bunzip2

bunzip2 décompresse les fichiers qui ont été compressés avec bzip2.

bzcat

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

bzip2recover

bzip2recover restaure les données des fichiers bzip2 endommagés.

Diffutils

Contenu

Le package Diffutils contient les programmes cmp, diff, diff3 et sdiff.

Description

cmp et diff

cmp et diff comparent tous deux deux fichiers et montrent leurs différences. Ces deux programmes ont des options pour comparer des fichiers dans différentes situations.

diff3

La différence entre diff et diff3 est que diff compare deux fichiers, et diff3 en compare trois.

sdiff

sdiff fusionne deux fichiers et affiche interactivement le résultat.

E2fsprogs

Contenu

Le package e2fsprogs contient les programmes chattr, lsattr, uuidgen, badblocks, debugfs, dumpe2fs, e2fsck, e2label, fsck, fsck.ext2, mke2fs, mkfs.ext2, mklost+found et tune2fs.

Description

chattr

chattr modifie les attributs d'un fichier sur un système de fichier Linux ext2.

---

lsattr

lsattr liste les attributs d'un fichier sur un système de fichiers 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.

---

badblocks

badblocks est utilisé pour rechercher les secteurs défectueux sur un périphérique (en général une partition de disque).

---

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.

---

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.

---

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 répertoire 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.

---

tune2fs

tune2fs ajuste les paramètres d'optimisation de système de fichiers sur un système de fichiers Linux ext2.

File

Contenu

Le pacakge File contient le programme file.

Description

file teste chaque fichier spécifié pour essayer de le classer. Il y a trois sortes de tests, réalisés dans cet ordre: tests du système de fichiers, test des nombres magiques et test de langage. Le premier test qui réussit provoque l'affichage du type de fichier.

Fileutils

Contenu

Le package Fileutils contient les programmes chgrp, chmod, chown, cp, dd, df, dir, dircolors, du, install, ln, ls, mkdir, mkfifo, mknod, mv, rm, rmdir, sync, touch et vdir.

Description

chgrp

chgrp modifie le groupe d'appartenance de chaque fichier donné en lui affectant le groupe spécifié, qui peut être un nom de groupe ou un ID de groupe numérique.

---

chmod

chmod modifie les permissions de chaque fichier spécifié selon le mode donné, qui peut être soit une représentation symbolique des changements à effectuer, ou un nombre octal représentant le schéma de bits pour les nouvelles permissions.

---

chown

chown modifie l'utilisateur et/ou le groupe d'appartenance de chaque fichier spécifié.

---

cp

cp copie les fichiers d'un endroit dans un autre.

---

dd

dd copie un fichier (depuis l'entrée standard vers la sortie standard, par défaut) avec une taille de bloc choisie par l'utilisateur, en pouvant éventuellement y faire des conversions.

---

df

df affiche la quantité d'espace disque disponible sur le système de fichiers contenant chaque nom de fichier en argument. Si aucun nom de fichier n'est donné, l'espace disponible sur tous les systèmes de fichier actuellement montés est affiché.

---

ls, dir et vdir

dir et vdir sont des versions de ls avec différents formats de sortie par défaut. Ces programmes listent chaque fichier ou répertoire spécifié. Le contenu des répertoires est trié alphabétiquement. Pour ls, les fichiers sont listés par défaut en colonne, triés verticalement, si la sortie standard est un terminal; autrement, ils sont listés un par ligne. Pour dir, les fichiers sont listés par défaut en colonne, triés verticalement. Pour vdir, les fichiers sont listés par défaut en format long.

---

dircolors

dircolors affiche les commandes pour positionner la variable d'environnement LS_COLOR. La variable LS_COLOR est utilisée pour modifier le schéma de couleurs par défaut utilisé par ls et les utilitaires associés.

---

du

du affiche la quantité d'espace disque utilisée par chaque argument, et pour chaque sous−répertoire du répertoire en argument.

---

install

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

---

ln

ln crée des liens physiques ou logiques (symboliques) entre fichiers.

---

mkdir

mkdir crée des répertoires avec un nom spécifié.

---

mkfifo

mkfifo crée un FIFO avec chaque nom spécifié.

---

mknod

mknod crée un FIFO, fichier spécial de type caractère ou bloc, avec le nom de fichier spécifié.

---

mv

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

---

rm

rm supprime des fichiers ou des répertoires.

---

rmdir

rmdir supprime des répertoires, s'ils sont vides.

---

sync

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

---

touch

touch positionne les heures d'accès et de modification de chaque fichier spécifié à l'heure courant. Les fichiers qui n'existent pas sont créés vides.

Gettext

Contenu

Le pacakge gettext contient les programmes gettext, gettextize, msgcmp, msgcomm, msgfmt, msgmerge, msgunfmt et xgettext.

Description

gettext

Le pacakge gettext est utilisé pour l'internationalisation (connue aussi sous le terme de i18n) et pour la localisation (connue aussi sous le terme de l10n). Les programmes peuvent être compilés avec le Support du Langage Natal (NLS) qui leur permet d'afficher les messages dans votre langue natale plutôt qu'en anglais, par défaut.

Grep

Contenu

Le package grep contient les programmes egrep, fgrep et grep.

Description

egrep

egrep affiche les lignes des fichiers qui vérifient une expression régulière étendue.

---

fgrep

fgrep affiche les lignes des fichiers qui vérifient une liste de chaînes fixes, séparées par des caractères de fin de ligne, l'une quelconque de ces chaînes devant correspondre.

---

grep

grep affiche les lignes des fichiers qui vérifient une expression régulière basique.

Gzip

Contenu

Le package Gzip contient les programmes compress, gunzip, gzexe, gzip, uncompress, zcat, zcmp, zdiff, zforece, zgrep, zmore et znew.

Description

gunzip

gunzip décompresse les fichiers qui ont été compressés avec gzip.

---

gzexe

gzexe vous permet de compresser des exécutables en les remplaçant, et de faire en sorte qu'ils se décompressent automatiquement et s'exécutent lorsque vous les lancez (avec une dégradation des performances).

---

gzip

gzip réduit la taille des fichiers nommés en utilisant le codage de Lempel−Zip (LZ77).

---

zcat

zcat décompresse soit une liste de fichiers fournie en ligne de commande ou sur son entrée standard, et écrit les données décompressées sur la sortie standard.

---

zcmp

zcmp invoque le programme cmp sur des fichiers compressés.

---

zdiff

zdiff invoque le programme diff sur des fichiers compressés.

---

zforce

zforce force l'extension .gz sur tous les fichiers gzip, pour que gzip ne les compresse pas deux fois. Ceci peut être utile pour des fichiers dont les noms ont été tronqués après un transfert.

---

zgrep

zgrep invoque le programme grep sur des fichiers compressés.

---

zmore

zmore est un filtre qui permet d'examiner des fichiers texte simples ou compressés, écran par écran, sur un terminal (de la même manière que more).

---

znew

znew recompresse des fichiers du format .Z (compress) en format .gz (gzip).

Ld.so

Contenu

Du package Ld.so, nous utiliserons les programmes ldconfig et ldd.

Description

ldconfig

ldconfig crée les liens et la mémoire cache nécessaires (pour une utilisation par le lieur dynamique, ld.so) pour les bibliothèques partagées les plus récentes trouvées dans les répertoires spécifiés sur la ligne de commande, dans le fichier /etc/ld.so.conf, et dans les répertoires par défaut (/usr/lib et /lib). ldconfig vérifie l'en−tête et le nom de fichier des bibliothèques trouvées en déterminant quelles versions doivent voir leurs liens mis à jour.

---

ldd

ldd affiche les bibliothèques partagées requises par chaque programme ou la bibliothèque partagée spécifiée sur la ligne de commande.

Libtool

Contenu

Le package Libtool contient les programmes libtool et libtoolize. Il contient aussi la bibliothèque ltdl.

Description

libtool

libtool fournit des services généraux de support de construction des bibliothèques.

---

libtoolize

libtoolize fournit un moyen standard d'ajouter un support libtool à votre package.

Bibliothèque ltdl

Libtool fournit une petite bibliothèque, appelée `libltdl', qui a pour but de masquer les différentes difficultés des bibliothèques dlopening aux programmeurs.

Bin86

Contenu

Le paquet Bin86 contient les programmes as86, as86_encap, ld86, objdump86, nm86 et size86.

Description

as86

as86 est un assembleur pour les processeurs 8086...80386.

as86_encap

as86_encap est un script shell pour appeler as86 et convertir le binaire créé en fichier C prog.v à inclure ou à lier avec des programmes comme les installeurs de bloc de démarrage.

ld86

ld86 comprend uniquement les fichiers objet construit par l'assembleur as86; il peut soit les lier en un exécutable I&D séparé ou impur.

objdump86

Pas de description disponible.

nm86

Pas de description disponible.

size86

Pas de description disponible.

Lilo

Contenu

Le package Lilo contient le programme lilo.

Description

lilo installe le chargeur de démarrage de Linux, qui est utilisé pour lancer un système Linux.

Make

Contenu

Le package Make contient le programme make.

Description

make détermine automatiquement quels morceaux d'un gros programme ont besoin d'être recompilés, et exécute les commandes pour les recompiler.

Shellutils

Contenu

Le package Shellutils contient les 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.

Description

basename

basename supprime le répertoire et les suffixes des noms de fichier.

---

chroot

chroot lance une commande ou un shell interactif avec un répertoire racine particulier.

---

date

date affiche la date et l'heure courantes dans un format spécifié, ou positionne la date du système.

---

dirname

dirname supprime les suffixes autres que le répertoire du nom de fichier.

---

echo

echo affiche une ligne de texte.

---

env

env lance un programme dans un environnement modifié.

---

expr

expr évalue les expressions.

---

factor

factor affiche les facteurs premiers de tous les nombres entiers spécifiés.

false

false se termine toujours avec un code de statut indiquant une erreur.

groups

groups affiche les groupes auxquels appartient un utilisateur.

hostid

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

hostname

hostname positionne ou affiche le nom de l'hôte courant du système.

id

id affiche les GID et UID réels et effectifs d'un utilisateur ou de l'utilisateur courant.

logname

logname affiche le nom de connexion de l'utilisateur courant.

nice

nice lance un programme avec une priorité de cadencement modifiée.

nohup

nohup lance une commande insensible aux interruptions, avec une sortie autre qu'un terminal.

pathchk

pathchk vérifie si des noms de fichier sont valides ou portables.

pinky

pinky est un utilitaire léger qui recherche des informations sur un utilisateur particulier.

printenv

printenv affiche tout ou partie de l'environnement.

printf

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

pwd

pwd affiche le nom du répertoire courant de travail.

seq

seq affiche des nombres dans une certaine plage, avec un incrément spécifié.

sleep

sleep attend pendant un laps de temps spécifié.

stty

stty modifie et affiche les paramètres de ligne du terminal.

su

su lance un shell avec des ID d'utilisateur et de groupe substitués.

tee

tee lit depuis l'entrée standard et écrit sur la sortie standard et dans des fichiers.

---

test

test vérifie les types de fichier et compare les valeurs.

---

true

true se termine 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 sur le système.

---

uptime

uptime indique depuis combien de temps le système tourne.

---

users

users affiche le nom des utilisateurs actuellement connectés sur l'hôte courant.

---

who

who montre qui est connecté.

---

whoami

whoami affiche votre id utilisateur effectif.

---

yes

yes sort une chaîne à répétition, jusqu'à ce qu'il soit tué.

Suite de Masquage des mots de passe

Contenu

La Suite de Masquage des mots de passe contient les programmes chage, chfn, chsh, expiry, faillog, gpasswd, lastlog, login, newgrp, passwd, sg, su, chpasswd, dpasswd, groupadd, groupdel, groupmod, grpck, grpconv, grpunconv, logoutd, mkpasswd, newusers, pwck, pwconv, pwunconv, useradd, userdel, usermod et vipw.

Description

chage

chage modifie le nombre de jours entre deux changements de mot de passe et la date du dernier changement de mot de passe.

---

chfn

chfn modifie le nom complet de l'utilisateur, le numéro de travail, le poste de travail et le numéro de téléphone du domicile pour un compte utilisateur.

---

chsh

chsh modifie le shell de connexion de l'utilisateur.

---

expiry

Ce que fait ce programme est actuellement inconnu.

---

faillog

faillog formate le contenu du journal d'échec /var/log/faillog, et conserve le nombre et les limites des échecs.

---

gpasswd

gpasswd est utilisé pour administrer le fichier /etc/group.

---

lastlog

lastlog formate et affiche le contenu du journal de dernière connexion, /var/log/lastlog. Le nom de connexion, le port et l'heure de dernière connexion seront affichés.

---

login

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

---

newgrp

newgrp est utilisé pour modifier l'ID du groupe courant pendant une session de connexion.

---

passwd

passwd modifie les mots de passe des comptes utilisateur et groupe.

---

sg

sg exécute une commande avec un ID de groupe différent.

---

su

su modifie les ID effectifs de groupe et d'utilisateur en ceux d'un utilisateur. Ceci remplace le programme su installé avec le paquet Shellutils.

---

chpasswd

chpasswd lit un fichier contenant des paires nom d'utilisateur et mot de passe depuis l'entrée standard, et utilise ces informations pour mettre à jour un groupe d'utilisateurs existants.

---

dpasswd

dpasswd ajoute, supprime et met à jour les mots de passe d'appel pour les shells de connexion utilisateur.

---

groupadd

La commande groupadd crée un nouveau compte de groupe avec les valeurs spécifiées en ligne de commande et les valeurs par défaut du système.

groupdel

La commande groupdel modifie les fichiers de compte du système, supprimant toutes les entrées faisant référence au groupe.

groupmod

La commande groupmod modifie les systèmes de compte du système pour prendre en compte les changements spécifiés sur la ligne de commande.

grpck

grpck vérifie l'intégrité des informations d'authentification du système.

grpconv

grpconv convertit en fichiers de groupe masqués des fichiers de groupe normaux.

grpunconv

grpunconv convertit en fichiers de groupe normaux des fichiers de groupe masqués.

logoutd

logoutd applique les restrictions de temps de connexion et de port spécifiées dans /etc/porttime.

mkpasswd

mkpasswd lit un fichier dans le format donné par les options et le convertit dans le formet de fichier de base de données correspondant.

newusers

newusers lit un fichier contenant des paires nom d'utilisateur et mot de passe en clair, et utilise ces informations pour mettre à jour un groupe d'utilisateurs existants ou pour créer de nouveaux utilisateurs.

pwck

pwck vérifie l'intégrité des informations d'authentification du système.

---

pwconv

pwconv convertit en fichiers de mot de passe masqués des fichiers de mot de passe normaux.

---

pwunconv

pwunconv convertit en fichiers de mots de passe normaux des fichiers de mot de passe masqués.

---

useradd

useradd crée un nouvel utilisateur ou met à jour les informations par défaut du nouvel utilisateur.

---

userdel

userdel modifie les fichiers de compte du système, supprimant toutes les entrées faisant référence au nom de connexion spécifié.

---

usermod

usermod modifie les fichiers de compte du système pour refléter les modifications spécifiées sur la ligne de commande.

---

vipw et vigr

vipw et vigr éditent respectivement les fichiers /etc/passwd et /etc/group. Avec l'option –s, ils éditent les versions masquées de ces fichiers, respectivement /etc/shadow et /etc/gshadow.

Modutils

Contenu

Le package Modutils contient les programmes depmod, genksyms, insmod, insmod_ksymoops_clean, kerneld, kernelversion, ksyms, lsmod, modinfo, modprobe et rmmod.

Description

depmod

depmod gère les descriptions de dépendance pour les modules du noyau qui peuvent être chargés.

---

genksyms

genksyms lit (sur l'entrée standard) la sortie de gcc −E source.c et génère un fichier contenant des informations de version.

---

insmod

insmod installe dans le noyau utilisé un module qui peut être chargé.

---

insmod_ksymoops_clean

insmod_ksymoops_clean supprime les ksym et les modules sauvés qui n'ont pas été accédés depuis deux jours.

---

kerneld

kerneld exécute une action du noyau dans l'espace utilisateur (comme le chargement sur demande de modules).

---

kernelversion

kernelversion renvoie la version majeure du noyau utilisé.

---

ksyms

ksyms affiche les symboles exportés du noyau.

lsmod

lsmod fournit les informations sur tous les modules chargés.

modinfo

modinfo examine un fichier objet associé à un module du noyau et affiche toutes les informations qu'il peut y trouver.

modprobe

modprobe utilise un fichier de dépendance ressemblant à Makefile, créé par depmod, pour charger automatiquement le(s) module(s) pertinent(s) de l'ensemble des modules disponibles, dans l'arborescence des répertoires prédéfinie.

rmmod

rmmod décharge des modules chargés du noyau utilisé.

Procinfo

Contenu

Le package Procinfo contient le programme procinfo.

Description

procinfo recueille des données système depuis le répertoire /proc et les affiche de façon correctement formatée sur le périphérique de sortie standard.

Procps

Contenu

Le package Procps contient les programmes free, kill, oldps, ps, skill, snice, sysctl, tload, top, uptime, vmstat, w et watch.

Description

free

free affiche la quantité totale de mémoire, libre et utilisée, physique ou d'échange, dans le système, ainsi que la mémoire partagée et les mémoires tampon utilisées par le noyau.

---

kill

kills envoie des signaux aux processus.

---

oldps and ps

ps fournit un instantané des processus en cours.

---

skill

skill envoie des signaux aux processus répondant à un critère donné.

---

snice

snice modifie la priorité d'ordonnancement des processus répondant à un critère donné.

---

sysctl

sysctl modifie les paramètres du noyau en exécution.

---

tload

tload affiche un graphe de la charge moyenne du système courant sur le terminal spécifié (ou le terminal du processus tload si aucun autre n'est spécifié).

---

top

top fournit une vue de l'activité en cours du processeur en temps réel.

---

uptime

uptime affiche une ligne contenant les informations suivantes: l'heure courante, la durée de fonctionnement du système, le nombre d'utilisateurs actuellement connectés, et les moyennes de charge du système pour les 1, 5 et 15 dernières minutes.

---

vmstat

vmstat donne des informations sur les processus, la mémoire, la pagination, les entrées/sorties de blocs, les trappes et l'activité du cpu.

---

w

w affiche des informations sur les utilisateurs actuellement connectés à la machine, et leurs processus.

---

watch

watch lance une commande de façon répétitive, affichant sa sortie (le premier écran complet).

Vim

Contenu

Le paquet Vim contient les programmes ctags, etags, ex, gview, gvim, rgview, rgvim, rview, rvim, view, vim, vimtutor et xxd.

Description

ctags

ctags génère des fichiers de marquage pour le code source.

etags

etags fait la même chose que ctags, mais il peut générer des fichiers de références croisées qui listent des informations sur les différents objets source trouvés dans un ensemble de fichiers de langage.

ex

ex démarre vim en mode Ex.

gview

gview est la version GUI de view.

gvim

gvim est la version GUI de vim.

rgview

rgview est la version GUI de rview.

rgvim

rgvim est la version GUI de rvim.

rview

rview est une version restreinte de view. Aucune commande shell ne peut être lancée et Vim ne peut pas être interrompu.

---

rvim

rvim est la version restreinte de vim. Aucune commande shell ne peut être lancée et Vim ne peut pas être interrompu.

---

view

view lance vim en mode lecture seule.

---

vim

vim lance vim de la manière normale, par défaut.

---

vimtutor

vimtutor lance le tutorial de Vim.

---

xxd

xxd crée un dump hexadécimal, ou fait l'inverse.

Psmisc

Contenu

Le package Psmisc contient les programmes fuser, killall et pstree.

Description

fuser

fuser affiche le PID des processus utilisant les fichiers ou les systèmes de fichiers spécifiés.

killall

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

pstree

pstree montre les processus en cours sous la forme d'un arbre.

Sed

Contenu

Le package Sed contient le programme sed.

Description

sed est un éditeur de flux. Un éditeur de flux est utilisé pour opérer des transformations de texte basiques sur un flux d'entrée (un fichier ou une entrée depuis un pipe).

Sysklogd

Contenu

Le package Sysklogd contient les programmes klogd et syslogd.

Description

klogd

klogd est un démon système qui intercepte et journalise les messages du noyau de Linux.

syslogd

syslogd fournit une sorte de journalisation que de nombreux programmes modernes utilisent. Chaque message journalisé contient au moins une heure et un champ nom d'hôte, normalement un champ nom de programme également, mais cela dépend de la fidélité du programme qui journalise.

Sysvinit

Contenu

Le package Sysvinit contient les programmes pidof, last, lastb, mesg, utmpdump, wall, halt, init, killall5, poweroff, reboot, runlevel, shutdown, sulogin et telinit.

Description

pidof

pidof trouve les id des processus (pid) des programmes spécifiés et affiche ces id sur la sortie standard.

---

last

last recherche en arrière dans le fichier /var/log/wtmp (ou le fichier désigné par l'option –f) et affiche une liste de tous les utilisateurs s'étant connectés (et déconnectés) depuis la création du fichier.

---

lastb

lastb est identique à last, sauf que, par défaut, il affiche un journal du fichier /var/log/btmp, qui contient toutes les tentatives de connexion ayant échouées.

---

mesg

mesg contrôle l'accès à votre terminal par d'autres. Typiquement, ce programme est utilisé pour autoriser ou interdire à d'autres utilisateurs d'écrire sur votre terminal.

---

utmpdump

utmpdumps affiche le contenu d'un fichier (en général /var/run/utmp) sur la sortie standard dans un format compréhensible par l'utilisateur.

---

wall

wall envoie un message à toutes les personnes connectées ayant leur permission mesg positionnée à yes.

---

halt

halt note que le système est en train de s'arrêter dans le fichier /var/log/wtmp, puis dit au noyau soit de s'arrêter, soit de redémarrer, ou d'éteindre le système. Si halt ou reboot est appelé quand le système n'est pas dans un niveau d'exécution de 0 ou 6, shutdown sera appelé à la place (avec une option −h ou −r).

---

init

init est le père de tous les processus. Son rôle principal est de créer les processus à partir d'un script enregistré dans le fichier /etc/inittab. Ce fichier a en général des entrées sur chaque ligne indiquant à init d'engendrer des getty, pour que l'utilisateur puisse se connecter. Il contrôle aussi les processus autonomes requis par un système particulier.

---

killall5

killall5 est la commande killall de SystemV. Il envoie un signal à tous les processus, sauf les processus dans sa propre session, de manière à ne pas tuer le shell qui fait tourner le script qui l'a appelé.

---

poweroff

poweroff est équivalent à shutdown −h −p now. Il arrête l'ordinateur et l'éteint (si un BIOS compatible APM est utilisé, et qu'APM est activé dans le noyau).

---

reboot

reboot est équivalent à shutdown −r now. Il redémarre l'ordinateur.

---

runlevel

runlevel lit le fichier utmp du système (typiquement, /var/run/utmp) pour localiser l'enregistrement du niveau d'exécution, puis écrit les niveaux d'exécution du système, le précédent et l'actuel, sur la sortie standard, séparés par un seul espace.

---

shutdown

shutdown arrête le système de façon sécurisée. Tous les utilisateurs connectés sont avertis que le système va s'arrêter, et la connexion est bloquée.

---

sulogin

sulogin est invoqué par init quand le système passe en mode mono−utilisateur (ceci est réalisé par une entrée dans /etc/inittab). init tente également d'exécuter sulogin quand l'option −b lui est passée par le gestionnaire de démarrage (comme LILO).

---

telinit

telinit envoie les signaux appropriés à init, lui indiquant le niveau d'exécution sous lequel passer.

Tar

Contenu

Le package tar contient les programmes tar et rmt.

Description

tar

tar est un programme d'archivage conçu pour stocker et extraire des fichiers d'une archive au format d'un fichier tar.

rmt

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

Textutils

Contenu

Le paquet Textutils contient les programmes cat, cksum, comm, split, cut, expand, fmt, fold, head, join, md5sum, nl, od, paste, pr, ptx, sort, split, sum, tac, tail, tr, tsort, unexpand, uniq et wc.

Description

cat

cat concatène le(s) fichier(s) ou l'entrée standard vers la sortie standard.

---

cksum

cksum affiche la somme de contrôle CRC et le nombre d'octets de chaque fichier spécifié.

---

comm

comm compare deux fichiers triés ligne à ligne.

---

csplit

cplit découpe un fichier en morceaux séparés par un(des) motif(s) dans des fichiers xx01, xx02, ..., et affiche le nombre d'octets de chaque morceau sur la sortie standard.

---

cut

cut écrit les parties sélectionnées des lignes des fichiers spécifiés sur la sortie standard.

---

expand

expand convertit les tabulations dans les fichiers en espaces, en écrivant sur la sortie standard.

---

fmt

fmt reformate chaque paragraphe dans le(s) fichier(s) spécifié(s), en écrivant sur la sortie standard.

---

fold

fold emballe les lignes d'entrée de chaque fichier spécifié (entrée standard par défaut), en écrivant sur la sortie standard.

---

head

head affiche les xx (10 par défaut) premières lignes de chaque fichier spécifié sur la sortie standard.

---

join

join fusionne les lignes identiques de deux fichiers dans un fichier commun.

---

md5sum

md5sum affiche ou vérifie les sommes de contrôle MD5.

---

nl

nl écrit chaque fichier spécifié sur la sortie standard, en y ajoutant les numéros de ligne.

---

od

od écrit une représentation non−ambiguë, en octal par défaut, d'un fichier spécifié, sur la sortie standard.

---

paste

paste écrit des lignes, d'une manière consistant à prendre séquentiellement les lignes correspondantes de chaque fichier spécifié, séparées par des tabulations, sur la sortie standard.

---

pr

pr met en pages ou en colonnes des fichiers pour impression.

---

ptx

ptx produit un index de permutation du contenu d'un fichier.

---

sort

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

split

split extrait des morceaux de taille fixe d'un fichier d'entrée dans PREFIXaa, PREFIXab, ...

sum

sum écrit la somme de contrôle et le nombre de blocs de chaque fichier spécifié.

tac

tac écrit chaque fichier spécifié sur la sortie standard, la dernière ligne en premier.

tail

tail écrit les xx (10 par défaut) dernières lignes de chaque fichier spécifié sur la sortie standard.

tr

tr convertit, répète et/ou supprime des caractères de l'entrée standard, en écrivant sur la sortie standard.

tsort

tsort écrit des listes totalement ordonnées en accord avec l'ordonnancement partiel dans les fichiers spécifiés.

unexpand

unexpand convertit les espaces dans chaque fichier en tabulations, en écrivant sur la sortie standard.

uniq

uniq supprime toutes les lignes identiques successives sauf une depuis des fichiers ou l'entrée standard, et écrit dans des fichiers ou la sortie standard.

wc

wc affiche le nombre de lignes, mots et octets de chaque fichier spécifié, et une ligne de totalisation si plus d'un fichier est spécifié.

Util Linux

Contenu

Le paquet Util−linux contient les programmes arch, dmesg, kill, more, mount, umount, agetty, blockdev, cfdisk, ctrlaltdel, elvtune, fdisk, fsck.minix, hwclock, kbdrate, losetup, mkfs, mkfs.bfs, mkfs.minix, mkswap, sfdisk, swapoff, swapon, cal, chkdupexe, col, colcrt, colrm, column, cytune, ddate, fdformat, getopt, hexdump, ipcrm, ipcs, logger, look, mcookie, namei, rename, renice, rev, script, setfdprm, setsid, setterm, ul, whereis, write, ramsize, rdev, readprofile, rootflags, swapdev, tunelp et vidmode.

Description

arch

arch affiche l'architecture de la machine.

---

dmesg

dmesg est utilisé pour examiner ou vérifier la mémoire tampon circulaire du noyau (messages de démarrage du noyau).

---

kill

kill envoie un signal donné au processus spécifié.

---

more

more est un filtre pour paginer le texte écran par écran.

---

mount

mount monte un système de fichiers depuis un périphérique dans un répertoire (point de montage).

---

umount

umount démonte un système de fichiers monté.

---

agetty

agetty ouvre un port tty, demande un nom de connexion et appelle la commande /bin/login.

blockdev

blockdev vous permet d'appeler ioctls sur un périphérique de type bloc depuis la ligne de commande.

cfdisk

cfdisk est un manipulateur de table de partitions de disque basé sur libncurses.

ctrlaltdel

ctrlaltdel positionne la fonction associée à la combinaison de touche CTRL+ALT+DEL (remise à zéro logicielle ou matérielle).

elvtune

elvtune permet d'optimiser l'augmentation d'entrées/sorties par file de base de périphérique bloc.

fdisk

fdisk est un manipulateur de table de partitions de disque.

fsck.minix

fsck.minix effectue une vérification de cohérence pour le système de fichiers Linux MINIX.

hwclock

hwclock interroge et positionne l'horloge matérielle (appelée aussi horloge RTC ou BIOS).

kbdrate

kbdrate réinitialise le taux de répétition du clavier et son délai.

losetup

losetup configure et contrôle les périphériques de boucle.

---

mkfs

mkfs construit un système de fichiers Linux sur un périphérique, en général une partition de disque dur.

---

mkfs.bfs

mkfs.bfs crée un système de fichiers SCO bfs sur un périphérique, en général une partition de disque dur.

---

mkfs.minix

mkfs.minix crée un système de fichiers Linux MINIX sur un périphérique, en général une partition de disque dur.

---

mkswap

mkswap met en place une zone d'échange Linux sur un périphérique ou dans un fichier.

---

sfdisk

sfdisk est un manipulateur de table de partitions de disque.

---

swapoff

swapoff désactive les échanges de page mémoire pour les périphériques et les fichiers.

---

swapon

swapon active les échanges de page mémoire pour les périphériques et les fichiers.

---

cal

cal affiche un calendrier simple.

---

chkdupexe

chkdupexe trouve les exécutables en double.

---

col

col filtre les fins de ligne inversées de son entrée.

---

colcrt

colcrt filtre les sorties de nroff pour une prévisualisation via CRT.

---

colrm

colrm supprime les colonnes d'un fichier.

---

column

column met des listes en colonne.

---

cytune

cytune interroge et modifie le seuil d'interruption pour le pilote Cyclades.

---

ddate

ddate convertit les dates en système Grégorien en dates en système Discordien.

---

fdformat

fdformat réalise le formatage de bas niveau d'une disquette.

---

getopt

getopt sépare les options de commande, de la même manière que la commande C getopt.

---

hexdump

hexdump affiche les fichiers spécifiés, ou l'entrée standard, dans un format spécifié par l'utilisateur (ascii, décimal, hexadécimal, octal).

---

ipcrm

ipcrm supprime une ressource spécifiée.

---

ipcs

ipcs fournit des informations sur les utilitaires ipc.

---

logger

logger crée des entrées dans le journal système.

---

look

look affiche les lignes commençant par une chaîne donnée.

---

mcookie

mcookie génère des nombres magiques pour xauth.

---

namei

namei suit un chemin jusqu'à trouver un point terminal.

---

rename

rename renomme les fichiers.

---

renice

renice modifie la priorité des processus en exécution.

---

rev

rev inverse les lignes d'un fichier

script

script enregistre une session de terminal dans un fichier texte.

setfdprm

setfdprm positionne les paramètres fournis par l'utilisateur pour la disquette.

setsid

setsid lance les programmes dans une nouvelle session.

setterm

setterm positionne les attributs d'un terminal.

ul

ul lit un fichier et traduit les occurrences de blancs soulignés en séquence qui indiquent le soulignement sur le terminal utilisé.

whereis

whereis localise un binaire, un fichier source et une page de manuel pour une commande.

write

write envoie un message à un autre utilisateur.

ramsize

ramsize interroge et positionne la taille du disque mémoire.

rdev

rdev interroge et positionne le périphérique racine image, le périphérique d'échange, la taille du disque mémoire ou le mode vidéo.

readprofile

readprofile lit les informations de profil du noyau.

rootflags

rootflags interroge et positionne des informations supplémentaires utilisées lors du montage de la racine.

swapdev

swapdev interroge et positionne le périphérique d'échange.

tunelp

tunelp positionne différents paramètres pour le périphérique lp.

vidmode

vidmode interroge et positionne le mode vidéo.

Console−tools

Contenu

Le package Console−tools contient les programmes charset, chvt, consolechars, deallocvt, dumpkeys, fgconsole, fix_bs_and_del, font2psf, getkeycodes, kbd_mode, loadkeys, loadunimap, mapscrn, mk_modmap, openvt, psfaddtable, psfgettable, psfstriptable, resizecons, saveunimap, screendump, setfont, setkeycodes, setleds, setmetamode, setvesablank, showcfont, showkey, splitfont, unicode_start, unicode_stop, vcstime, vt−is−URF8 et writevt.

Description

charset

charset positionne un ACM à utiliser dans un des emplacements de jeu de caractères G0/G1.

chvt

chvt modifie le terminal virtuel en avant−plan.

codepage

Pas de description disponible.

consolechars

consolechars charge les fontes d'écran pour console EGA/VGA, les cartes de fonte d'écran et/ou les cartes de jeu de caractères d'application.

deallocvt

deallocvt désalloue les terminaux virtuels inutilisés.

dumpkeys

dumkeys renvoie les tables de traduction du clavier.

fgconsole

fgconsole affiche le nombre de terminaux virtuels actifs.

---

fix_bs_and_del

Pas de description disponible.

---

font2psf

Pas de description disponible.

---

getkeycodes

getkeycodes affiche la table de correspondance du noyau: "code saisi vers code touche".

---

kbd_mode

kbd_mode renvoie ou positionne le mode du clavier.

---

loadkeys

loadkeys charge les tables de traduction du clavier.

---

loadunimap

Pas de description disponible.

---

mapscrn

Pas de description disponible.

---

mk_modmap

Pas de description disponible.

---

openvt

openvt lance un programme sur un nouveau terminal virtuel.

---

psfaddtable

psfaddtable ajoute une table de caractères Unicode à une fonte de console.

---

psfgettable

psfgettable extrait une table de caractères Unicode contenue dans une fonte de console.

---

psfstriptable

psfstriptable supprime la table de caractères Unicode contenue dans une fonte de console.

---

resizecons

resizecons modifie l'image qu'a le noyau de la taille de la console.

---

saveunimap

Pas de description disponible.

---

screendump

Pas de description disponible.

---

setfont

Pas de description disponible.

---

setkeycodes

setkeycodes charge les entrées de la table de correspondance du noyau: "code saisi vers code touche".

---

setleds

setleds positionne les diodes du clavier.

setmetamode

setmetamode définit la gestion des méta−touches du clavier.

setvesablank

Pas de description disponible.

showcfont

showcfont affiche tous les caractères dans la fonte d'écran courante.

showkey

showkey examine les codes saisis et les codes touche envoyés par le clavier.

splitfont

Pas de description disponible.

unicode_start

unicode_start met la console en mode Unicode.

unicode_stop

Pas de description disponible.

vcstime

Pas de description disponible.

vt−is−UTF8

vt−is−UTF8 vérifie si le terminal virtuel courant est en mode UTF8 ou octet.

writevt

Pas de description disponible.

Console−data

Contenu

Le paquet console−data contient les fichiers de données utilisés et réclamés par le package console−tools.

Man−pages

Contenu

Le paquet Man−pages contient différentes pages de manuel qui ne sont pas fournies avec les paquets.

Description

Des exemples de pages de manuel fournies sont les pages de manuel décrivant toutes les fonctions C et C++, quelques fichiers /dev/ importants, etc.

Annexe B. Ressources

Introduction

Une liste de livres, de HOWTO et d'autres documents que vous pourrez trouver utile de télécharger ou d'acheter suit. Cette liste est simplement un petit choix pour commencer. Nous espérons pouvoir étendre cette liste au fur et à mesure que nous découvrirons d'autres livres ou documents utiles.

Livres

- Guide de l'Administrateur Réseau Linux publié par O'Reilly. ISBN: 1−56502−087−2 (Linux Network Administrator's Guide)
- Lancer Linux publié par O'Reilly. ISBN: 1−56592−151−8 (Running Linux)

HOWTOs et Guides

Tous les HOWTO suivants peuvent être téléchargés depuis le site du Projet de Documentation Linux à http://www.linuxdoc.org.

- Guide de l'Administrateur Réseau Linux (Linux Network Administrator's Guide)
- HOWTO Du démarrage à Bash (Powerup2Bash−HOWTO)

Autres

- Les différentes pages de man ou d'information qui sont fournies avec les paquets.

Annexe C. Sites de téléchargement officiels

Ci−dessous, vous trouverez la liste des packages du chapitre 3 avec leur site original de téléchargement. Ceci pourra vous aider à trouver plus rapidement une version plus récente d'un package.

- Bash (2.04): ftp://ftp.gnu.org/gnu/bash/
- Binutils (2.10.1): ftp://ftp.gnu.org/gnu/binutils/
- Bzip2 (1.0.1): ftp://sourceware.cygnus.com/pub/bzip2/
- Diff Utils (2.7): ftp://ftp.gnu.org/gnu/diffutils/
- File Utils (4.0): ftp://ftp.gnu.org/gnu/fileutils/
- GCC (2.95.2): ftp://ftp.gnu.org/gnu/gcc/
- Noyau Linux (2.2.18): ftp://ftp.kernel.org/pub/linux/kernel/
- Glibc (2.1.3): ftp://ftp.gnu.org/gnu/glibc/
- Glibc−crypt (2.1.3): ftp://ftp.gnu.org/gnu/glibc/
- Glibc−linuxthreads (2.1.3): ftp://ftp.gnu.org/gnu/glibc/
- Patch de Glibc (2.1.3): ftp://packages.linuxfromscratch.org/pub/common−packages
- Grep (2.4.2): ftp://ftp.gnu.org/gnu/grep/
- Gzip (1.2.4a): ftp://ftp.gnu.org/gnu/gzip/
- Make (3.79.1): ftp://ftp.gnu.org/gnu/make/
- Sed (3.02): ftp://ftp.gnu.org/gnu/sed/
-

Sh−utils (2.0): ftp://ftp.gnu.org/gnu/sh−utils/

- Tar (1.13): ftp://ftp.gnu.org/gnu/tar/
- Patch de Tar (1.13): http://sourceware.cygnus.com/bzip2/
- Text Utils (2.0): ftp://ftp.gnu.org/gnu/textutils/
- MAKEDEV (2.5): ftp://ftp.ihg.uni−duisburg.de/Linux/system/
- Bison (1.28): ftp://ftp.gnu.org/gnu/bison/
- Mawk (1.3.3): ftp://ftp.whidbey.net/pub/brennan/
- Patch (2.5.4): ftp://ftp.gnu.org/gnu/patch/
- Find Utils (4.1): ftp://ftp.gnu.org/gnu/findutils/
- Patch de Find Utils (4.1): ftp://packages.linuxfromscratch.org/pub/common−packages
- Ncurses (5.2): ftp://ftp.gnu.org/gnu/ncurses/
- Less (358): ftp://ftp.gnu.org/gnu/less/
- Groff (1.16.1): ftp://ftp.gnu.org/gnu/groff/
- Man (1.5h1): ftp://ftp.win.tue.nl/pub/linux−local/utils/man/
- Perl (5.6.0): http://www.perl.com
- M4 (1.4): ftp://ftp.gnu.org/gnu/m4/
- Texinfo (4.0): ftp://ftp.gnu.org/gnu/texinfo/
-

Autoconf (2.13): ftp://ftp.gnu.org/gnu/autoconf/

- Automake (1.4): ftp://ftp.gnu.org/gnu/automake/
- Flex (2.5.4a): ftp://ftp.gnu.org/non−gnu/flex/
- File (3.33): ftp://ftp.gw.com/mirrors/pub/unix/file/
- Libtool (1.3.5): ftp://ftp.gnu.org/gnu/libtool/
- Bin86 (0.15.4): http://www.cix.co.uk/~mayday/
- Gettext (0.10.35): ftp://ftp.gnu.org/gnu/gettext/
- Console−tools (0.2.3): ftp://metalab.unc.edu/pub/Linux/system/keyboards/
- Patch de Console−tools (0.2.3): ftp://packages.linuxfromscratch.org/pub/common−packages
- Console−data (1999.08.29): ftp://metalab.unc.edu/pub/Linux/system/keyboards/
- E2fsprogs (1.19): ftp://download.sourceforge.net/pub/sourceforge/e2fsprogs/
- Ed (0.2): ftp://ftp.gnu.org/gnu/ed/
- Ld.so (1.9.9): ftp://packages.linuxfromscratch.org/pub/common−packages
- Lilo (21.6): ftp://brun.dyndns.org/pub/linux/lilo
- Modutils (2.4.0): ftp://ftp.kernel.org/pub/linux/utils/kernel/modutils
- Vim−rt (5.7) 1 073 Ko: ftp://ftp.vim.org/pub/editors/vim/unix/
- Vim−src (5.7) 1 202 Ko: ftp://ftp.vim.org/pub/editors/vim/unix/
-

Procinfo (17): ftp://ftp.cistron.nl/pub/people/svm/

- Procps (2.0.7): ftp://people.redhat.com/johnsonm/procps/
- Psmisc (19): ftp://lrcftp.epfl.ch/pub/linux/local/psmisc/
- La suite Masquage des mots de passe (20000902): ftp://ftp.ists.pwr.wroc.pl/pub/linux/shadow/
- Sysklogd (1.4): ftp://sunsite.unc.edu/pub/Linux/system/daemons/
- Patch de Sysklogd (1.4): ftp://packages.linuxfromscratch.org/pub/common−packages/
- Sysvinit (2.78): ftp://ftp.cistron.nl/pub/people/miquels/sysvinit/
- Patch de Sysvinit (2.78): ftp://packages.linuxfromscratch.org/pub/common−packages/
- Util Linux (2.10r): ftp://ftp.win.tue.nl/pub/linux−local/utils/util−linux/
- Man−pages (1.33): ftp://ftp.win.tue.nl/pub/linux−local/manpages/
- Netkit−base (0.17): ftp://ftp.uk.linux.org/pub/linux/Networking/netkit/
- Net−tools (1.57): http://www.tazenda.demon.co.uk/phil/net−tools/