Linux From Sratch

Gerard Beekmans Version 2.2, 3 avril 2000

Ce document décrit comment créer de toute pièce un système Linux à partir des sources des applications et d'une distribution déjà installée.

Table des matières

1 Introduction

1.1 De quoi s'agit-il ?

J'ai utilisé de nombreuses distributions Linux différentes, toutefois aucune d'elles ne m'a jamais vraiment satisfait. Parfois l'agencement des scripts de démarrage ne me convenait pas, parfois la configuration par défaut des programmes n'était pas à mon goût. J'en suis venu à penser que je ne serai jamais complètement satisfait par un système Linux tant que je n'en construirais pas un moi-même. L'idéal était de partir uniquement des fichiers sources de chaque application et donc de n'utiliser aucun paquetage préconstruit ni disquette ou CDROM installant un système de base.

A première vue, cette idée semblait saugrenue, voire presque irréalisable. La plupart des problèmes qui freinaient ma progression venaient de mon manque de connaissances au sujet de certains programmes et mécanismes. Ainsi après avoir rencontré et résolu tous les problèmes liés à la dépendance, à la compilation, etc, je parvins à mettre en place un système Linux opérationnel. J'ai nommé alors ce système LFS, qui signifie en anglais Linux From Scratch (NDT : approximativement "Linux de Zéro").

En pratique, nous mettrons en place ce système idéal à partir d'une distribution Linux déjà en place .

1.2 Mises à jour

La dernière version anglaise de ce document est disponible à tout moment à l'adresse :

http ://www.linuxfromscratch.org

1.3 Historique de ce document

2.1.5 - 26 mars 2000

Il ne s'agit pas d'une liste exhaustive des modifications. Comme la version 2.0 est une mise à jour majeure, seuls les changements majeurs sont mentionnés.

- Modification de la structure de l'arborescence - LFS est conforme à la norme FHS. Peut-être pas à 100%.
- Nouvelle méthode d'installation de la Glibc
- Nouvelle méthode d'installation de GCC
- Suppression de l'utilisation de paquetages préconstruits Debian.
- Méthode d'installation des applications complètement révisée - élimination de la nécessité d'utiliser des paquetages liés statiquement dans le chapitre 6.1.
- Divers bogues corrigés concernant l'installation des applications
- Quelques programmes supplémentaires du paquetage util-linux installés
- Ajout de l'installation du programme Bzip2
- Explications plus détaillées concernant $LFS - ce qu'il faut et ce qu'il ne faut pas faire.
- Procédure d'installation simplifiée pour tous les paquetages des chapitres 5 à 9.1
- Installation de la Glibc et de GCC, possédant jusqu'à présent leur propre chapitre, incluse dans le chapitre 7.
- Chapitre concernant les serveurs Internet modifié : création de deux chapitres : Démons Réseau et Clients Réseau. Le chapitre concernant Internet a été partagé entre ceux-ci.
- Echange des chapitres 13 et 14 (X et Internet) et concaténation des chapitres concernant X et WindowMaker.
- Un nouveau programme Man est utilisé. Il est plus facile à manipuler et à configurer que le man-db utilisé précédemment. Les deux versions réalisent sensiblement le même travail.
- Ajout du chapitre 13 : les ressources. Ce chapitre contient les références de nombreux livres et HOWTOs qui vous seront utilisés au cours ou au terme de la construction du système LFS.

- Chapitre 3 : lien bzip2 corrigé
- Chapitre 7.2.42 : méthode d'installation d'Util-Linux simplifiée.
- Chapitre 3.1 : emplacement de procps modifié
- Chapitre 7.2 : échange de l'installation de Vim et d'Util-Linux (car nous avons besoin d'un éditeur de textes pour installer Util-Linux)
- Chapitre 7.3.33 : installation de procps corrigée
- Chapitre 5.2 : fichier inittab modifié de manière à ce que le système ne signale plus de fichiers manquants au démarrage de la machine.
- Chapitre 6 : réécriture de l'installation du noyau
- Chapitre 10.3 : Script de démarrage d'Apache modifié
- Chapitre 10.3.2 : section concernant la modification du fichier httpd.conf supprimée. Elle n'est plus nécessaire. Ajout des fichiers de /usr/apache/man dans le fichier /usr/share/misc/man.conf.
- Chapitre 11.1 : Correction du fichier Makefile du paquetage mailx afin de simplifier la procédure d'installation.
- Chapitre 11.3.1 : Ajout de l'option –shared lors de l'exécution de configure afin que la Zlib soit installée comme une bibliothèque dynamique plutôt que statique.
- Chapitre 11.6 : Lynx lié à la bibliothèque Ncurses plutôt que Slang.
- Chapitre 12 : Le fichier man_db.config de la nouvelle application man-db contient toujours le répertoire X11/man.

2.2 - 3 avril 2000

- Les domaines linuxfromscratch.org et linuxfromscratch.com sont opérationnels. Tous les liens vers huizen.dds.nl/glb et tts.ookhio.dds.nl ont été remplacés par les liens appropriés vers www.linuxfromscratch.org.
- Après le redémarrage de la machine au chapitre 7.3, la partition swap est activée avant de commencer à compiler les applications.

1.4 Listes de diffusion

Il existe deux listes auxquelles vous pouvez vous inscrire : lfs-discuss et lfs-announce. La première est une liste publique non modérée concernant tout ce qui a trait à ce document. La seconde est une liste publique modérée. Tout le monde peut s'y inscrire mais personne ne peut y poster de message, à l'exception des modérateurs. Cette liste est principalement dédiée aux annonces de nouvelles versions de ce document.

Si vous êtes inscrits à la liste de diffusion lfs-discuss, vous n'avez pas besoin de vous inscrire à lfs-announce. En effet, tout ce qui est envoyé sur la liste lfs-announce est également envoyé sur lfs-discuss.

1.4.1 Inscription

Pour vous inscrire à une liste, envoyez un email, dont le corps contient soit *subscribe lfs-discuss*, soit *subscribe lfs-announce,* à *mailto :majordomo@fist.org*

Majordomo vous enverra une demande de confirmation. Celle-ci contiendra un code d'authentification et la marche à suivre pour poursuivre la demande d'inscription. Une fois que vous aurez répondu à ce mail, vous serez alors inscrit à la liste.

1.4.2 Résilier une inscription

Pour résilier votre inscription à une liste, envoyez un email, dont le corps contient soit *unsubscribe lfs-discuss* soit *unsubscribe lfs-announce* à l'adresse suivante : *mailto :majordomo@fist.org*

1.5 Contact

Envoyez de préférence toutes vos questions à la liste de diffusion. Si vous avez besoin de me contacter personnellement, envoyez un mail à l'adresse suivante : *mailto :gerard@linuxfromscratch.org*

2 Conventions utilisées dans ce document

2.1 A propos de $LFS

Lisez ce qui suit attentivement : tout au long de ce document, vous rencontrerez fréquemment des références à une variable dont le nom est $LFS. Celle-ci doit être systématiquement remplacée par le point de montage de la partition sur laquelle vous créez votre système LFS. La façon de créer et de monter cette partition sera expliquée en détail au chapitre 4. Dans mon cas, j'utilise le répertoire `/mnt/hda5` comme point de montage pour la partition LFS. Ainsi toutes les occurrences de $LFS de ce document sont à remplacer par `/mnt/hda5`. Donc si je lis la commande : `cp inittab $LFS/etc`, je taperai en réalité `cp inittab /mnt/hda5/etc`

Vous devez effectuer cette substitution quel que soit l'endroit où vous rencontrez $LFS, que ce soit dans une commande, ou dans un fichier que vous devez créer ou éditer.

Il existe une deuxième méthode qui consiste à définir la variable d'environnement $LFS. De cette manière, vous tapez les commandes comportant une référence à $LFS telles quelles et le shell procédera à la substitution lui-même. Pour définir $LFS, utilisez la commande : `export LFS=/mnt/hda5`

Dans ce cas, vous saisissez sans modification la commande `cp inittab $LFS/etc` rencontrée dans ce document et le shell exécutera en réalité `cp inittab /mnt/hda5/etc` .

Si vous utilisez cette méthode, il est très important que la variable d'environnement $LFS soit toujours déclarée. Si ce n'est pas le cas, les occurrences de $LFS présentes dans les commandes que vous taperez seront ignorées et le reste sera exécuté. Ainsi la commande `cp inittab $LFS/etc` sera traduite par le shell en `cp inittab /etc` avec pour conséquence d'écraser le fichier `inittab` de votre système Linux. Un fichier comme `inittab` n'est pas très difficile à reconstruire, mais imaginez ce qui se passera si $LFS n'est pas défini au cours de l'installation de la bibliothèque C. Vous endommagerez gravement votre système et à moins que vous ne soyez un expert Linux, vous serez obligé de tout réinstaller. Je vous conseille donc d'utiliser la méthode de substitution manuelle. Ainsi au pire, si vous vous trompez en saisissant le nom du point de montage de votre partition LFS, le système renverra une erreur du type "*no such file or directory*" sans aucun dommage pour votre système. Ne dites pas que je ne vous ai pas prévenu ;-)

2.2 Comment télécharger les applications

Tout au long de ce document, je supposerai que vous avez enregistré tous les paquetages que vous avez téléchargé dans un sous-répertoire de $LFS/usr/src.

J'utilise par convention le répertoire `$LFS/usr/src/sources`. "`sources`" contient les répertoires 0 à 9 et a à z. Ainsi le paquetage Sysvinit-2.78.tar.gz est sauvegardé dans `$LFS/usr/sources/s/` , tandis que bash-3.02.tar.gz se trouve dans `$LFS/usr/src/sources/b/` . Vous n'êtes pas obligé de suivre ce modèle, ce n'est qu'un exemple. Il est par contre préférable de ne pas conserver de paquetages dans `$LFS/usr/src` car ce répertoire servira à décompresser les archives au fur et à mesure que nous en aurons besoin.

Le prochain chapitre contient la liste des paquetages que vous devez télécharger. A ce point du document, la partition destinée à recevoir le système LFS n'existe pas. Vous devez donc sauvegarder ces paquetages dans un endroit temporaire et vous rappelez de les copier de cet endroit vers $LFS/usr/src/<répertoiredevotrechoix> au terme du chapitre 4.

2.3 Comment installer les applications

Avant de pouvoir faire quelque chose des paquetages, vous devez les désarchiver. Généralement, il s'agit d'archives au format tar/gz (l'extension est soit .tar.gz soit .tgz) ou au format tar/bz (extension .tar.bz2). Le compresseur Bzip2 est un peu plus efficace que gzip mais n'est pas toujours installé par défaut. Ainsi si vous téléchargez des archives au format bz, vérifiez que Bzip2 est disponible sur votre système. Je n'expliquerai qu'une seule fois, dans ce paragraphe, comment désarchiver les paquetages, aussi lisez attentivement ce qui suit.

- Commencez par *copier* les paquetages depuis l'emplacement où vous les avez enregistrés vers le répertoire `$LFS/usr/src`.
- Si l'archive est au format tar/gz, désarchivez-la grâce à la commande : `tar xvfz filename.tar.gz ; rm filename.tar.gz` ou `tar xvfz filename.tgz ; rm filename.tgz`
- S'il s'agit d'une archive tar/bz2, utilisez la commande : `tar -use-compress-prog=bzip2 -xvf filename.tar.bz2 ; rm filename.tar.bz2`
- Enfin si l'archive est au format tar (pas de compression), utilisez la commande : `tar xvf filename.tar ; rm filename.tar`

Remarquez que nous effaçons le paquetage juste après l'avoir désarchivé car nous n'en avons plus besoin. C'est pourquoi vous devez *copier* l'archive et non la déplacer. Si vous la déplacez, la désarchivez et l'effacez, vous devrez la télécharger à nouveau lorsque vous en aurez besoin.

Lorsque le paquetage est désarchivé, un nouveau répertoire est créé dans le répertoire actif (`$LFS/usr/src` si vous suivez ce document à la lettre). Vous devez vous positionner dans ce répertoire avant de poursuivre l'installation. Finalement tout ce qui précède peut se résumer à "Désarchivez xxx". Ainsi lorsque vous lirez cette phrase, copiez l'archive dans `$LFS/usr/src`, désarchivez-la grâce à la commande tar et positionnez-vous dans le répertoire créé. Ensuite vous pouvez continuer à suivre les instructions.

Après avoir installé un paquetage, vous pouvez en faire deux choses. Soit vous effacez le répertoire qui contient ses sources, soit vous le conservez. Quelle que soit votre décision, je ne m'en porterai pas plus mal. Toutefois si vous avez besoin de réutiliser un paquetage plus tard dans le document (tous les logiciels du chapitre 7.2 seront réinstallés dans le chapitre 7.3), vous devez d'abord effacer le répertoire et désarchiver à nouveau l'archive. Si vous ne le faites pas, vous pourriez avoir quelques problèmes inhérents aux paramètres utilisés lors de la précédente compilation. Ceux-ci s'appliquaient à votre système Linux normal mais pas au système LFS. Même un simple `make clean` ne suffit pas à nettoyer totalement les sources de l'arborescence. Le script `configure` génère également des fichiers répartis dans divers sous-répertoires et ces fichiers sont rarement effacés au cours de l'exécution de `make clean`.

3 Les paquetages que vous devez télécharger

Vous trouverez ci-dessous la liste des paquetages que vous devez absolument télécharger pour mener à bien la méthode exposée dans ce document et obtenir un système LFS opérationnel. J'ai également indiqué les serveurs sur lesquels ils sont disponibles, mais c'est à vous de vous assurer que vous téléchargez bien la dernière version des applications. Les versions indiquées dans la liste correspondent à celles utilisées dans ce document et ont donc été testées. Ainsi si vous rencontrez le moindre problème que vous ne pouvez résoudre avec les archives que vous avez récupérées, téléchargez la version indiquée dans ce document (dans le cas où vous avez téléchargé une version plus récente).

3.1 Applications indispensables

Sysvinit (2.78) : *ftp ://ftp.cistron.nl/pub/people/miquels/sysvinit/*

Bash (2.03) : *ftp ://ftp.gnu.org/gnu/bash/*

Linux Kernel (2.2.14) : *ftp ://ftp.kernel.org/pub/linux/kernel/*

Binutils (2.9.1) : *ftp ://ftp.gnu.org/gnu/binutils/*

Bzip2 (0.9.5d) : *http ://sourceware.cygnus.com/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/*

Glibc (2.1.3) : *ftp ://ftp.gnu.org/gnu/glibc/*

Glibc-crypt (2.1.2) : *ftp ://ftp.gwdg.de/pub/linux/glibc/*

Glibc-linuxthreads (2.1.3) : *ftp ://ftp.gnu.org/gnu/glibc/*

Grep (2.4) : *ftp ://ftp.gnu.org/gnu/grep/*

Gzip (1.2.4) : *ftp ://ftp.gnu.org/gnu/gzip/*

Make (3.78.1) : *ftp ://ftp.gnu.org/gnu/make/*

Sed (3.02) : *ftp ://ftp.gnu.org/gnu/sed/*

Shell Utils (2.0) : *ftp ://ftp.gnu.org/gnu/sh-utils/*

Tar (1.13) : *ftp ://ftp.gnu.org/gnu/tar/*

Text Utils (2.0) : *ftp ://ftp.gnu.org/gnu/textutils/*

Util Linux (2.10f) : *ftp ://ftp.win.tue.nl/pub/linux/utils/util-linux/*

Bison (1.28) : *ftp ://ftp.gnu.org/gnu/bison/*

Mawk (1.3.3) : *ftp ://ftp.whidbey.net/pub/brennan/*

Find Utils (4.1) : *ftp ://ftp.gnu.org/gnu/findutils/*

Ncurses (5.0) : *ftp ://ftp.gnu.org/gnu/ncurses/*

Less (340) : *ftp ://ftp.gnu.org/gnu/less/*

Perl (5.005_03) : *ftp ://ftp.gnu.org/gnu/perl/*

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/gnu/flex/*

E2fsprogs (1.18) : *ftp ://tsx-11.mit.edu/pub/linux/packages/ext2fs/*

File (3.26) : *http ://www.linuxfromscratch.org/download/file-3.26-lfs.tar.gz*

Groff (1.15) : *ftp ://ftp.gnu.org/gnu/groff/*

Ld.so (1.9.9) : *ftp ://tsx-11.mit.edu/pub/linux/packages/GCC/*

Libtool (1.3.4) : *ftp ://ftp.gnu.org/gnu/libtool/*

Linux86 (0.14.3) : *http ://www.linuxfromscratch.org/download/linux86-0.14.3-lfs.tar.gz*

Lilo (21) : *ftp ://sunsite.unc.edu/pub/Linux/system/boot/lilo/*

Shadow Password Suite (19990827) : *ftp ://piast.t19.ds.pwr.wroc.pl/pub/linux/shadow/*

Man (1.5h1) : *ftp ://ftp.win.tue.nl/pub/linux-local/utils/man/*

Modutils (2.3.9) : *ftp ://ftp.ocs.com.au/pub/modutils/*

Termcap (1.3) : *ftp ://ftp.gnu.org/gnu/termcap/*

Procinfo (17) : *ftp ://ftp.cistron.nl/pub/people/svm/*

Procps (2.0.6) : *ftp ://people.redhat.com/johnsonm/procps/*

Psmisc (19) : *ftp ://lrcftp.epfl.ch/pub/linux/local/psmisc/*

Start-stop-daemon (0.4.1) : *http ://www.linuxfromscratch.org/download/ssd-0.4.1-lfs.tar.gz*

Sysklogd (1.3.31) : *ftp ://sunsite.unc.edu/pub/Linux/system/daemons/*

Vim (5.6) : *ftp ://ftp.vim.org/pub/editors/vim/unix/*

3.2 Applications facultatives

Toutes les applications listées ci-dessous, traitées à partir du chapitre 13, ne sont pas indispensables. C'est à vous de déterminer les paquetages dont vous avez besoin. Par exemple, si vous souhaitez pouvoir accéder à Internet par le biais du système LFS, vous devez installer les utilitaires concernant le courrier électronique, le web, etc.

Netkit-base (0.17) : *ftp ://ftp.uk.linux.org/pub/linux/Networking/netkit-devel/*

Net-tools (1.54) : *http ://www.tazenda.demon.co.uk/phil/net-tools/*

Procmail (3.14) : *ftp ://ftp.procmail.org/pub/procmail/*

Sendmail (8.9.3) : *ftp ://ftp.sendmail.org/pub/sendmail/*

Mailx (8.1.1) : *http ://www.linuxfromscratch.org/download/mailx-8.1.1-fixed.tar.gz*

Mutt (1.0i) : *ftp ://ftp.mutt.org/pub/mutt/*

Fetchmail (5.2.0) : *http ://www.tuxedo.org/~esr/fetchmail/*

Netkit-telnet (0.17) : *ftp ://ftp.uk.linux.org/pub/linux/Networking/netkit-devel/*

Proftpd (1.2.0pre9) : *ftp ://ftp.tos.net/pub/proftpd/*

Netkit-ftp (0.17) : *ftp ://ftp.uk.linux.org/pub/linux/Networking/netkit-devel/*

Apache (1.3.11) : *http ://www.apache.org/dist/*

Zlib Library (1.1.3) : *http ://www.cdrom.com/pub/infozip/zlib/*

Lynx (2.8.2) : *http ://www.slcc.edu/lynx/release/*

PPP (2.3.11) : *ftp ://cs.anu.edu.au/pub/software/ppp/*

Xfree86 (3.3.5) : *ftp ://ftp.xfree86.org/pub/XFree86/*

libPropList (0.9.1) : *ftp ://ftp.windowmaker.org/pub/libs/*

libXpm (4.7) : *ftp ://sunsite.unc.edu/pub/Linux/libs/X/*

libpng (1.0.3) : *http ://www.cdrom.com/pub/png/*

libtiff (3.4) : *ftp ://ftp.sgi.com/graphics/tiff/*

libjpeg (6b) : *http ://www.ijg.org/*

libungif (4.1.0) : *ftp ://prtr-13.ucsc.edu/pub/libungif/*

WindowMaker (0.61.1) : *ftp ://ftp.windowmaker.org/pub/release/*

4 Préparer le nouveau système

4.1 Comment allons-nous procéder

Nous allons construire le système LFS en utilisant une distribution Linux déjà en place, comme par exemple Debian, SuSE, Slackware, Mandrake, RedHat, etc. Vous n'avez pas besoin de disquette de démarrage particulière. Nous utiliserons tous les outils du système déjà en place (compilateur, éditeur de liens, éditeur de textes et quelques autres).

Si vous ne disposez pas d'un système déjà en place, vous ne pourrez pas utiliser la méthode décrite dans ce document. Il est donc dans ce cas indispensable que vous en installiez un, peu importe la distribution, tant qu'elle n'est pas trop vieille. Je pense que toute distribution âgée d'au maximum un an devrait convenir. Par ailleurs, vous vous éviterez pas mal de soucis si cette distribution est basée sur glibc-2.0 ou mieux. Je n'ai pas pris en compte les systèmes basés sur Libc5 puisque je n'ai pas eu l'occasion d'en utiliser un.

4.2 Créer une nouvelle partition

Pour construire notre nouveau système Linux, nous avons besoin d'une partition Linux vide. Prévoyez un minimum de 500 Mo, cela devrait suffire. Vous pouvez éventuellement utiliser une partition de 250 Mo, si vous souhaitez construire un système Linux simple (sans système X Window ou outils dédiés à la navigation sur Internet). Si vous disposez déjà d'une partition Linux Native vide, vous pouvez passer cette sous-section.

Démarrez le programme `fdisk` (ou tout partitionneur de votre choix, du moment qu'il autorise la création de partitions Linux) en indiquant le disque dur sur lequel vous souhaitez créer la nouvelle partition Linux (par exemple `fdisk /dev/hda` si la nouvelle partition doit être créée sur le disque dur maître du premier contrôleur IDE). Créez une partition Linux Native, mettez à jour la table de partition et quittez `fdisk`. Si un message vous indique de redémarrer l'ordinateur afin d'être sûr que la table de partition est mise à jour, faites-le. Notez le nom de la nouvelle partition (par exemple `hda5`) car vous en aurez besoin plus tard. Cette partition sera appelée *partition LFS* dans la suite du document.

4.3 Créer un système de fichiers `ext2` sur la nouvelle partition

Une fois que la partition est créée, nous devons y installer un système de fichiers `ext2`. Pour ce faire, utilisez la commande `mke2fs`. Indiquez le nom de la nouvelle partition comme unique option et le système de fichiers sera créé. Ainsi si la partition porte le nom `hda5`, utilisez la commande `mke2fs /dev/hda5`.

4.4 Monter la nouvelle partition

Une fois que le système de fichiers est créé, il est prêt à être utilisé. Tout ce qu'il vous reste à faire pour le rendre accessible est de le monter. En reprenant notre exemple précédent, si vous montez la partition sous `/mnt/hda5`, vous pouvez vous positionner dans ce répertoire et faire ce que vous voulez. Je supposerai par la suite que vous avez monté la partition LFS dans un sous-répertoire de `/mnt`. Vous avez pu constater que, par convention, je monte mes partitions dans des répertoires portant le nom de cette partition (`hda5` monté sous `/mnt/hda5`). Libre à vous de faire comme bon vous semble, voire d'utiliser le répertoire `/mnt`.

Voici en résumé la méthode pour monter la partition LFS :

- Crééz le répertoire `/mnt` s'il n'existe pas encore
- Crééz le répertoire `/mnt/xxx` où `xxx` est à remplacer par le nom de la partition LFS.
- Montez la partition LFS en lançant : `mount /dev/xxx /mnt/xxx` où `xxx` est à remplacer par le nom de la partition LFS.

Ce répertoire (`/mnt/xxx`) correspond à la variable $LFS expliquée plus haut. Je vous rappelle donc que si vous lisez "`cp inittab $LFS/etc`", en réalité vous devez taper "`cp inittab /mnt/xxx/etc`" où `xxx` est à remplacer par le nom de la partition LFS.

4.5 Créer les répertoire

Créez l'arborescence de base, au standard FHS, du système LFS. Si vous souhaitez plus d'informations sur le standard FHS, consultez le site *http ://www.pathname.com/fhs/*. Tapez les commandes suivantes :

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

Maintenant que les répertoires sont créés, copiez les archives téléchargées au chapitre 3 dans un sous-répertoire de $LFS/usr/src (vous devez créer ce sous-répertoire vous-même).

4.6 Copier le répertoire `/dev`

Nous pourrions utiliser la commande `mknod` pour créer les fichiers du répertoire `$LFS/dev`, mais cela prend du temps. C'est pourquoi je vous conseille de simplement recopier le répertoire `/dev` vers la partition $LFS. Pour ce faire, utilisez la commande suivante, tous les droits et liens symboliques originaux seront conservés.

```
cp -av /dev $LFS
chown root.root $LFS/dev/*
```

5 Rendre le système LFS amorçable

5.1 Installer Sysvinit

Normalement, lorsque le noyau a terminé de se charger et d'initialiser divers composants systèmes, il tente de charger un programme appelé `init` qui terminera le processus de démarrage. Le paquetage utilisé sur la majorité sinon tous les systèmes Linux est Sysvinit et c'est justement lui que nous utilisons pour notre système LFS.

- Désarchivez Sysvinit
- Positionnez-vous dans le répertoire src

– Editez le fichier `Makefile`
– Quelque part dans ce fichier, mais avant la règle "all", ajoutez cette ligne : *ROOT = $LFS*
– Préfixez toutes les occurrences `/dev` des quatre dernières lignes par *$(ROOT)*

Après avoir fait cela, les quatre dernières lignes du fichier doivent ressembler à :

```
@if [! -p $(ROOT)/dev/initctl ]; then \
echo "Creating $(ROOT)/dev/initctl";  \
rm -f $(ROOT)/dev/initctl;  \
mknod -m 600 $(ROOT)/dev/initctl p; fi
```

– Installez le paquetage en exécutant :

```
make -e LDFLAGS=-static; make install
```

5.2 Configurer Sysvinit

Pour que Sysvinit fonctionne, vous devez créer un fichier de configuration approprié. Créez le fichier `$LFS/etc/inittab` contenant :

```
# Begin /etc/inittab

id:2:initdefault:

ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now

1:2345:respawn:/sbin/sulogin

# End /etc/inittab
```

5.3 Créer les fichiers passwd et group

Comme vous pouvez le voir dans le fichier `inittab`, lorsque nous amorçons le système, `init` démarre le programme sulogin qui vous demandera le mot de passe root. Cela signifie que nous avons besoin de créer un fichier de mots de passe sur le système LFS.

– Créez le fichier `$LFS/etc/passwd` contenant : *root :s394ul1Bkvmq2 :0 :0 :root :/root :/bin/bash*
– Créez le fichier `$LFS/etc/group` contenant : *root : :0 :*

Le mot de passe encodé dans le fichier `passwd` est : *lfs123*

Lorsque le système vous demandera le mot de passe root, c'est celui-ci que vous devez entrer.

5.4 Installer le shell Bash

Après que `sulogin` ait validé le mot de passe root, il tentera de lancer un interpréteur de commandes (plus simplement appelé *shell*), généralement Bash. Nous devons donc le compiler. Mais comme aucune bibliothèque n'est encore disponible sur le système, nous devons le lier statiquement, ainsi que nous avons procédé pour Sysvinit.

– Désarchivez Bash

– Installez Bash en lançant :

```
./configure --enable-static-link
make; make -e prefix=$LFS/usr install
mv $LFS/usr/bin/bash $LFS/bin
cd $LFS/bin; ln -s bash sh
```

5.5 Ajouter une entrée à LILO

Afin de pouvoir démarrer le système LFS, nous devons mettre à jour le fichier`/etc/lilo.conf`. Ajoutez-y les lignes suivantes :

```
image=<image du noyau courant>
   label=<label>
   root=$LFS
   read-only
```

Remplacez <image du noyau courant> par le nom du fichier contenant l'image du noyau utilisé par votre système. Vous pouvez remplacer <label> par ce que bon vous semble. Pour ma part, j'utilise comme label "lfs". Ce label correspond à ce que vous devez taper au démarrage de la machine lorsque LILO vous demande le système que vous souhaitez démarrer.

Maintenant lancez `lilo` afin de mettre à jour l'amorceur du système.

5.6 Tester le système

Après avoir effectué les opérations précédentes, vous pouvez vérifier s'il est possible de se connecter au système LFS en redémarrant la machine. N'oubliez pas à l'affichage de LILO de taper le label que vous avez choisi plus haut. Si tout se déroule correctement, `sulogin` vous demandera le mot de passe root. Saisissez-le. Vous devriez alors vous retrouver sous le shell. Remarquez que vous ne pourrez pas arrêter le système par l'intermédiaire de la commande `shutdown`. Bien que ce programme soit présent, il vous renverra comme erreur "You don't exist. Go away" si vous essayez de l'exécuter. Cela signifie que le système est incapable de localiser le fichier des mots de passe (`/etc/passwd`). `Sulogin`, bien que statiquement lié, dépend quand même de la bibliothèque NSS (Name Server Switch) qui fait partie de la bibliothèque C que nous installerons ultérieurement. La bibliothèque NSS est utilisée dans ce cas pour indiquer à `sulogin` où se trouve le fichier de mots de passe.

Pour le moment, vous pouvez redémarrer le système en utilisant `reboot -f` . Cette commande outrepassera le redémarrage du système normalement opéré par `shutdown` et redémarrera instantanément. Comme le système de fichiers est monté en lecture seule, cette opération n'endommagera rien. Par contre vous pourriez être informé au prochain démarrage que le système de fichiers n'a pas été correctement démonté et qu'il faut lancer `e2fsck` pour vérifier s'il n'y a pas de problème.

6 Installer un noyau

6.1 Remarque au sujet de ftp.kernel.org

A la section 2, j'ai indiqué que vous pouviez télécharger un noyau depuis ftp ://ftp.kernel.org/ . Toutefois ce site est souvent chargé et ses administrateurs vous encouragent à utiliser un miroir près de chez vous. Dans ce cas, l'adresse est du type ftp ://ftp.<code pays>.kernel.org/ (par exemple ftp.fr.kernel.org).

6.2 Configurer le noyau

- Renommez le répertoire `/usr/src/linux` (même s'il s'agit d'un lien symbolique) en utilisant `mv /usr/src/linux /usr/src/linux-old`
- Effacez les liens `/usr/include/linux` et `/usr/include/asm` grâce à la commande `rm -r /usr/include/linux /usr/include/asm`
- Désarchivez les sources du noyau dans le répertoire `/usr/src/` (un répertoire `/usr/src/linux` sera créé)
- Créez les liens symboliques `/usr/include/linux` et `/usr/include/asm` en exécutant :

```
cd /usr/include
ln -s ../src/linux/include/linux linux
ln -s ../src/linux/include/asm-<cpu> asm
```

Regardez dans le répertoire `/usr/src/linux/include` quels sont les sous-répertoires asm-* présents. Choisissez celui qui correspond à votre plateforme. Par exemple s'il s'agit d'un PC Intel, vous lancerez la commande `ln -s ../src/linux/include/asm-i386 asm`

- Choisissez une méthode pour configurer le noyau (consultez le fichier README pour plus détails concernant la méthode de configuration) et assurez-vous de ne rien configurer en module pour le moment car nous ne disposerons pas des applications nécessaires pour charger des modules avant un bout de temps.
- Après en avoir terminé avec la configuration du noyau, lancez `make dep`
- Compilez-le grâce à la commande `make bzImage`
- Copiez le fichier `arch/<cpu>/boot/bzImage` dans `/boot` (ou à l'endroit utilisé par votre système Linux pour enregistrer les images noyau)
- Vous pouvez éventuellement renommer le fichier `/boot/bzImage` en quelque chose comme `/boot/lfskernel`
- Copiez l'arborescence complète des sources du noyau sur la partition LFS grâce à la commande : `cp -av /usr/src/linux $LFS/usr/src`
- Créez les liens symboliques `$LFS/usr/include/linux` et `$LFS/usr/include/asm` en tapant :

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

6.3 Mettre à jour LILO

- Editez le fichier `/etc/lilo.conf` et rendez-vous à la section LFS
- Remplacez le nom de l'image du noyau par `lfskernel` (ou tout autre nom que vous lui avez donné)
- Lancez `lilo` pour mettre à jour l'amorceur.

6.4 Tester le système

Redémarrez votre système sous LFS. Vérifiez que le nouveau noyau n'effectue aucune opération anormale (comme un crash).

7 Installer les applications de base

Dans ce chapitre, nous installerons toutes les applications qui constituent un système Linux de base. Au terme de ce chapitre vous disposerez d'un système Linux pleinement opérationnel. Les chapitres suivants traitent de sujets facultatifs comme par exemple la configuration du réseau, des serveurs et clients Internet (telnet, ftp, http, email) et du système X Window. Vous êtes libre d'ignorer autant de chapitres qu'il vous plaira. Ainsi si vous ne souhaitez pas vous connecter à Internet, vous n'aurez que peut d'intérêt à consulter le chapitre dédié.

Il y a un certain nombre de paquetages que vous devez impérativement installer avant de pouvoir mettre en place les applications de base. En effet un script typique aura besoin de programmes comme rm, grep, sed, mv, cat, cp, diff. Vous devez également pouvoir décompresser des archives, compiler et éditer les liens des applications. Tous les outils correspondants doivent être installés avant toute chose. Ils seront de plus liés statiquement. En effet votre système Linux utilise peut-être une version de bibliothèque C différente de celle qui sera utilisée par le système LFS. Or les applications que vous allez installer dans cette section seraient théoriquement liées avec la bibliothèque C de votre système normal. Ceci pourrait engendrer des conflits de bibliothèque si vous exécutez ces applications sous le système LFS. C'est pourquoi nous devons les lier statiquement dans un premier temps. Puis au cours de la mise en place du système LFS nous lierons à nouveau ces applications avec la bibliothèque C utilisée par le système LFS.

7.1 A propos des symboles de débogage

Par défaut, chaque programme et bibliothèque est compilé avec les symboles de débogage. Cela signifie que vous pouvez exécuter ce programme ou cette bibliothèque par l'intermédiaire d'un débogueur et que ce dernier produira une sortie de l'exécution plus compréhensible. Toutefois ces symboles de débogage augmentent la taille du fichier produit de manière significative. Par convention, tous les logiciels installés dans ce document incluront ces symboles (puisque je ne sais pas si la majorité des lecteurs fait ou non du débogage). Libre à vous de supprimer par la suite les symboles de débogage en utilisant le programme `strip` de la manière suivante : `strip -strip-debug nomdufichier` (le fichier doit être au format elf ou a.out). Vous pouvez utiliser des jokers si vous souhaitez traiter plusieurs fichiers d'un coup (exemple : `strip -strip-debug $LFS/usr/bin/*`).

Si vous vous demandez l'influence des symboles de débogage sur la taille des fichiers, voici quelques statistiques :

- Binaire Bash statique avec symboles de débogage : 2.3MB
- Binaire Bash statique sans symbole de débogage : 645KB
- Binaire Bash dynamique avec symboles de débogage : 1.2MB
- Binaire Bash dynamique sans symbole de débogage : 478KB
- $LFS/usr/lib (fichiers glibc and gcc) avec symboles de débogage : 87MB
- $LFS/usr/lib (fichiers glibc and gcc) sans symbole de débogage : 16MB

Ces tailles sont données à titre indicatif et dépendent grandement du compilateur et de la version de bibliothèque C utilisés, mais au final les résultats avec ou sans symboles de débogage seront comparables. Pour ma part, au terme de ce chapitre et après avoir supprimé les symboles de débogage des fichiers exécutables et bibliothèques, j'ai récupéré environ 102 Mo d'espace disque. Différence remarquable. Mais elle sera encore plus grande en réalisant cette opération au terme du document.

7.2 Préparer le système LFS pour installer les applications de base

7.2.1 Installer Binutils

- Désarchivez binutils

– Installez le paquetage en tapant :

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

7.2.2 Installer Bzip2

– Désarchivez Bzip2
– Editez le fichier Makefile
– Trouvez les lignes commençant par : $(CC) $(CFLAGS) -o
– Remplacez-les par : $(CC) $(CFLAGS) $(LDFLAGS) -o
– Installez le paquetage en lançant :

```
make -e LDFLAGS=-static
make -e PREFIX=$LFS/usr install
cd $LFS/usr/bin
mv bunzip2 bzip2 $LFS/bin
```

7.2.3 Installer Diffutils

– Désarchivez diffutils
– Installez le paquetage en lançant :

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

Ce paquetage peut causer des erreurs relatives à l'édition statique des liens sur certaines plate-formes. Si c'est votre cas, vous pouvez télécharger une version corrigée du paquetage à l'adresse : *http ://www.linuxfromscratch.org/download/diffutils-2.7-fixed.tar.gz*

7.2.4 Installer Fileutils

– Désarchivez fileutils
– Installer le paquetage en lançant :

```
./configure --disable-nls
make -e LDFLAGS=-static
make -e prefix=$LFS/usr install
cd $LFS/usr/bin
mv chgrp chmod chown cp dd df ln ls mkdir mknod mv rm rmdir sync $LFS/bin
```

7.2.5 Installer GCC sur le système normal si nécessaire

Afin de compiler Glibc-2.1.3, le compilateur gcc-2.95.2 doit être installé sur le système Linux normal. Remarquez que n'importe quelle version supérieure ou égale à 2.8 fera l'affaire. Néanmoins de nombreux systèmes basés sur glibc-2.0 disposent de gcc-2.7.2.3. Celui-ci n'est pas apte à compiler et ne doit pas être utilisé pour compiler glibc-2.1. L'installation de la version 2.95.2 de gcc sur le système normal n'écrasera pas la version déjà en place. Avant de procéder à cette installation, assurez-vous que vous en avez réellement besoin. Vérifiez le numéro de version de votre compilateur au moyen de la commande `gcc -version`. S'il est supérieur ou égal à 2.8, tout va bien, vous n'avez pas besoin de faire la moindre mise à jour. Cela dit, si vous rencontrez des problèmes lors de la compilation de glibc, n'hésitez pas à installer gcc-2.95.2

- Désarchivez GCC
- Installez le paquetage en lançant :

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

7.2.6 Installer GCC sur le système LFS

- Désarchivez GCC
- Installez le paquetage en lançant :

```
mkdir $LFS/usr/src/gcc-build;cd $LFS/usr/src/gcc-build
../gcc-2.95.2/configure --enable-languages=c --disable-nls
make -e LDFLAGS=-static bootstrap
make -e prefix=$LFS/usr local_prefix=$LFS/usr install
```

Création des liens symboliques nécessaires Il est nécessaire de créer quelques liens symboliques afin que chaque application puisse trouver le compilateur et le préprocesseur. Ainsi certains programmes lancent le programme cc, d'autres gcc, certains programmes supposent que cpp est dans `/lib` (c'est-à-dire `/usr/lib` sur notre système LFS) et d'autres, dans `/usr/bin`.

- Créez les liens symboliques suivants grâce aux commandes :

```
cd $LFS/lib; ln -s ../usr/lib/gcc-lib/<host>/2.95.2/cpp cpp
cd $LFS/usr/lib; ln -s gcc-lib/<host>/2.95.2/cpp cpp
cd $LFS/usr/bin; ln -s gcc cc
```

Remplacez <host> par le répertoire où réside gcc-2.95.2 (i686-unknown-linux dans mon cas). Généralement, vous trouverez deux répertoires différents.

7.2.7 Installer Glibc

Remarque concernant le paquetage glibc-crypt :

```
-*-*-*-*-*-
Glibc-crypt n'est pas inclus dans la distribution principale
 de la bibliothèque GNU C du fait de restrictions gouvernementales, principalement
 en France, Russie et Etats-Unis, concernant la diffusion et l'utilisation de
 logiciels de cryptage. Consultez la note "Legal Problems" du manuel pour plus
 d'informations.
En particulier, les Etats-Unis interdisent l'exportation de ce logiciel
```

```
sans licence, même par Internet. Ainsi ne le téléchargez pas sur le site FTP
principal de la FSF (ftp.gnu.org) si vous résidez à l'extérieur des Etats-Unis.
Ce logiciel a été complètement développé à l'extérieur des Etats-Unis.-*-*-*-*-*-
```

"Ce logiciel" fait référence au paquetage glibc-crypt disponible sur ftp ://ftp.gwdg.de/pub/linux/glibc. Cette loi n'affecte que les personnes ne résidant pas aux Etats-Unis. Il n'est pas interdit d'importer des logiciels basés sur DES, donc si vous résidez aux Etats-Unis, vous êtes libre de l'importer depuis ce site allemand.

- Désarchivez Glibc
- Copiez les archives glibc-crypt et glibc-linuxthreads dans le répertoire créé au cours du désarchivage de Glibc
- Décompressez ces archives depuis ce point mais ne vous positionnez pas dans les répertoires créés par cette nouvelle décompression. Contentez-vous de décompresser ces archives.
- Créez un nouveau fichier `configparms` contenant :

```
# Begin configparms
slibdir=/lib
sysconfdir=/etc
# End configparms
```

- Si votre système Linux normal dispose d'une version de gcc apte à compiler glibc, installez ce paquetage en lançant :

```
mkdir $LFS/usr/src/glibc-build;cd $LFS/usr/src/glibc-build
../glibc-2.1.3/configure --enable-add-ons
make; make install_root=$LFS install
```

- Si votre système Linux normal ne dispose pas d'une version de gcc apte à compiler glibc et que vous avez installé gcc-2.95.2 pour ce faire, procédez comme suit :

```
mkdir $LFS/usr/src/glibc-build; cd $LFS/usr/src/glibc-build
CC=/usr/gcc2952/bin/gcc ../glibc-2.1.3/configure --enable-add-ons
make; make install_root=$LFS install
```

Copier les anciens fichiers de la bibliothèque NSS Si votre système Linux normal est basé sur glibc-2.0.x, vous devez copier les fichiers de la bibliothèque NSS sur le système LFS. Quelques programmes liés statiquement ont besoin de cette bibliothèque, notamment ceux recherchant les noms, identifiants et groupes des utilisateurs. Vous pouvez vérifier quelle est la version de bibliothèque C utilise grâce à la commande : `ls -l libc.so.*`

Votre système est basé sur glibc-2.0 si l'affichage produit ressemble à : `/lib/libc.so.6 -> libc-2.0.7.so`

Votre système est basé sur glibc-2.1 si l'affichage produit ressemble à : `/lib/libc.so.6 -> libc-2.1.2.so`

Si vous avez un fichier libc-2.0.x.so (où x est le numéro de micro version comme par exemple 7), copiez les fichiers de la bibliothèque NSS grâce à la commande : `cp -av /lib/*nss* $LFS/lib`

7.2.8 Installer grep

- Désarchivez grep
- Installez le paquetage en lançant :

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

Ce paquetage peut causer des erreurs relatives à l'édition statique des liens sur certaines plateformes. Si c'est votre cas, vous pouvez télécharger une version corrigée du paquetage à l'adresse : *http ://www.linuxfromscratch.org/download/grep-2.4-fixed.tar.gz*

7.2.9 Installer gzip

- Désarchivez gzip
- Installez le paquetage en lançant :

```
./configure
make -e LDFLAGS=-static
make -e prefix=$LFS/usr install
cd $LFS/usr/bin
mv gunzip gzip $LFS/bin
```

Ce paquetage peut causer des erreurs de compilation sur certaines plateformes. Si c'est votre cas, vous pouvez télécharger une version corrigée du paquetage à l'adresse : *http ://www.linuxfromscratch.org/download/gzip-1.2.4-fixed.tar.gz*

7.2.10 Installer Make

- Désarchivez Make
- Installez le paquetage en lançant :

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

7.2.11 Installer Sed

- Désarchivez sed
- Installez le paquetage en lançant :

```
./configure
make -e LDFLAGS=-static
make -e prefix=$LFS/usr install
mv $LFS/usr/bin/sed $LFS/bin
```

Ce paquetage peut causer des erreurs relatives à l'édition statique des liens sur certaines plateformes. Si c'est votre cas, vous pouvez télécharger une version corrigée du paquetage à l'adresse : *http ://www.linuxfromscratch.org/download/sed-3.02-fixed.tar.gz*

7.2.12 Installer Sh-utils

- Désarchivez sh-utils
- Installez le paquetage en lançant :

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

7.2.13 Installer Tar

- Désarchivez tar
- Installez le paquetage en lançant :

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

7.2.14 Installer Textutils

- Désarchivez textutils
- Installez le paquetage en lançant :

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

7.2.15 Installer Util-linux

- Désarchivez util-linux
- Installez le paquetage en lançant :

```
./configure
cd lib;make
cd ../mount;make -e LDFLAGS=-static
cp mount umount $LFS/bin
cp swapon $LFS/sbin
```

7.3 Installer les applications de base

L'installation de toutes les applications est franchement répétitive et vous pensez peut-être qu'il serait plus simple de donner un mode d'emploi générique et d'ajouter quelques explications concernant les paquetages qui ne suivraient pas ce mode d'emploi. Bien que je sois d'accord avec vous sur ce point, j'ai délibérément choisi d'écrire le mode d'emploi pour chaque paquetage. Cela permet d'éviter toute confusion ou erreur.

Avant d'aller plus loin dans l'installation, vous devez démarrer le système LFS. Pour ce faire, vous devez déterminer quelle partition est utilisée comme swap par votre système. Vous trouverez cette information dans le fichier `/etc/fstab`. La ligne concernée ressemble à : */dev/hda6 none swap sw 0 0* . Le quatrième champ de la ligne doit contenir 'sw'. Cela indique qu'il s'agit de la partition de swap. Vous devez alors noter quelque part la désignation de cette partition (comme par exemple `/dev/hda6` dans notre cas). Lorsque vous avez déterminé quelle partition est celle de swap, vous pouvez redémarrer la machine.

7.3.1 Remonter la partition et activer le swap

Avant d'être en mesure d'installer des applications supplémentaires depuis le système LFS, vous devez remonter la partition en mode lecture/écriture. Il est également nécessaire d'activer la partition de swap de manière à éviter tout risque de saturation de mémoire au cours de compilations importantes (par exemple la compilation de gcc) :

```
mount -o remount,rw / /
/sbin/swapon <swap device>
```

7.3.2 Installer GCC

– Désarchivez GCC et installez-le en tapant :

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

7.3.3 Installer Bison

– Désarchivez bison et installez-le en tapant :

```
./configure --datadir=/usr/share/bison
make; make install
```

7.3.4 Installer Mawk

– Désarchivez mawk et installez-le en tapant :

```
./configure
make; make install
cd /usr/bin; ln -s mawk awk
```

7.3.5 Installer Findutils

– Désarchivez findutils et installez-le en tapant :

```
./configure
make; make install
```

Ce paquetage peut causer des erreurs de compilation sur certaines plateformes. Si c'est votre cas, vous pouvez télécharger une version corrigée du paquetage à l'adresse : *http ://www.linuxfromscratch.org/download/findutils-4.1-fixed.tar.gz*

7.3.6 Installer Termcap

– Désarchivez Termcap et installez-le en tapant :

```
./configure
make; make install
```

7.3.7 Installer Ncurses

– Désarchivez ncurses et installez-le en tapant :

```
./configure --with-shared
make; make install
```

7.3.8 Installer Less

– Désarchivez Less et installez-le en tapant :

```
./configure
make; make install
mv /usr/bin/less /bin
```

7.3.9 Installer Perl

– Désarchivez Perl et installez-le en tapant :

```
./configure.gnu
make; make install
```

Remarquez que nous sautons l'étape '`make test`', car pour le moment le système n'est pas en mesure de lancer le test de Perl. Nous faisons donc confiance à la compilation.

7.3.10 Installer M4

– Désarchivez M4 et installez-le en tapant :

```
./configure
make; make install
```

7.3.11 Installer Texinfo

– Désarchivez Texinfo et installez-le en tapant :

```
./configure
make; make install
```

7.3.12 Installer Autoconf

– Désarchivez Autoconf et installez-le en tapant :

```
./configure
make; make install
```

7.3.13 Installer Automake

– Désarchivez Automake et installez-le en tapant :

```
./configure
make install
```

7.3.14 Installer Bash

– Désarchivez Bash et installez-le en tapant :

```
./configure
make; make install
mv /usr/bin/bash /bin
```

7.3.15 Installer Flex

– Désarchivez Flex et installez-le en tapant :

```
./configure
make; make install
```

7.3.16 Installer Binutils

– Désarchivez Binutils et installez-le en tapant :

```
./configure
make; make install
```

7.3.17 Installer Bzip2

– Désarchivez Bzip2 et installez-le en tapant :

```
make; make install
cd /usr/bin; mv bunzip2 bzip2 /bin
```

7.3.18 Installer Diffutils

– Désarchivez Diffutils et installez-le en tapant :

```
./configure
make; make install
```

7.3.19 Installer E2fsprogs

– Désarchivez E2fsprogs et installez-le en tapant :

```
./configure
make; make install
mv /usr/sbin/mklost+found /sbin
```

7.3.20 Installer File

– Désarchivez File et installez-le en tapant :

```
./configure
make; make install
```

7.3.21 Installer Fileutils

– Désarchivez Fileutils et installez-le en tapant :

```
./configure
make; make install
cd /usr/bin
mv chgrp chmod chown cp dd df ln ls mkdir mknod mv rm rmdir sync /bin
```

7.3.22 Installer Grep

– Désarchivez Grep et installez-le en tapant :

```
./configure
make; make install
```

7.3.23 Installer Groff

– Désarchivez Groff et installez-le en tapant :

```
./configure
make; make install
```

7.3.24 Installer Gzip

– Désarchivez Gzip et installez-le en tapant :

```
./configure
make; make install
cd /usr/bin; mv z* gunzip gzip /bin
```

7.3.25 Installer Ld.so

– Désarchivez Ld.so et installez-le en tapant :

```
cd util; make ldd ldconfig
cp ldd /bin; cp ldconfig /sbin
rm /usr/bin/ldd
```

7.3.26 Installer Libtool

– Désarchivez Libtool et installez-le en tapant :

```
./configure
make; make install
```

7.3.27 Installer Linux86

– Désarchivez Linux86 et installez-le en tapant :

```
cd as
make; make install
cd ../ld
make ld86; make install
```

7.3.28 Installer Lilo

– Désarchivez Lilo et installez-le en tapant :

```
make; make install
```

7.3.29 Installer Make

– Désarchivez Make et installez-le en tapant :

```
./configure
make; make install
```

7.3.30 Installer Sh-Utils

– Désarchivez Sh-utils et installez-le en tapant :

```
./configure
make; make install
cd /usr/bin
mv date echo false pwd stty su true uname hostname /bin
```

7.3.31 Installer Shadow Password Suite

– Désarchivez Shadow et installez-le en tapant :

```
./configure
make; make install
cd etc
cp limits login.access login.defs.linux shells suauth /etc
mv /etc/login.defs.linux /etc/login.defs
cd /usr/sbin
mv chpasswd dpasswd groupadd groupdel groupmod logoutd mkpasswd  \
newusers useradd userdel usermod grpck pwck vipw grpconv grpunconv  \
pwconv pwunconv /sbin
```

7.3.32 Installer Man

– Désarchivez Man et installez-le en tapant :

```
./configure -default
make all; make install
```

7.3.33 Installer Modutils

– Désarchivez Modutils et installez-le en tapant :

```
./configure
make; make install
```

7.3.34 Installer Procinfo

– Désarchivez Procinfo et installez-le en tapant :

```
make; make install
```

7.3.35 Installer Procps

– Désarchivez Procps et installez-le en tapant :

```
gcc -O3 -Wall -Wno-unused -c watch.c
make; make -e XSCPT="" install
mv /usr/bin/kill /bin
```

7.3.36 Installer Psmisc

– Désarchivez Psmisc et installez-le en tapant :

```
make; make install
```

7.3.37 Installer Sed

– Désarchivez Sed et installez-le en tapant :

```
./configure
make; make install
mv /usr/bin/sed /bin
```

7.3.38 Installer le démon start-stop

– Désarchivez start-stop-daemon et installez-le en tapant :

```
make start-stop-daemon
cp start-stop-daemon /sbin
cp start-stop-daemon.8 /usr/share/man/man8
```

7.3.39 Installer Sysklogd

– Désarchivez Sysklogd et installez-le en tapant :

```
make; make install
```

7.3.40 Installer Sysvinit

– Désarchivez Sysvinit et installez-le en tapant :

```
cd src
make; make install
```

7.3.41 Install Tar

– Désarchivez Tar et installez-le en tapant :

```
./configure
make; make install
mv /usr/bin/tar /bin
```

7.3.42 Installer Textutils

– Désarchivez Textutils et installez-le en tapant :

```
./configure
make; make install
mv /usr/bin/cat /bin
```

7.3.43 Installer Vim

– Décompressez les archives Vim-rt et Vim-src et installez-les en tapant :

```
./configure
make; make install
```

7.3.44 Installer Util-linux

– Désarchivez le paquetage Util-linux
– Editez le fichier `MCONFIG`, cherchez et modifiez les variables suivantes comme indiqué ci-dessous :

```
HAVE_PASSWD=yes
HAVE_SLN=yes
HAVE_TSORT=yes
```

– Installez le paquetage en lançant :

```
groupadd -g 5 tty
./configure
make; make install
```

7.4 Suppression des anciens fichiers de la bibliothèque NSS

Si vous avez copié les fichiers de la bibliothèque NSS de votre système Linux normal vers le système LFS (car votre système normal utilise glibc-2.0), il est temps de les supprimer en tapant :

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

7.5 Configurer les applications

Maintenant que toutes les applications sont installées, vous devez en configurer certaines afin qu'elles fonctionnent correctement.

7.5.1 Configurer Glib

Vous devez créer le fichier `/etc/nsswitch.conf`. Bien que glibc fournisse des valeurs par défaut si ce fichier est absent ou corrompu, celles-ci sont incompatibles avec le fonctionnement réseau du système que nous aborderons plus tard. Par ailleurs, vous devez également régler le fuseau horaire. Voici donc comment procéder :

– Créez un nouveau fichier `/etc/nsswitch.config` contenant :

```
# Begin /etc/nsswitch.conf
passwd: files
group: files
shadow: files

hosts: files dns
networks: files

protocols: db files
services: db files
ethers: db files
rpc: db files

netgroup: db files
# End /etc/nsswitch.conf
```

- Lancez le script tzselect et répondez aux questions concernant votre fuseau horaire.
- Lorsque vous avez terminé, le programme vous fournit l'emplacement d'un fichier dont vous aurez besoin ci-dessous.
- Créez un lien symbolique pour l'heure locale en lançant : `ln -s /usr/share/zoneinfo/<emplacement donné par tzselect> /etc/localtime`

La sortie de `tzselect` ressemble à "EST5EDT" ou "Canada/Eastern". Le lien symbolique que vous devez créer avec cette information sera dans ce cas : `ln -s /usr/share/zoneinfo/EST5EDT /etc/localtime` ou `ln -s /usr/share/zoneinfo/Canada/Eastern /etc/localtime`

7.5.2 Configurer LILO

Il n'est pas nécessaire de créer le fichier de configuration de lilo de toute pièce. Utilisez plutôt le fichier de votre système Linux normal. Ce fichier est différent pour chaque machine et je ne peux donc pas vous donner de modèle. Comme il n'y a aucune raison de modifier la configuration LILO actuelle de votre système, contentez-vous de copier le fichier original sans le modifier.

- Créez le répertoire `/mnt/original`
- Montez votre système Linux normal sur ce point de montage en lançant `mount /dev/xxx /mnt/original` (remplacez `/dev/xxx` par le nom de la partition Linux normale).
- Copiez le fichier de configuration de lilo ainsi que les images de noyau qu'il utilise :

```
cp /mnt/original/etc/lilo.conf /etc
cp /mnt/original/boot/* /boot
```

Si votre système Linux normal ne sauvegarde pas les images des noyaux dans le répertoire `/mnt/original/boot`, cherchez alors dans le fichier `lilo.conf` l'emplacement adéquat et copiez les images dans le répertoire où lilo s'attend à les trouver. Ainsi si lilo indique que le noyau se trouve dans `/boot/kernel`, vous devez copier le noyau contenu dans `/mnt/original/boot/kernel` vers `/boot/kernel`. Vous pouvez également copier le noyau dans `/boot` et modifier le fichier `lilo.conf` pour tenir compte de ce changement. Ces deux méthodes sont valables, à vous de choisir.

7.5.3 Configurer Sysklogd

- Créez le répertoire `/var/log` grâce à : `mkdir /var/log`

– Créez le fichier `/etc/syslog.conf` contenant :

```
# Begin /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 *

# End /etc/syslog.conf
```

7.5.4 Configurer Shadow Password Suite

Ce paquetage contient notamment les utilitaires nécessaires pour modifier les mots de passe des utilisateurs et ajouter ou supprimer de nouveaux groupes et utilisateurs. Je n'expliquerai pas ce que '*password shadowing*' signifie. Vous trouverez toutes les informations utiles dans les fichiers de documentation et HOWTO. Il y a cependant une chose que vous devez garder à l'esprit : si vous décidez d'utiliser le support '*shadow*', tous les programmes qui ont besoin de vérifier les mots de passe (xdm, les démons ftp, pop3d, etc) doivent être compatible avec ce support.

Même si vous pensez que vous n'avez pas besoin d'utiliser les '*shadowed passwords*' (NDT : littéralement mots de passe masqués) (après avoir lu la documentation à ce sujet), vous devez néanmoins conserver l'archive correspondante car les outils qu'elle contient fonctionnent également sans '*shadow password*'. Notez que vous pouvez choisir à tout moment d'activer ou de désactiver les mots de passe '*shadow*'.

Consultez la section 5 du fichier Shadow-Password-HOWTO. Vous y apprendrez comment tester si le système '*shadow*' fonctionne et si ce n'est pas le cas, comment le désactiver. S'il ne fonctionne pas et que vous n'avez effectué aucune modification, vous vous retrouverez avec un système totalement inutilisable lorsque vous vous voudrez vous connecter. Vous pouvez facilement remédier à ce problème en passant comme paramètre `init=/sbin/sulogin` au noyau. Il sera alors nécessaire de décompresser l'archive util-linux, d'aller dans le répertoire login-utils, de compiler le programme `login` et de le copier à la place de `/bin/login`. La situation n'est jamais vraiment désespérée, du moins pas sous Linux. Cela dit, vous pouvez vous épargner bien des désagréments en vous documentant et en effectuant les tests nécessaires. ;-)

7.5.5 Configurer Sysvinit

Après avoir modifié comme suit le fichier `/etc/inittab`, vous aurez la possibilité de vous connecter au système par l'intermédiaire de `agetty` ou `login`. `Sulogin` ne sera plus utilisé pour les connexions normales.

– Editez le fichier `/etc/inittab` et modifiez-le comme suit :

```
# Begin /etc/inittab

id:2:initdefault:

si::sysinit:/etc/init.d/rcS

su:S:wait:/sbin/sulogin
```

```
l0:0:wait:/etc/init.d/rc 0
l1:1: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:6:respawn:/sbin/sulogin

ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now

1:2345:respawn:/sbin/agetty /dev/tty1 9600
2:2345:respawn:/sbin/agetty /dev/tty2 9600
3:2345:respawn:/sbin/agetty /dev/tty3 9600
4:2345:respawn:/sbin/agetty /dev/tty4 9600
5:2345:respawn:/sbin/agetty /dev/tty5 9600
6:2345:respawn:/sbin/agetty /dev/tty6 9600

# End /etc/inittab
```

7.5.6 Créer le fichier `/var/run/utmp`

Les programmes comme login, shutdown ont besoin d'écrire dans le fichier `/var/run/utmp`. Celui-ci contient des informations concernant les personnes actuellement connectées au système et la date du dernier arrêt du système.

- Créez le fichier `/var/run/utmp` en utilisant : `touch /var/run/utmp`
- Modifiez les permissions en lançant : `chmod 644 /var/run/utmp`

8 Créez les scripts de démarrage du système

Ces scripts sont lancés au démarrage du système et sont responsables de diverses tâches comme, par exemple, monter le système de fichiers en lecture/écriture, activer le *swap*, régler quelques paramètres systèmes et lancer divers démons dont le système a besoin.

8.1 Préparer les répertoires et les fichiers principaux

Dans cette section vous avez besoin du paquetage Sysvinit.

- Créez les répertoires nécessaires en tapant :

```
cd /etc
mkdir rc0.d rc1.d rc2.d rc3.d rc4.d rc5.d rc6.d init.d rcS.d
```

- Positionnez-vous dans le répertoire contenant l'archive Sysvinit décompressée
- Copiez le fichier `Debian/etc/init.d/rc` vers `/etc/init.d`
- Allez dans le répertoire `/etc/init.d`

– Créez un nouveau fichier rcS contenant :

```
#!/bin/sh
# Begin /etc/init.d/rcS

runlevel=S
prevlevel=N
umask 022
export runlevel prevlevel

trap ":" INT QUIT TSTP

for i in /etc/rcS.d/S??*
do
   [ ! -f  "$i" ] && continue;
   $i start
done

# End /etc/init.d/rcS
```

8.2 Créer le script de redémarrage `reboot`

– Créez un nouveau fichier `reboot` contenant ce qui suit :

```
#!/bin/sh
# Begin /etc/init.d/reboot

echo -n "System reboot in progress..."

/sbin/reboot -d -f -i

# End /etc/init.d/reboot
```

8.3 Créer le script d'arrêt `halt`

– Créez un nouveau fichier `halt` contenant ce qui suit :

```
#!/bin/sh
# Begin /etc/init.d/halt

/sbin/halt -d -f -i -p

# End /etc/init.d/halt
```

8.4 Créer le script `mountfs`

– Créez un nouveau fichier `mountfs` contenant :

```
#!/bin/sh
# Begin /etc/init.d/mountfs

check_status()
{
  if [ $? = 0 ]
  then
    echo "OK"
  else
    echo "FAILED"
  fi
}

echo -n "Remounting root file system in read-write mode..."
/bin/mount -n -o remount,rw /
check_status

> /etc/mtab
/bin/mount -f -o remount,rw /

echo -n "Mounting proc file system..."
/bin/mount proc
check_status

# End /etc/init.d/mountfs
```

8.5 Créer le script umountfs

– Créez un nouveau fichier umountfs contenant ce qui suit :

```
#!/bin/sh
# Begin /etc/init.d/umountfs

check_status()
{
  if [ $? = 0 ]
  then
    echo "OK"
  else
    echo "FAILED"
  fi
}

echo "Deactivating swap..."
/bin/swapoff -av
check_status

echo -n "Unmounting file systems..."
/bin/umount -a -r
check_status
```

```
# End /etc/init.d/umountfs
```

8.6 Créer le script sendsignals

– Créez un nouveau fichier sendsignals contenant ce qui suit :

```
#!/bin/sh
# Begin /etc/init.d/sendsignals

check_status()
{
  if [ $? = 0 ]
  then
    echo "OK"
  else
    echo "FAILED"
  fi
}
echo -n "Sending all processes the TERM signal..."
/sbin/killall5 -15
check_status

echo -n "Sending all processes the KILL signal..."
/sbin/killall5 -9
check_status
```

8.7 Créer le script de démarrage checkroot

– Créez un fichier /etc/init.d/checkroot contenant ce qui suit :

```
#!/bin/sh
# Begin /etc/init.d/checkroot

echo "Activating swap..."
/sbin/swapon -av

if [ -f /fastboot ]
then
  echo "Fast boot, no file system check"
else
  /bin/mount -n -o remount,ro /
  if [ $? = 0 ]
  then
    if [ -f /forcecheck ]
    then
      force="-f"
    else
      force=""
```

```
    fi

    echo "Checking root file system..."
    /sbin/fsck $force -a /

    if [ $? -gt 1 ]
    then
      echo
      echo "fsck failed. Please repair your file system manually by"
      echo "running fsck without the -a option"

      echo "Please note that the file system is currently mounted in"
      echo "read-only mode."
      echo
      echo "I will start sulogin now. CTRL+D will reboot your system."
      /sbin/sulogin
      /reboot -f
    fi
  else
    echo "Cannot check root file system because it is not mounted in"
    echo "read-only mode."
  fi
fi

# End /etc/init.d/checkroot
```

8.8 Créer le script de démarrage `sysklogd`

– Créez un nouveau fichier `/etc/init.d/sysklogd` contenant ce qui suit :

```
#!/bin/sh
# Begin /etc/init.d/sysklogd

check_status()
{
  if [ $? = 0 ]
  then
    echo "OK"
  else
    echo "FAILED"
  fi
}

case "$1" in
  start)
    echo -n "Starting system log daemon..."
    start-stop-daemon -S -q -o -x /usr/sbin/syslogd -- -m 0
    check_status

    echo -n "Starting kernel log daemon..."
```

```
    start-stop-daemon -S -q -o -x /usr/sbin/klogd
    check_status
    ;;

  stop)
    echo -n "Stopping kernel log daemon..."
    start-stop-daemon -K -q -o -p  /var/run/klogd.pid
    check_status

    echo -n "Stopping system log daemon..."
    start-stop-daemon -K -q -o -p /var/run/syslogd.pid
    check_status
    ;;

  reload)
    echo -n "Reloading system load daemon configuration file..."
    start-stop-daemon -K -q -o -s 1 -p /var/run/syslogd.pid
    check_status
    ;;

  restart)
    echo -n "Stopping kernel log daemon..."
    start-stop-daemon -K -q -o -p /var/run/klogd.pid
    check_status

    echo -n "Stopping system log daemon..."
    start-stop-daemon -K -q -o -p /var/run/syslogd.pid
    check_status

    sleep 1

    echo -n "Starting system log daemon..."
    start-stop-daemon -S -q -o -x /usr/sbin/syslogd -- -m 0
    check_status

    echo -n "Starting kernel log daemon..."
    start-stop-daemon -S -q -o -x /usr/sbin/klogd
    check_status
    ;;

  *)
    echo "Usage: $0 {start|stop|reload|restart}"
    exit 1
    ;;
esac

# End /etc/init.d/sysklogd
```

8.9 Créer les liens symboliques et fixer les permissions

– Paramétrez les permissions et liens symboliques comme suit :

```
chmod 755 rcS reboot halt mountfs umountfs sendsignals checkroot sysklogd
cd ../rc0.d
ln -s ../init.d/sysklogd K90sysklogd
ln -s ../init.d/sendsignals S80sendsignals
ln -s ../init.d/umountfs S90umountfs
ln -s ../init.d/halt S99halt
cd ../rc6.d
ln -s ../init.d/sysklogd K90sysklogd
ln -s ../init.d/sendsignals S80sendsignals
ln -s ../init.d/umountfs S90umountfs
ln -s ../init.d/reboot S99reboot
cd ../rcS.d
ln -s ../init.d/checkroot S05checkroot
ln -s ../init.d/mountfs S10mountfs
cd /etc/rc2.d
ln -s ../init.d/sysklogd S03sysklogd
```

8.10 Créer le fichier `/etc/fstab`

– Créez un fichier `/etc/fstab` contenant ce qui suit :

```
/dev/<LFS-partition designation> / ext2 defaults 0 1
/dev/<swap-partition designation> none swap sw 0 0
proc /proc proc defaults 0 0
```

9 Configurer la base du réseau

9.1 Installer Netkit-base

– Désarchivez Netkit-base et installez-le en tapant :

```
./configure
make; make install
cd etc.sample; cp services protocols /etc
mv /usr/bin/ping /bin
```

9.2 Installer Net-tools

– Désarchivez Net-tools et installez-le en tapant :

```
make; make install
mv /usr/bin/netstat /bin
cd /usr/sbin; mv ifconfig route /sbin
```

9.2.1 Créer le script de démarrage /etc/init.d/localnet

– Créez un nouveau fichier /etc/init.d/localnet contenant ce qui suit :

```
#!/bin/sh
# Begin /etc/init.d/localnet

check_status()
{
  if [ $? = 0 ]
  then
    echo "OK"
  else
    echo "FAILED"
  fi
}

echo -n "Setting up loopback device..."
/sbin/ifconfig lo 127.0.0.1
check_status

echo -n "Setting up hostname..."
/bin/hostname --file /etc/hostname
check_status

# End /etc/init.d/localnet
```

9.2.2 Paramétrer les permissions et les liens symboliques

– Fixez les permissions adéquates en lançant : `chmod 755 /etc/init.d/localnet`
– Créez les liens symboliques appropriés en lançant : `cd /etc/rcS.d ; ln -s ../init.d/localnet S03localnet`

9.2.3 Créer le fichier /etc/hostname

Créez un nouveau fichier /etc/hostname contenant le nom d'hôte de la machine. Il ne s'agit pas du FQDN (nom de domaine pleinement qualifié). Il ne s'agit que du nom que vous souhaitez donner à votre machine sur un réseau (yoda - gandalf - neo - etc).

9.2.4 Créez le fichier /etc/hosts

Si vous souhaitez configurer une carte réseau, vous devez choisir une adresse IP, un nom de domaine pleinement qualifié et les alias que vous mettrez dans le fichier /etc/hosts. Petit exemple :

```
<monip> myhost.mydomain.org aliases
```

Assurez-vous que l'adresse IP appartient à un réseau privé. Les plages d'adresses valides à utiliser sont :

```
Class Networks
A     10.0.0.0
B     172.16.0.0 through 172.31.0.0
C     192.168.0.0 through 192.168.255.0
```

`192.168.1.1` est une adresse valide. `me.linuxfromscratch.org` est un nom de domaine pleinement qualifié valide.

Si vous ne souhaitez pas utiliser de carte réseau, vous aurez simplement besoin d'un FQDN. Des programmes comme Sendmail en ont besoin pour travailler correctement. D'ailleurs Sendmail ne démarrera même pas s'il ne dispose pas d'un FQDN.

Voici le fichier `/etc/hosts` dont vous avez besoin si vous ne configurez pas de carte réseau :

```
# Begin /etc/hosts (no network card version)
127.0.0.1 me.lfs.org <contents of /etc/hostname> localhost
# End /etc/hosts (no network card version)
```

Voici le fichier `/etc/hosts` dont vous avez besoin si vous configurez une carte réseau :

```
# Begin /etc/hosts (version sans carte réseau)
127.0.0.1 localhost
192.168.1.1 me.lfs.org <contenu de /etc/hostname>
# End /etc/hosts (version sans carte réseau)
```

Evidemment, vous n'êtes pas obligé de choisir `192.168.1.1` et `me.lfs.org`. Choisissez ce que bon vous semblera.

9.2.5 Créer le fichier /etc/init.d/ethnet

Cette sous-section est utile uniquement si vous configurez une carte réseau. Si ce n'est pas le cas, continuez votre lecture au-delà.

Créez un nouveau fichier `/etc/init.d/ethnet` contenant ce qui suit :

```
#!/bin/sh
# Begin /etc/init.d/ethnet

check_status()
{
  if [ $? = 0 ]
  then
    echo "OK"
  else
    echo "FAILED"
  fi
}

/sbin/ifconfig eth0 <addresse_ip>
check_status
```

```
# End /etc/init.d/ethnet
```

9.2.6 Paramétrer les permissions et liens symboliques pour /etc/init.d/ethnet

- Fixez correctement les permissions en lançant `chmod 755 ethnet`
- Créez les liens symboliques appropriés en lançant `cd ../rc2.d ; ln -s ../init.d/ethnet S10ethnet`

9.2.7 Tester la configuration du réseau

- Démarrez le script `localnet` en lançant `/etc/init.d/localnet`
- Démarrez le script `ethnet`, si vous en avez créé un, en lançant `/etc/init.d/ethnet`
- Testez si `/etc/hosts` est correctement configuré en lançant :

```
ping <votre FQDN>
ping <votre nom d'hôte>
ping localhost
ping 127.0.0.1
ping 192.168.1.1 (uniquement si vous avez configuré votre carte réseau)
```

Les cinq commandes `ping` qui précédent doivent fonctionner sans problème. Si ce n'est pas le cas, la base du réseau n'est pas correctement configurée.

9.3 Tester le système

Maintenant que tous les logiciels sont installés, que les scripts de démarrage existent et que le réseau local est opérationnel, il est temps de redémarrer l'ordinateur afin de tester si ces nouveaux scripts fonctionnent correctement. Dans un premier temps, lorsque vous créez ces scripts, il est préférable de les exécuter manuellement depuis le répertoire `/etc/init.d` afin de corriger les erreurs les plus évidentes (erreurs de saisie notamment). Lorsque ces scripts seront opérationnels, il faudra s'assurer qu'ils se comportent également correctement au démarrage et à l'arrêt du système. Il n'existe qu'une seule façon de vérifier cela. Redémarrez votre système LFS grâce à la commande `shutdown -r now`. Au terme de ce redémarrage, vous arriverez à une invite normale, comparable à ce que vous avez sous votre système Linux normal (sauf si vous utilisez un gestionnaire d'affichage comme XDM, KDM etc).

A ce niveau, votre système LFS basique est opérationnel. Tout ce qui suit est optionnel, vous pouvez donc ignorer les chapitres que vous souhaitez. Mais gardez à l'esprit que si vous passez certaines sections comme l'installation de bibliothèques, certains programmes ne fonctionneront pas. Ainsi Lynx a besoin de la bibliothèque zlib. De plus zlib n'est pas uniquement utilisée par Lynx. A vous donc de vérifier ces problèmes de dépendance.

10 Installer les démons réseau

10.1 Configurer SMTP

10.1.1 Créer les groupes et utilisateurs

Créez les groupes dont Sendmail a besoin en tapant :

```
groupadd -g 1 bin
groupadd -g 2 kmem
groupadd -g 3 mail
useradd -u 1 -g bin -d /bin -s /bin/sh bin
```

10.1.2 Créer les répertoires

Le courrier électronique sortant est stocké dans le répertoire `/var/spool/mqueue` . Le courrier entrant est transmis à Procmail par Sendmail, puis stocké dans le répertoire `/var/mail`. Créez ces répertoires et fixez les permissions :

```
mkdir /var/spool/mqueue
mkdir /var/mail
cd /var/spool; ln -s ../mail mail
chmod 700 /var/spool/mqueue
chmod 775 /var/mail
chgrp mail /var/mail
chmod 1777 /tmp
```

10.1.3 Installer Sendmail

– Désarchivez Sendmail et installez-le en tapant :

```
cd src
./Build; ./Build install
```

10.1.4 Configuring Sendmail

Configurer Sendmail est plus facile à dire qu'à faire. Il y a un tas de choses à prendre en compte lorsque l'on configure cette application et je ne peux pas toutes les évoquer. C'est pourquoi pour le moment vous allez créer une configuration basique et standard. Si vous souhaitez peaufiner cette configuration, libre à vous, mais ce n'est pas le bon document pour apprendre comment faire. Vous pouvez éventuellement utiliser le fichier `/etc/sendmail.cf` (ou `/etc/mail/sendmail.cf`) existant.

– Positionnez-vous dans le répertoire `cf` des sources
– Créez un nouveau fichier `cf/lfs.mc` contenant ce qui suit :

```
OSTYPE(LFS)
FEATURE(nouucp)
define(`LOCAL_MAILER_PATH', /usr/bin/procmail)
MAILER(local)
MAILER(smtp)
```

– Créez un fichier `ostype/LFS.m4` vide en lançant `touch ostype/LFS.m4`
– Compilez le fichier `lfs.mc` en lançant `m4 m4/cf.m4 cf/lfs.mc > cf/lfs.cf`
– Copiez `cf/lfs.cf` vers `/etc/sendmail.cf`
– Créez un fichier `/etc/aliases` vide en lançant `touch /etc/aliases`
– Initialisez cette base de données vide en tapant : `sendmail -v -bi`

10.1.5 Installer Procmail

– Désarchivez Procmail et installez-le en tapant :

```
make; make install; make install-suid
```

10.1.6 Créer le script de démarrage /etc/init.d/sendmail

– Créez un nouveau fichier /etc/init.d/sendmail contenant ce qui suit :

```
#!/bin/sh
# Begin /etc/init.d/sendmail

check_status()
{
  if [ $? = 0 ]
  then
    echo "OK"
  else
    echo "FAILED"
  fi
}

case "$1" in
  start)
    echo -n "Starting Sendmail..."
    start-stop-daemon -S -q -o -x /usr/sbin/sendmail -- -bd
    check_status
    ;;

  stop)
    echo -n "Stopping Sendmail..."
    start-stop-daemon -K -q -o -p /var/run/sendmail.pid
    check_status
    ;;

  reload)
    echo -n "Reloading Sendmail configuration file..."
    start-stop-daemon -K -q -s 1 -p /var/run/sendmail.pid
    check_status
    ;;

  restart)
    echo -n "Stopping Sendmail..."
    start-stop-daemon -K -q -o -p /var/run/sendmail.pid
    check_status

    sleep 1

    echo -n "Starting Sendmail..."
```

```
    start-stop-daemon -S -q -o -x /usr/sbin/sendmail -- -bd
    check_status
    ;;

  *)
    echo "Usage: $0 {start|stop|reload|restart}"
    exit 1
    ;;

esac

# End /etc/init.d/sendmail
```

10.1.7 Paramétrer les permissions et liens symboliques

- Fixez les permissions appropriées en lançant `chmod 755 /etc/init.d/sendmail`
- Créez les liens symboliques adéquats en lançant :

```
cd /etc/init.d/rc2.d; ln -s ../init.d/sendmail S20sendmail
cd ../rc0.d; ln -s ../init.d/sendmail K20sendmail
cd ../rc6.d; ln -s ../init.d/sendmail K20sendmail
```

10.2 Configurer FTP

10.2.1 Créer les groupes et utilisateurs

- Créez les groupes nécessaires en lançant :

```
groupadd -g 65534 nogroup
groupadd -g 4 ftp
```

- Créez les utilisateurs nécessaires en lançant :

```
useradd -u 65534 -g nogroup -d /home nobody
useradd -u 4 -g ftp -s /bin/sh -m ftp
```

10.2.2 Installer Proftpd

- Désarchivez Proftpd et installez-le en tapant :

```
./configure
make; make install
```

10.2.3 Créer le script de démarrage `/etc/init.d/proftpd`

- Créez un nouveau fichier `/etc/init.d/proftpd` contenant ce qui suit :

```
#!/bin/sh
# Begin /etc/init.d/proftpd

check_status()
{
  if [ $? = 0 ]
  then
    echo "OK"
  else
    echo "FAILED"
  fi
}

case "$1" in
  start)
    echo -n "Starting Pro FTP daemon..."
    start-stop-daemon -S -q -o -x /usr/sbin/proftpd
    check_status
    ;;

  stop)
    echo -n "Stopping Pro FTP daemon..."
    start-stop-daemon -K -q -o -x /usr/sbin/proftpd
    check_status
    ;;

  restart)
    echo -n "Stopping Pro FTP daemon..."
    start-stop-daemon -K -q -o -x /usr/sbin/proftpd
    check_status

    sleep 1

    echo -n "Starting Pro FTP daemon..."
    start-stop-daemon -S -q -o -x /usr/sbin/proftpd
    check_status
    ;;

  *)
    echo "Usage: $0 {start|stop|restart}"
    ;;

esac

# End /etc/init.d/proftpd
```

10.2.4 Paramétrer les permissions et les liens symboliques

- Fixez les permissions appropriées en lançant : `chmod 755 /etc/init.d/proftpd`
- Créez les liens symboliques nécessaires en lançant :

```
cd /etc/rc2.d; ln -s ../init.d/proftpd S40proftpd
cd ../rc0.d; ln -s ../init.d/proftpd K40proftpd
cd ../rc6.d; ln -s ../init.d/proftpd K40proftpd
```

10.3 Configurer HTTP

10.3.1 Installer Apache

– Désarchivez Apache et installez-le en tapant :

```
./configure
make; make install
```

10.3.2 Configurer Apache

Il n'y a pas grande configuration à faire. La seule chose nécessaire est d'ajouter le chemin d'accès `/usr/apache/man` au fichier `/usr/share/misc/man.conf`

– Editez le fichier `/usr/share/misc/man.conf`
– Ajoutez cette ligne après les autres lignes commençant par MANPATH : *MANPATH /usr/apache/man*

10.3.3 Créer le script de démarrage `/etc/init.d/apache`

– Créez un nouveau fichier `/etc/init.d/apache` contenant ce qui suit :

```
#!/bin/sh
# Begin /etc/init.d/apache

case "$1" in
  start)
    echo -n "Starting Apache HTTP daemon..."
    /usr/apache/bin/apachectl start
    ;;

  stop)
    echo -n "Stopping Apache HTTP daemon..."
    /usr/apache/bin/apachectl stop
    ;;

  restart)
    echo -n "Restarting Apache HTTP daemon..."
    /usr/apache/bin/apachectl restart
    ;;

  force-restart)
    echo -n "Stopping Apache HTTP daemon..."
    /usr/apache/bin/apachectl stop

    sleep 1
```

```
    echo -n "Starting Apache HTTP daemon..."
    /usr/apache/bin/apachectl start
    ;;

  *)
    echo "Usage: $0 {start|stop|restart|force-restart}"
    ;;

esac

# End /etc/init.d/apache
```

10.3.4 Paramétrer les permissions et liens symboliques

- Fixez les permissions appropriées en lançant `chmod 755 /etc/init.d/apache`
- Créez les liens symboliques appropriés en lançant :

```
cd /etc/rc2.d; ln -s ../init.d/apache S50apache
cd ../rc0.d; ln -s ../init.d/apache K50apache
cd ../rc6.d; ln -s ../init.d/apache K50apache
```

10.4 Configurer Telnet

10.4.1 Installer le démon et le client telnet

- Désarchivez Netkit-telnet et installez-le en tapant :

```
./configure
make; make install
```

10.4.2 Créer le fichier de configuration `/etc/inetd.conf`

- Créez un nouveau fichier `/etc/inetd.conf` contenant ce qui suit :

```
# Begin /etc/inetd.conf

telnet stream tcp nowait root /usr/sbin/in.telnetd

# End /etc/inetd.conf
```

10.4.3 Créer le script de démarrage `/etc/init.d/inetd`

- Créez un nouveau fichier `/etc/init.d/inetd` contenant ce qui suit :

```
#!/bin/sh
# Begin /etc/init.d/inetd

check_status()
```

```
{
  if [ $? = 0 ]
  then
    echo "OK"
  else
    echo "FAILED"
  fi
}

case "$1" in
  start)
    echo -n "Starting Internet Server daemon..."
    start-stop-daemon -S -q -o -x /usr/sbin/inetd
    check_status
    ;;

  stop)
    echo -n "Stopping Internet Server daemon..."
    start-stop-daemon -K -q -o -p /var/run/inetd.pid
    check_status
    ;;

  reload)
    echo -n "Reloading Internet Server configuration file..."
    start-stop-daemon -K -q -s 1 -p /var/run/inetd.pid
    check_status
    ;;

  restart)
    echo -n "Stopping Internet Server daemon..."
    start-stop-daemon -K -q -o -p /var/run/inetd.pid
    check_status

    sleep 1

    echo -n "Starting Internet Server daemon..."
    start-stop-daemon -S -q -o -x /usr/sbin/inetd
    check_status
    ;;

  *)
    echo "Usage: $0 {start|stop|reload|restart}"
    ;;

esac

# End /etc/init.d/inetd
```

10.4.4 Paramétrer les permissions et liens symboliques

- Fixez les permissions appropriées en lançant : `chmod 755 /etc/init.d/inetd`
- Créez les liens symboliques appropriés en lançant :

```
cd /etc/rc2.d; ln -s ../init.d/inetd S30inetd
cd ../rc0.d; ln -s ../init.d/inetd K30inetd
cd ../rc6.d; ln -s ../init.d/inetd K30 inetd
```

10.5 Configurer PPP

10.5.1 Configurer le noyau

Avant de pouvoir vous connecter à Internet, vous devez configurer le support PPP dans le noyau. Pour ce faire, soit vous compilez le noyau avec support PPP intégré, soit vous compilez le support PPP en module qui sera chargé lorsque nécessaire. Quelle que soit la méthode que vous préférez, vous devez recompiler le noyau maintenant s'il ne supporte pas le protocole PPP.

10.5.2 Créer le groupe

- Créez le groupe du démon PPP en lançant : `groupadd -g7 daemon`

10.5.3 Installer PPP

- Désarchivez PPP et installez-le en tapant :

```
./configure
make; make install
```

10.5.4 Créer le fichier `/etc/resolv.conf`

- Créez un nouveau fichier `/etc/resolv.conf` contenant ce qui suit :

```
# Begin /etc/resolv.conf

nameserver <adresse IP du serveur DNS primaire de votre FAI>
nameserver <adresse IP du serveur DNS secondaire de votre FAI>

# End /etc/resolv.conf
```

10.5.5 Créer le fichier `/etc/ppp/peers/provider`

- Créez un répertoire `/etc/ppp/peers`
- Créez un nouveau fichier `/etc/ppp/peers/provider` contenant ce qui suit :

```
# Begin /etc/ppp/peers/provider

noauth
connect "/usr/sbin/chat -v -f /etc/chatscripts/provider"
```

```
/dev/ttyS1
115200
defaultroute
noipdefault

# End /etc/ppp/peers/provider
```

10.5.6 Créer le fichier /etc/chatscripts/provider

– Créez le répertoire `/etc/chatscripts`
– Créez un nouveau fichier `/etc/chatscripts/provider` contenant ce qui suit :

```
# Begin /etc/chatscripts/provider

ABORT BUSY
ABORT "NO CARRIER"
ABORT VOICE
ABORT "NO DIALTONE"
ABORT "NO ANSWER"
"" ATZ
OK ATDT <numéro de téléphone de votre FAI>
TIMEOUT 35
CONNECT ''
TIMEOUT 10
ogin: \ q<nom d'utilisateur FAI>
TIMEOUT 10
assword: \ q<mot de passe FAI>

# End /etc/chatscripts/provider
```

10.5.7 Remarque concernant l'authentification des mots de passe

Comme vous pouvez le voir dans les scripts ci-dessus (ce sont les scripts que j'utilise lorsque je ne travaille pas sous X), je me connecte chez mon FAI (Fournisseur d'Accès Internet) en utilisant ces scripts de communication plutôt qu'en utilisant l'authentification pap ou chap. Bien que mon FAI supporte pap, j'ai choisi de procéder de façon légèrement différente, avec les avantages et inconvénients que cette méthode comporte. Dans mon cas, les avantages sont plus nombreux que les inconvénients. En fait, en utilisant cette méthode, j'ai plus de contrôle sur la procédure de connexion et je peux voir ce qui se passe et quand ça se passe.

Par exemple, la plupart du temps, lorsque je me connecte, j'ouvre une fenêtre dans laquelle j'exécute

`tail -f /var/log/syslog` de manière à pouvoir voir quand (avec mon FAI ce serait plutôt 'si') les informations comme l'identifiant et le mot de passe sont envoyés.

11 Installer les clients réseau

11.1 Installer les clients Email

11.1.1 Installer Mailx

– Désarchivez Mailx et installez-le en tapant

```
make; make install
```

11.1.2 Installer Mutt

Mon client mail favori est Mutt, c'est donc celui-là que je vous conseille d'installer. Il ne s'agit pas d'une obligation, si vous avez un client mail préféré, installez-le. Après tout, c'est votre système, pas le mien. Notez que, si votre client est sous X Window, vous devrez attendre le chapitre concernant l'installation du système X Window pour pouvoir l'installer.

– Désarchivez Mutt et installez-le en tapant :

```
./configure
make; make install
```

11.1.3 Installer Fetchmail

– Désarchivez Fetchmail et installez-le en tapant :

```
./configure
make; make install
```

11.1.4 Tester le système email

Il est grand temps de tester notre système de courrier électronique.

- Lancez Sendmail en tapant `/usr/sbin/sendmail -bd` (vous devez indiquer le chemin d'accès à Sendmail complet. Si vous ne le faites pas, Sendmail n'aura pas la possibilité de recharger le fichier `sendmail.cf` lorsque vous utiliserez la commande `kill -1 <sendmail pid>`).
- Envoyez-vous un mail en lançant `echo "Ceci est un email d'essai" | mail -s test root`
- Démarrez le programme `mail`, vous devez voir apparaître votre mail.
- Créez un nouvel utilisateur en lançant `useradd -m testuser ; passwd testuser`
- Envoyez un mail à ce nouvel utilisateur en lançant `echo "mail d'essai vers testuser" | mail -s test testuser`
- Connectez-vous sur le compte de cet utilisateur et lancez le programme `mail` pour voir si l'envoi a fonctionné. Si c'est le cas, envoyez un mail à root de la même manière que vous avez envoyé un mail à testuser.

Si toutes ces étapes se sont terminées correctement, vous êtes parvenu à configurer un système mail local. Cela ne signifie pas qu'il est opérationnel pour Internet. Vous pouvez supprimer l'utilisateur de test en lançant `userdel -r testuser`

11.2 Installer le client FTP

11.2.1 Installer Netkit-ftp

– Désarchivez Netkit-ftp et installez-le en tapant :

```
./configure
make; make install
```

11.2.2 Tester le système FTP

– Lancez le démon Pro FTP en tapant : `/etc/init.d/proftpd start`
– Ouvrez une session FTP sur l'hôte local en tapant : `ftp localhost`
– Connectez-vous en tant qu'utilisateur anonyme et déconnectez-vous à nouveau.

11.3 Installer le client HTTP

11.3.1 Installer Zlib

Zlib est une bibliothèque de compression utilisée par des programmes comme zip et unzip de PKware. Lynx utilise cette bibliothèque pour compresser certains fichiers.

– Désarchivez Zlib et installez-le en tapant :

```
./configure --shared
make; make install
```

11.3.2 Installer Lynx

– Désarchivez Lynx et installez-le en tapant :

```
./configure --libdir=/etc --with-zlib
make; make install
make install-help; make install-doc
```

11.3.3 Tester le système HTTP

– Démarrez le démon Apache en tapant : `/etc/init.d/apache start`
– Démarrez une session http vers l'hôte local en tapant : `lynx http ://localhost`
– Quittez lynx.

11.4 Installer le client Telnet

Le client Telnet a déjà été installé lorsque nous avons mis en place le démon correspondant, au chapitre précédent.

11.4.1 Tester le système Telnet

– Démarrez le démon Serveur Internet (et avec lui telnetd) en tapant : `/etc/init.d/inetd start`
– Démarrez une session telnet vers l'hôte local en tapant : `telnet localhost`
– Connectez et déconnectez-vous.

11.5 Installer les clients PPP

11.5.1 Créer le script de connexion

– Créez un nouveau fichier `/usr/bin/pon` contenant ce qui suit :

```
#!/bin/sh
# Begin /usr/bin/pon

/usr/sbin/pppd call provider

# End /usr/bin/pon
```

11.5.2 Créer le script de déconnexion

– Créez un nouveau fichier `/usr/bin/poff` contenant ce qui suit :

```
#!/bin/sh
# Begin /usr/bin/poff

set -- `cat /var/run/ppp*.pid`

case $# in
  0)
    kill -15 `ps axw|grep "pppd call [[allnum:]]+"|grep -v grep|awk '{print $1}'`
    exit 0
    ;;
  1)
    kill -15 $1
    exit 0
    ;;
esac

# End /usr/bin/poff
```

11.5.3 Tester le système PPP

– Connectez-vous à Internet en tapant `pon`
– Essayez de vous connecter à un site comme `http ://www.linuxfromscratch.org`
– Déconnectez-vous d'Internet en lançant `poff`

12 Installer le système X Window

12.1 Installer X

– Désarchivez X et installez-le en tapant :

```
make World
make install; make install.man
```

Au cours de la compilation, vous verrez quelques erreurs indiquant que le script "`makedepend`" est incapable de trouver les fichiers `stddef.h`, `stdarg.h` et `float.h`. Manifestement ce script n'est pas aussi efficace que le compilateur car la compilation se déroule sans problème. Créer des liens symboliques ne résoudra pas le problème, au contraire cela en créera d'autres.

Vous pouvez donc superbement ignorer ces erreurs, ainsi que celles du type "pointer targets in passing arg x of somefunction differ in signedness". D'ailleurs vous pouvez réécrire ces fichiers si vous le souhaitez, pour ma part, je m'abstiendrai.

12.2 Créer `/etc/ld.so.conf`

Créez un nouveau fichier `/etc/ld.so.conf` contenant ce qui suit :

```
# Begin /etc/ld.so.conf

/lib
/usr/lib
/usr/X11R6/lib

# End /etc/ld.so.conf
```

– Mettez à jour le chargeur dynamique en tapant `ldconfig`

12.3 Créer le lien symbolique `/usr/include/X11`

– Afin que le préprocesseur trouve les fichiers X11/*.h, vous devez créer le lien symbolique suivant : `ln -s /usr/X11R6/include/X11 /usr/include/X11`

12.4 Créer le lien symbolique `/usr/X11`

Souvent, les applications copient des fichiers vers `/usr/X11`, sans avoir à se soucier de la version X utilisée. Ce lien symbolique n'a pas été créé à l'installation du système X Window, vous devez donc le faire.

– Créez le lien symbolique `/usr/X11` en tapant `ln -s /usr/X11R6 /usr/X11`

12.5 Ajouter `/usr/X11/bin` à la variable d'environnement $PATH

Il y a différentes méthodes pour ajouter le chemin `/usr/X11/bin` à la variable d'environnement $PATH . Voici une façon de procéder :

– Créez un nouveau fichier `/root/.bashrc` contenant : *export PATH=$PATH :/usr/X11/bin*

Vous devez vous déconnecter et vous reconnecter au système afin que cette modification soit prise en compte.

Une autre méthode consiste à mettre à jour la variable $PATH en tapant : `export PATH=$PATH :/usr/X11/bin` manuellement

12.6 Configurer X

– Configurez le serveur X en lançant `xf86config`

Si le fichier `XF86Config` créé par `xf86config` n'est pas apte à faire démarrer le serveur, il est alors préférable de copier le fichier `XF86Config` existant sur votre système Linux normal dans `/etc`. Dans certains cas, vous serez même obligé de modifier manuellement ce fichier de configuration lorsque `xf86config` sera incapable de gérer certains aspects de la configuration. Quoi qu'il en soit, modifier manuellement ce fichier est très long et le résultat obtenu n'est pas toujours celui attendu.

12.7 Tester X

Maintenant que X est configuré proprement, il est temps de faire un premier test.

– Lancez le serveur X en tapant `startx`

Le serveur X démarrera et affichera trois terminaux `xterm` à l'écran. Si c'est effectivement le cas, votre système X fonctionne correctement.

12.8 Installer Window Maker

J'ai choisi d'installer Window Maker car j'utilise ce gestionnaire de fenêtres depuis un certain temps et j'en suis très satisfait. Comme d'habitude, vous êtes libre d'installer votre gestionnaire de fenêtres favori, voire d'en installer plusieurs en fonction de votre humeur. Il suffira de spécifier dans le fichier `$HOME/.xinitrc` (ou `$HOME/.xsession` si vous utilisez `xdm`) lequel lancer.

12.9 Préparer le système pour l'installation de Window Maker

12.9.1 Installer libPropList

– Désarchivez libPropList et installez-le en tapant :

```
./configure
make; make install
```

12.9.2 Installer libXpm

– Désarchivez libXpm et installez-le en tapant :

```
xmkmf; make Makefiles; make includes; make depend
cd lib; make; make install
cd ..; make; make install
```

Cette installation légèrement différente est due à un *bug* dans un des fichiers `Makefile`. Il dépend des fichiers du répertoire `lib` qui ne sont pas encore installés mais ne les cherche pas dans le répertoire `lib`. Nous devons donc installer ces fichiers avant de compiler le paquetage.

12.9.3 Installer libpng

– Désarchivez libpng et installez-le en tapant :

```
make -f scripts/makefile.lnx; make -f scripts/makefile.lnx install
```

12.9.4 Installer libtiff

– Désarchivez libtiff et installez-le en tapant :

```
./configure
make; make install
```

12.9.5 Installer libjpeg

– Désarchivez libjpeg et installez-le en tapant :

```
./configure --enable-shared --enable-static
make; make install
```

12.9.6 Installer libungif

– Désarchivez libungif et installez-le en tapant :

```
./configure
make; make install
```

12.9.7 Installer WindowMaker

– Désarchivez WindowMaker et installez-le en tapant :

```
./configure
make; make install
```

12.10 Mettre à jour le cache du chargeur dynamique

– Mettez à jour le cache du chargeur dynamique en lançant : `ldconfig`

12.11 Configurer Window Maker

Tout utilisateur souhaitant utiliser WindowMaker doit d'abord lancer le script `wmaker.inst`. Ce dernier copiera les fichiers nécessaires dans le répertoire de l'utilisateur et modifiera le fichier `$HOME/.xinitrc` (ou le créera s'il n'existe pas).

– Configurer Window Maker en lançant `wmaker.inst`

12.12 Tester Window Maker

– Démarrez le serveur X et vérifiez que le gestionnaire de fenêtres Window Maker démarre convenablement. Pour ce faire, utilisez la commande `startx`

13 Ressources

Une liste de livres, HOWTOs et autres documents dont vous pourriez avoir besoin. Cette liste est loin d'être exhaustive mais contient le nécessaire pour démarrer. Nous espérons être en mesure de la compléter au fur et à mesure que nous trouverons de nouveaux documents utiles.

13.1 Livres

– *Sendmail* publié par O'Reilly. ISBN : 1-56592-222-0
– *Linux Network Administrator's Guide* publié par O'Reilly. ISBN : 1-56592-087-2
– *Running Linux* publié par O'Reilly. ISBN : 1-56592-151-8

13.2 HOWTOs

– ISP-Hookup-HOWTO à l'adresse *http ://www.linuxdoc.org*
– Linux Network Administrator's Guide en ligne à l'adresse : *http ://www.linuxdoc.org*

13.3 Autre

– Les diverses pages de manuel et pages info livrées avec les paquetages.

14 La Fin

Vous venez d'atteindre la fin du document Linux From Scratch HOWTO. J'espère que cette expérience vous a permis de compléter vos connaissances concernant le système Linux. Si vous souhaitez voir figurer ici certaines informations (corrections de bogues, applications supplémentaires oubliées que vous considérez comme importantes), faites-le nous savoir. Ensemble, grâce à votre aide et à vos suggestions, nous améliorerons ce document.

15 Copyright & information de licence