Mini-HOWTO : installation de Linux sur un lecteur ZIP sur port parallèle.

John Higgins, *jwiggins@comp.uark.edu*
Traduit par François Jeanmougin *pingouin@crystal.u-strasbg.fr*
La section consacrée à la debian à été traduite par cristofe le 13,4 floréal 2456.
v0.7, 26 Janvier 1998 (version Française : 01 Février 1998)

Ce document n'est destiné qu'à ceux qui disposent d'un lecteur ZIP sur port parallèle et qui veulent avoir un système Linux portable ou de sauvegarde.

Contents

1 Mise en garde

NOTE : Je n'ai aucune idée si le lecteur IDE fonctionne de la même manière que la version sur port parallèle, puisque je n'en ai pas, donc merci de ne pas me le demander.

La partie consacrée à la Debian a juste été recopiée avec pas ou peu de modification par l'auteur. En conséquence, il peut y avoir des redondaces dans les instructions.

Pré-requis :

- Vous avez un lecteur ZIP pour port parallèle (comme le ZIP Plus a les deux, je pense qu'il va fonctionner aussi bien).

- Vous avez déjà Linux installé et qui marche ; nous n'allons pas parler d'une première installation de Linux.
- Le pilote ppa du port parallèle est présent dans le noyau ou le module correspondant a été chargé.
- Le disque ZIP est monté sur le répertoire /iomega.

2 Introduction

Ce document est découpé en quatres parties qui décrivent chacune l'installation d'un système Linux minimal sur un disque ZIP de 100Mo utilisant le lecteur ZIP sur port parallèle. La première partie décrit comment préparer le disque ZIP, elle est commune aux distributions Slackware et Red Hat. Les deuxième, troisième et quatrièmes parties décrivent respectivement l'installation des distributions RedHat 4.2, Slackware 2.2 et Debian 1.2 sur le disque ZIP.

NOTE : J'ai bien vu que la Red Hat 5.0 était enfin sortie, mais entre les cours et le travail, bref, disons que je ne pourrai pas travailler dessus avant la fin mai. J'espère pouvoir aussi tester les autres distributions.

2.1 Nouveautés

J'ai finalement trouvé assez de temps (il faut dire que je préparais un concours de sociologie...) pour mettre à jour ce document. Merci à tous ceux qui m'ont envoyé des commentaires.

Nouveautés pour ce document

- La distribution Debian
- Lilo sur disquette
- Red Hat nouvelle version (4.2 Biltmore)
- Configuration réseau (non-testé pour la Red Hat)

2.2 Conventions

Le sigle ==> indique que le texte qui suit est une commande.
Ceci indique une remarque importante :

```
        NOTE:
```

Ceci indique une "capture d'écran"

```
        Texte visible à l'écran
```

2.3 Mises à jour

Pour les mises à jour, aussi rares qu'elles puissent être, consulter l'URL *http://comp.uark.edu/~jwiggins/linuxZIP/*

2.4 Autres lectures utiles

- Installation-HOWTO
- SCSI-HOWTO
- NET-3-HOWTO (pour sa section 4.3.3)
- ZIP-Drive (mini-HOWTO)
- la patch noyau ParPort (permet d'accéder au port imprimante) *http://www.cyberelk.demon.co.uk/parport/*

2.4.1 Remerciements, contributeurs

La section Slackware 2.2 m'a été confiée par Michael Littlejohn *mike@mesa7.mesa.colorado.edu*

La section Debian 1.2 m'a été confiée par John D. Blair *jdblair@uad.edu*

Les informations concernant LILO et beaucoup d'autres informations précises m'ont été confiées par Barcy Boese *possum@niagara.com* et Javier Rodriguez *jrodriguez@nextgeninter.net.mx*

3 Préparation du disque ZIP (commune aux distributions Slackware et Red Hat)

Avant de démarrer, assurez-vous d'avoir accès au lecteur ZIP; soit en ayant le pilote ppa du port parallèle dans le noyau ou en faisant charger le module correspondant. Un moyen facile de le vérifier est d'interroger dmesg :

==> **dmesg**

On peut rediriger la sortie vers less vu que dmesg a tendance à être un peu long. Voici un extrait du mien :

```
scsi0 : PPA driver version 0.26 using 4-bit mode on port 0x3bc.
scsi : 1 host.
  Vendor: IOMEGA    Model: ZIP 100          Rev: D.08
  Type:   Direct-Access                     ANSI SCSI revision: 02
Detected scsi removable disk sda at scsi0, channel 0, id 6, lun 0
SCSI device sda: hdwr sector= 512 bytes. Sectors= 196608 [96 MB] [0.1 GB]
 sda: Write Protect is off
 sda: sda1 sda2
```

Si vous voyez seulement quelque-chose du genre :

```
scsi : 0 hosts.
scsi : detected total.
```

Alors vous avez le support SCSI mais le lecteur ZIP n'a pas été détecté.

3.1 Partitionnement du disque ZIP

Pour partitionner le disque ZIP, lancer fdisk

==> **fdisk /dev/sda**

Voici un aperçu de la table de partition que j'ai faite :

```
Disk /dev/sda: 64 heads, 32 sectors, 96 cylinders
Units = cylinders of 2048 * 512 bytes

   Device Boot   Begin    Start      End   Blocks   Id  System
/dev/sda1            1        1       81    82928   83  Linux native
/dev/sda2           82       82       96    15360   82  Linux swap
```

J'ai décidé d'utiliser une partition de swap afin de pouvoir l'utiliser sur n'importe quelle machine (système portable).

3.2 Formater et monter le disque ZIP

Aprés avoir utilisé fdisk, formater la nouvelle partition :

==> **mke2fs -c /dev/sda1**

Ensuite, créer la partition de swap (15360 blocs comme le montre fdisk)

==> **mkswap -c 15360 /dev/sda2**

Finalement, monter le disque ZIP

==> **mount /dev/sda1 /iomega -t ext2**

3.3 Créer une disquette de démarrage (boot)

Étant donné que le lecteur ZIP sur port parallèle n'est pas un vrai périphérique SCSI, ce n'est pas un périphérique bootable, et donc, il nécessite une disquette de démarrage qui possède le pilote ppa dans le noyau et non sous forme de module.

3.3.1 Configurer et compiler le noyau

D'abord, il faut configurer et compiler un noyau qui a le pilote ppa du port parallèle actif et pas comme un module chargeable. Afin d'obtenir l'option port parallèle, sélectionner le support des périphériques SCSI :

SCSI support (CONFIG_SCSI) [Y/m/n/?]

Puis, le support des disques SCSI:

SCSI disk support (CONFIG_BLK_DEV_SD) [Y/m/n/?]

Et finalement, sous la rubrique des pilotes SCSI de bas niveau (SCSI low-level drivers), on trouve le pilote ppa pour port parallèle.

IOMEGA Parallel Port ZIP drive SCSI support (CONFIG_SCSI_PPA) [Y/m/n/?]

Encore une fois, il ne faut pas utiliser le pilote pour port parallèle en module, mais bien inclus dans le noyau. Jusqu'ici, sans le patch noyau PortPar (voir 1.4), le pilote du port parallèle ne permettra pas d'utiliser le port passif du lecteur ZIP pour une imprimante, donc il est alors préférable de désactiver le support pour imprimante sur port parallèle :

Parallel printer support (CONFIG_PRINTER) [N/y/m/?]

```
NOTE/ : Pour plus d'informations sur le pilote du port parallele,
          priere de se reporter au ZIP-Drive mini-HOWTO.<newline>
```

Une fois le noyau configuré, le compiler ainsi :

==> **make dep; make clean; make zImage**

Le nouveau noyau devrait se trouver dans arch/i386/boot/zImage.

3.3.2 Installer le noyau sur une disquette

Après avoir du me coltiner 4 disquetes différentes, en raison de différents noyaux nécessitant différents paramètres, (ainsi qu'après avoir lu le magnifique e-mail m'expliquant comment le faire), j'ai inclus une section sur LILO comme étant un des moyens de créer un disque de démarrage.

Installation via LILO Pour ceux qui ont besoin, ou pour ceux que ça intéresserait, d'avoir plusieurs noyaux sur une disquette (ces derniers temps, les miens sont trop gros) ou qui veulent juste pouvoir passer des arguments (comme le mode utilisateur unique (single)), j'ai reçu un email détaillant comment installer LILO sur une disquette.

Créer un système de fichier ext2 Pour créer un système de fichier ext2 sur une disquette, utiliser juste la même commande que pour e disque ZIP

==> **mke2fs -t /dev/fd0**

Copier les fichiers essentiels D'abord, s'assurer qu'il y ait un dossier cible pour le montage, et monter la disquette (j'utilise /dev/floppy) :

==> **mount /dev/fd0 /mnt/floppy -t ext2**

Pour démarrer correctement, vous avez besoin des mêmes fichiers que ceux que LILO utilise sur votre installation de Linux actuelle.

```
NOTE : Les chemins de fichiers donnés ici sont ceux
            de ma machine et peuvent ne pas être les mêmes pour tout
            le monde
```

==> **cp /boot/boot.b /mnt/floppy**

==> **cp /boot/map /mnt/floppy**

==> **cp /usr/src/linux/arch/i386/boot/zImage /dev/fd0 /mnt/floppy/vmlinuzDESK**

Maintenant, pour créer le fichier de configuration pour LILO, à ce point, j'ignore le programme liloconfig... (merci à Javier Rodriguez pour cette information). Premièrement, créer le fichier de configuration de LILO, /mnt/floppy/lilo.conf, pour le(s) noyau(x) pour le lecteur ZIP. Voici ce que j'ai utilisé et qui m'a permis d'avoir plusieurs noyaux à tester :

```
boot=/dev/fd0
map=/mnt/floppy/map
install=/mnt/floppy/boot.b
prompt
compact
timeout=50
image=/mnt/floppy/vmlinuzLAP
   label=Laptop
   root=/dev/sda1
   read-only
image=/mnt/floppy/vmlinuzDESK
   label=Desktop
```

```
      root=/dev/sda1
      read-only
    image=/mnt/floppy/vmlinuzDESK
      label=rescue
      root=/dev/hdc1
      read-only
```

J'ai ainsi deux noyaux. Un pour mon portable 486 qui a besoin de l'émulation du co-processeur mathématique dans le noyau, et l'autre pour mon ordinateur de bureau. Le "rescue" me permet de faire un démarrage dde secours sur le disque dur.

Enfin et surtout, lancer LILO pour l'installer sur la disquette grace à la commande :

==> **lilo -C /mnt/floppy/lilo.conf**

Une fois LILO installé sur la disquette, sauter les deux étapes suivantes, sauf si vous avez envie de tout recommencer :).

Installation du noyau seulement

```
    NOTE : Ceci n'a rien à voir avec l'installation
           via LILO.
```

Copier le nouveau noyau sur une disquette :

==> **cp arch/i386/boot/zImage /dev/fd0**

ou

==> **cat arch/i386/boot/zImage > /dev/fd0**

Bien sur, il existe pléthore de manières de copier un noyau sur une disquette, mais la dernière, ma favorite, est un peu plus cryptique. Essayez de ne pas oublier le > sauf si vous adorez voir des fichiers binaires.

Localiser la racine et le swap sur la disquette

```
    NOTE : Ceci n'a rien à voir avec l'installation
           via LILO.
```

Une fois que le noyau est sur la disquette, il faut définir le répertoire racine comme étant sur le disque ZIP :

==> **rdev /dev/fd0 /dev/sda1**

Je ne suis pas sur que l'option suivante soit nécessaire, quoiqu'il en soit, je le fais. Pour définir le swap :

==> **rdev -s /dev/fd0 /dev/sda2**

4 Installation de la RedHat 4.2 Dans l'univers de l'informatique, tout ce qui a plus de 3 mois est considéré comme obsolète et nécessite une mise à jour. Comme je n'aurai pas le temps de retoucher ce document à chaque mise à jour, je ferai de mon mieux pour le faire à chaque nouvelle version majeure. Comme pour les autres distributions, à moins que les auteurs désirent envoyer des corrections, celà restera en l'état.

4.1 Ma configuration

Pour mon installation, j'ai utilisé :

- noyau 2.0.30
- Lecteur ZIP Iomega sur port parallèle
- RedHat 4.2

4.2 Installation des paquetages

La première fois que j'ai tenté d'installer la Red Hat sur un disque ZIP, j'ai pensé qu'il aurait été plus simple de juste utiliser un disque de démarrage (boot disk) de la Red Hat. Je me suis alors réveillé. Je suis presque arrivé à créer un disque de démarrage, mais même avec l'aide par e-mail de plusieurs personnes de chez Red Hat, j'ai commencé à renoncer à tout ce projet lorsque j'ai découvert l'option –root du rpm.

4.2.1 Quels paquetages installer?

J'ai trouvé quels étaient les paquetages nécessaires en compulsant un fichier que j'ai trouvé sur un miroir RedHat. Ce fichier peut être trouvé sur n'importe quel miroir RedHat dans :

```
redhat/redhat4.2/i386/RedHat/base/comps
```

Pour cette installation, j'ai voulut inclure le support du réseau, mais, à cause de la La configuration du réseau sous X de la RedHat, j'ai du configurer à la main, ou plutot, tenter de configurer, les scripts qui se trouvent dans le /etc/sysconfig (cf section 4.3.3.). J'ai décidé de ne pas installer de paquetage de développement, car le lecteur ZIP, en tout cas avec un noyau non patché, est plutot lent pour compiler quoi que ce soit. J'ai aussi choisit de ne pas installer X, surtout pour des raisons d'espace disque disponible. Plus tard, j'essaierai de monter mon disque dur et de créer un lien symbolique avec /usr pour voir si j'arrive à faire fonctionner X.

Voici une liste des paquetages que j'ai installés, dans l'ordre de leur installation. Ceux marqués d'une * ont été mis à jour suit à des errata de RedHat. Entre parenthèse, j'ai indiqué le numéro du paquet de mise à jour.

```
Par exemple NetKit-B-0.09-6 a été mis à jour par NetKit-B-0.99-8, et
            apparait donc comme: *91) NetKit-B-0.99-6 (-8)
-----------------------------------------------------------------------
          (Liste créée par la commande rpm --root /iomega -qa)
```

```
  1) setup-1.7-2                      2) pamconfig-0.51-2
  3) filesystem-1.3-1                 4) MAKEDEV-2.2-9
  5) adduser-1.7-1                    6) libc-5.3.12-18
  7) SysVinit-2.64-8                  8) ash-0.2-8
  9) at-2.9b-2                       10) libtermcap-2.0.8-4
 11) bash-1.14.7-1                   12) bc-1.03-6
 13) bdflush-1.5-5                   14) cpio-2.4.2-4
 15) cracklib-dicts-2.5-1            16) tmpwatch-1.2-1
 17) crontabs-1.5-1                 *18) db-1.85-10 (-11)
 19) dev-2.5.1-1                     20) diffutils-2.7-5
 21) etcskel-1.3-1                   22) file-3.22-5
 23) fileutils-3.16-1                24) findutils-4.1-11
 25) grep-2.0-5                      26) groff-1.10-8
*27) ld.so-1.7.14-4 (-5)             28) getty_ps-2.0.7h-4
 29) gzip-1.2.4-7                    30) mingetty-0.9.4-3
*31) initscripts-2.92-1 (93-1)       32) ed-0.2-5
 33) info-3.9-1                      34) ncurses-1.9.9e-4
 35) libg++-2.7.1.4-5               *36) pwdb-0.54-3 (-4)
 37) rootfiles-1.5-1                *38) pam-0.57-2 (-4)
 39) redhat-release-4.2-1            40) less-321-3
 41) mount-2.5l-2                    42) zlib-1.0.4-1
 43) rpm-2.3.11-1                    44) e2fsprogs-1.10-0
 45) sysklogd-1.3-15                 46) tar-1.11.8-11
 47) passwd-0.50-7                   48) gawk-3.0.2-1
 49) gdbm-1.7.3-8                    50) gpm-1.10-8
 51) hdparm-3.1-2                    52) kbd-0.91-9
 53) slang-0.99.37-2                 54) newt-0.8-1
 55) kbdconfig-1.4-1                 56) ncompress-4.2.4-7
*57) sh-utils-1.16-4 (-5)            58) procinfo-0.9-1
*59) logrotate-2.3-3 (4-1)           60) lilo-0.19-1
 61) losetup-2.5l-2                  62) linuxthreads-0.5-1
*63) mkinitrd-1.6-1 (7-1)            64) mailcap-1.0-3
*65) man-1.4h-5 (j-1)                66) mt-st-0.4-2
 67) modules-2.0.0-5                 68) mailx-5.5.kw-6
 69) net-tools-1.32.alpha-2          70) procmail-3.10-10
 71) procps-1.01-11                  72) psmisc-11-4
 73) quota-1.55-4                    74) readline-2.0-10
 75) sed-2.05-6                      76) setconsole-1.0-1
 77) sendmail-8.8.5-4                78) shadow-utils-960530-6
 79) stat-1.5-5                      80) tcsh-6.06-10
 81) termcap-9.12.6-5                82) textutils-1.22-1
 83) time-1.7-1                      84) timeconfig-1.8-1
 85) util-linux-2.5-38               86) vim-4.5-2
 87) vixie-cron-3.0.1-14             88) which-1.0-5
 89) zoneinfo-96i-4                  90) tcp_wrappers-7.5-1
*91) NetKit-B-0.09-6 (-8)           *92) lpr-0.18-1 (19-1)
*93) bind-4.9.5p1-2 (9.6-1)         *94) bind-utils-4.9.5p1-2 (9.6-1)
*95) wu-ftpd-2.4.2b12-6 (b15-1)      96) anonftp-2.3-3
 97) zip-2.1-1                       98) unzip-5.12-5
 99) statserial-1.1-7               100) minicom-1.75-2
101) lrzsz-0.12.14-1                102) dip-3.3.7o-9
103) ppp-2.2.0f-3                   104) portmap-4.0-3
105) perl-5.003-8                  *106) traceroute-1.0.4.4bsd-2 (1.4a5-1)
*107) elm-2.4.25-7 (-8)             108) lynx-2.6-2
109) ncftp-2.3.0-5                  110) pine-3.95-2
```

```
 111) rdate-0.960923-1            112) apache-1.1.3-3
*113) nfs-server-2.2beta16-7     *114) nfs-server-clients-2.2beta16-7
      (2.2beta16-8)                    (2.2beta16-8)
```

Et avec tout ça, j'ai encore 32Mo de libre!

Mises à jour; errata Comme beaucoup d'utilisateurs de la RedHat (j'espère) le savent, certains paquetages peuvent présenter des trous de sécurité ou quelque chose d'autre qui pose problème. Pour cette raison, RedHat diffuse des mises à jour pour certains paquetages. J'ai installé ceux que j'ai signalé comme mis à jour ci-dessus. Veuillez vous référer à la page web RedHat qui concerne les mises à jour sur: *http://www.redhat.com/support/docs/rhl/rh42-errata-general.html*

```
NOTE: avant les mises à jour de paquetages, il faut
      lancer ldconfig comme décrit à la section 3.3.1.1.
```

4.2.2 Comment installer les paquetages en utilisant rpm; sans glint

Avec rpm, utiliser l'option –root pour spécifier le dossier monté comme la racine pour l'installation. J'ai découvert que plusieurs paquetages ne pouvaient pas être installés, en raison de scripts de préinstallation et/ou postinstallation qui ne s'exécutaient pas correctement à cause du répertoire racine différent. En conséquence, utiliser l'option –noscript :

==> **rpm –root /iomega -i –noscripts PAQUETAGE.i386.rpm**

Comme, j'en suis sur, beaucoup vont le voir, vous devriez avoir une erreur du style :

```
failed to open /iomega/var/lib/rpm/packages.rpm

error: cannot open /iomega/var/lib/rpm/packages.rpm
```

Bon, ben, il suffit de créer le répertoire /var/lib/rpm

==> **mkdir /iomega/var; mkdir /iomega/var/lib; mkdir /iomega/var/lib/rpm**

4.2.3 Lequel est arrivé en premier, pamconfig ou pam?

Si quelqu'un a déjà essayé d'installer pamconfig, celui-ci se plaint de dépendre de pam; alors, si l'on essaie d'installer pam, celui-ci se plaint d'avoir besoin de pamconfig! Ceci étant une nouvelle version de l'histoire de la poule et de l'oeuf, m'a laissé perplexe quelques temps, mais grâce à l'option –nodeps, on peut forcer l'installation de pamconfig; en outre, pam a plus de dépendances en échec que pamconfig.

==> **rpm –root /iomega -i –nodeps –noscripts pamconfig-0.51-2**

4.3 Problèmes après l'installation des paquetages

Une fois que le dernier paquetage a été parfaitement installé, malheureusement, le disque n'est pas entièrement fonctionnel, pour ainsi dire. Ce que je veux dire, c'est que, si vous essayez de démarrer maintenant avec la disquette, vous n'irez pas très loin. Dès qu'init essaie de démarrer, vous aurez deux adorables petites erreurs; chacune se plaignant de quelques fichiers qui auraient été là si les scripts avaient été lancés.

4.3.1 L'absence de ld.so.cache et libc.so.5

Si vous essayez de démarrer, vous avez donc deux messages d'erreurs. Le premier concerne l'absence du fichier /etc/ld/so.cache. Le second râle à propos de libc.so.5.

/etc/ld.so.cache Comme me l'ont signalé pas mal de lecteurs, mes précédentes instructions ne fonctionnaient pas comme prévu. Pour créer ce fichier, il faut lancer ldconfig lorsque le ZIP est encore monté.

==> **chroot /iomega /sbin/ldconfig**

Merci à Javier Rodriguez pour cette solution.

libc.so.5 Pour résoudre le cas de l'absence de cette librairie, il faut créer un lien symbolique qui aurait du être créé par les scripts d'installation.

==> **cd /iomega/lib; ln -s libc.so.5.3.12 libc.so.5**

Merci à Darcy Boese pour cette solution.

4.3.2 Configurer le mot de passe de root

De même qu'il a fallut lancer ldconfig en 4.3.1.1, il faudrait changer, créer le mot de passe de root pour ce nouveau système :

==> **chroot /iomega passwd root**

4.3.3 Ce que le programme d'installation a créé

```
NOTE: Ceci est just une configuration rapide, que je n'ai pas
été capable de tester pour voir si elle marche. En théorie, elle
devrait, mais ne venez pas pleurer si
elle ne fonctionne pas.
```

En parcourant mon CD Red Hat 4.2, je suis tombé sur quelque-chose de plutôt intéressant; le code source du programme d'installation. Je l'ai trouvé dans /misc/src/install et une chose que j'ai trouvé utile a été le fichier net.c. Là-dedans, j'ai trouvé quels autres fichiers devaient être créés au lancement du programme d'installation. La plupart participent au support du réseau (d'où le nom net.c) mais, même si vous n'avez pas de carte réseau, vous pouvez avoir à utiliser le localhost en réseau (en plus, apache va râler s'il ne peut pas déterminer de hostname). Ces fichiers comprennent :

```
/etc/hosts
/etc/HOSTNAME
/etc/resolv.conf
/etc/sysconfig/network
/etc/sysconfig/network-scripts/ifcfg-eth0
(ou tout autre périphérique réseau que vous auriez.)
```

4.3.4 Configuration du réseau

Pour rédiger ce document, il m'a fallut le support réseau pour ma carte fast-ethernet 3Com 3c595 (qui avait été utilisée ces derniers mois sur une autre machine).

D'abord, il me fallait un nom, et comme j'entretiens mon propre serveur de noms (ce qui est une autre et longue histoire), j'ai moi-même donné le nom : *dash-dot.wig.org* (J'aime bien comme ça sonne). Avoir un nom sans adresse IP est un peu vain, donc j'ai utilisé un numéro réseau réservé hors internet 192.168.10.0, que mon serveur de nom utilise aussi. Même si le nom de machine est logiquement indiqué dans /etc/HOSTNAME, Red Hat cherche ce nom dans /etc/sysconfig/network; alors allons-y : Extrait de mon /etc/sysconfig/network :

```
NETWORKING=yes
HOSTNAME=dash-dot
DOMAINNAME=wig.org
GATEWAY=
GATEWAYDEV=eth0
NS1=192.168.10.7
```

Ensuite, simplement, il faut recopier la même information, ici dans /etc/HOSTNAME, /etc/resolv.conf, et /etc/hosts:

/etc/HOSTNAME:

```
dash-dot.wig.org
```

/etc/resolv.conf:

```
search wig.org
nameserver ns.wig.org
```

/etc/hosts:

```
127.0.0.1 localhost
192.168.10.99 dash-dot.wig.org dash-dot
192.168.10.7  ns.wig.org ns
```

Red Hat configure tous les periphériques réseau à partir des scripts trouvés dans /etc/sysconfig/network-scripts. La configuration de chaque périphérique est habituellement créée une première fois par le programme d'installation, j'ai donc du créer ces fichiers de config à la main. Ils commencent tous par ifcfg-XXX où XXX est l'interface réseau qu'ifconfig va activer; par exemple ppp0, eth0, etc. Dans cet exemple, j'ai du créer un fichier ifcfg-eth0 qui contient ceci :;

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=none
BROADCAST=192.168.10.255
NETWORK=192.168.10.0
NETMASK=255.255.255.0
IPADDR=192.168.10.99
```

Et pour finir, afin que tout aille bien après ça, tant que l'on est dans /etc/sysconfig/network-scripts, lancer juste :

==> **./ifup ifcfg-eth0 boot**

Ceci va lancer le script qui configure l'interface réseau, si cette interface est configurée pour être lancée au démarrage.

Pour plus d'informations, prière de se reporter au NET-3-HOWTO.

5 Installation pour la Slackware 2.2

```
NOTE : Cette partie n'a pas été mise à jour.
```

5.1 Équipement nécessaire

- Disque et lecteur ZIP (évidement)
- Noyau avec support ZIP
- ZIP HowTo (recommandé)
- Une disquette formattée HD
- 1 à 2 heures de votre temps

5.2 Installation

Bon, d'abord, le plus jouissif : déterminer quels sont les fichiers qui sont nécessaires pour le disque système, et quels sont les paquetages que l'on veut (et que l'on peut) faire rentrer sur le disque ZIP.

J'ai pensé que le moyen le plus facile pour démarrer était d'installer directement la Slackware sur le lecteur ZIP. J'ai pensé à cette approche surtout parce que la Slackware est une distribution plus petite que la RedHat, et qu'il serait plus facile de supprimer ce que je ne voulais pas. Cela, et le fait que j'utilise de toute façon la distribution Slackware ont rendu ce choix logique.

Installer la Slackware sur un disque ZIP est facile, en tant que root, lancer le programme setup, et choisir /iomega comme partition d'installation. Configurer l'installation depuis la partition où se trouvent les sources de la Slackware (cdrom, disque dur, etc), sélectionner 'install' et suivre les instructions.

5.3 Qu'y mettre ?

La plus grosse difficulté est de décider ce que l'on doit ajouter et ce dont on peut se passer. Bien évidemment, on a besoin de la série 'A' (qui fait a peu prés 8 Mo), le reste dépend de chacun.

J'ai essayé de ramener la Slackware à une taille raisonnable de 70 Mo, et cela inclut gcc/g++, perl, X11R6 (MAIS PAS TOUT!), sendmail, de la doc (moins les pages de man des outils de développement, mais avec tous les howto), et un assortiment d'autres bonnes choses, ce qui laisse environ 10 Mo de libre pour les fichiers utilisateurs. YMMV (note du traducteur : hein???). C'est selon votre choix.

6 Créer le fichier /etc/fstab (commun aux deux distributions.)

La dernière chose que l'on doit faire avant de redémarrer est de créer le fichier fstab sur la partition qui va bientôt devenir root. Éditer le fichier /iomega/etc/fstab qui devra au minimum ressembler à ça:

```
/dev/sda1      /         ext2        defaults   1   1
/dev/sda2      none      swap        sw
none           /proc     proc        defaults   1   1
```

Sauver le fichier, et redémarrer à partir de la disquette fraîchement préparée, et amusez-vous bien!

(Un grand merci à Mike pour m'avoir rappelé ce point très important et même crucial. - John)

7 Installation avec Debian 1.2

```
NOTE : l'auteur de cette section me l'a envoyée le 11 juin 1997
```

7.1 Conditions requises

- disque et lecteur ZIP pour port parallèle.
- 2 disquettes 1,44 M vierges
- un jeu complet de disquettes d'installation Debian (voir la doc d'installation Debian si vous ne savez pas comment les faire)
- 2 ou 3 heures de libres

7.2 Vue d'ensemble

Après avoir passé quelques heures à me débattre avec `dpkg`, j'ai décidé qu'il serait plus simple de modifier le disque de secours (*rescue disk*) Debian pour qu'il reconnaisse le lecteur ZIP ppa. Cela s'est avéré être très facile. Vous pouvez utiliser ce disque modifié pour effectuer l'installation de base normale du système Debian. Une fois que vous avez installé le système de base vous pouvez utiliser une disquette de démarrage pour démarrer le nouveau système de base et compléter l'installation avec `dselect`. Pour utiliser cette technique, il faut construire deux noyaux - un avec le support ppa et disque mémoire initial, et un autre sans le support du disque mémoire.

Si vous voulez, vous pouvez sauter toutes les étapes de la section 2 et laisser la procédure d'installation de Debian formatter le disque ZIP pour vous.

7.3 Création du disque de secours modifié

Le disque de secours Debian est un disque de démarrage de type SYSLINUX, qui utilise un formattage DOS et un lanceur (*boot loader*) spécial pour empêcher le chargement de MS-DOS. Ce type de disque est très facile à modifier pour fabriquer vos propres disques de démarrage personnalisés. Le paquetage Debian “boot-floppies” contient un ensemble de scripts pour automatiser la création de disques de démarrage. Mais c'est tellement simple que j'ai préféré le faire à la main. C'est s'écarter un peu de la philosophie Debian, mais je suis au dessus de ça :). De brèves instructions se trouvent dans le fichier '`readme.txt`' de la disquette de secours.

7.3.1 Utiliser `dd` (ou `RAWRITE` sous DOS) pour créer la nouvelle disquette de secours

Consultez les docs d'installation de Debian si vous ne vous rappelez pas comment faire.

7.3.2 Construction d'un noyau avec le support ZIP ppa

Construisez un nouveau noyau avec le support ZIP ppa (comme à l'étape 3.3.1), mais sélectionnez également les supports “RAM disk” et “initial RAM disk”. Vous devez aussi sélectionner le support pour les systèmes de fichiers msdos, fat, minix, ext2fs et procfs.

Configurez également les modules que vous voulez avoir dans votre installation finale sur le disque ZIP.

Une fois la configuration du noyau effectuée, compilez le avec :

==> **make dep; make clean**

==> ***make bzImage***

Compilez les modules avec :

==> **make modules**

Vous les installerez plus tard.

```
NOTE : soyez bien sur d'utiliser 'make bzImage' et pas 'make zImage'.
```

7.3.3 Monter le disque de secours

==> **fdmount fd0**

ou

==> **mount /dev/fd0 /mnt**

ou

==> **que sais-je :)**

7.3.4 Copier l'image du noyau

Copiez l'image du noyau (sur les plateformes i386 c'est le fichier `arch/i386/boot/bzImage`) dans le fichier '`linux`' de la disquette.

7.3.5 Éditer le script '`rdev.sh`'

Ouvrez le script '`rdev.sh`' du disque de secours avec votre éditeur favori. Remplacez la dernière ligne :

```
rdev /mnt/linux /dev/ram0
```

par

```
rdev /mnt/linux /dev/sda1
```

Il faudra aussi remplacer toutes les occurrences de `/mnt/linux` par le chemin approprié. Comme je monte mes disquettes sous `/fd0`, j'ai dû changer `/mnt/linux` en `/fd0/linux`.

7.3.6 Lancer le script '`rdev.sh`' modifié

==> **./rdev.sh**

7.4 Installer le système de base sur le disque ZIP

Démarrez à partir du disque de secours. Si tout se passe bien vous retrouverez le processus d'installation à base de menu de la Debian, sauf qu'il sait maintenant que votre disque ZIP ppa existe. Effectuez l'installation normalement, comme s'il s'agissait d'un disque dur, mais montez `/dev/sda1` comme racine et choisissez `/dev/sda2` comme swap.

Il y a une différence par rapport au processus d'installation standard – n'installez pas et ne configurez pas de modules chargeables. Vous installerez les modules que vous avez compilés à l'étape 7.3.2 plus tard.

7.5 Création de la disquette de démarrage

Vous pouvez créer la disquette de démarrage comme décrit aux étapes 3.3.1 à 3.3.2, ou, si vous préférez, utiliser simplement l'option "Create Boot Disk" pendant l'installation Debian. J'aime bien la seconde possibilité car elle me permet d'avoir une autre disquette de démarrage SYSLINUX dans laquelle je peux modifier le message de bienvenue pour décrire la nature de ce disque de démarrage personnalisé et ajouter des paramètres noyau. Vous pouvez même inclure des fichiers d'aide accessibles depuis les touches de fonctions. Vous pouvez également recompiler le noyau et modifier à la main la disquette pour enlever le support RAM disk. À vous de voir.

7.6 Redémarrer le système

Insérez votre disquette de démarrage et choisissez l'option "Reboot" du menu d'installation.

7.7 Configuration du système de base et poursuite de l'installation

Quand le système redémarre vous avez une installation de base Debian tout à fait utilisable (mais lente) depuis votre disque ZIP ppa. Continuez l'installation normalement. j'ai installé tous les utilitaires UNIX classiques, avec les documentations, make, gcc, les bibliothèques (*libraries*), et divers utilitaires de manipulation de fichiers. Le résultat est un système de démarrage de secours très puissant que je peux utiliser pour "secourir" :-) n'importe lequel des systèmes de mon département qui aurait un gros problème.

```
NOTE : vous devrez au moins installer ``make'' pour réaliser l'étape suivante.
```

7.8 Installation des modules que vous avez construits en 7.3.2 (Construction d'un noyau avec le support ZIP ppa)

Si vous avez installé `make` à l'étape précédente, vous devriez pouvoir monter la partition du disque dur qui contient votre noyau, aller dans le répertoire approprié et lancer `make modules_install`. Par exemple :

==> **mount /dev/hda2 /mnt**

==> **cd /mnt/usr/src/linux**

==> **make modules_install**

```
NOTE : vous n'aurez pas à modifier le fichier /etc/fstab, comme expliqué à
l'étape 5. Le processus d'installation Debian s'en est déjà occupé.
```

8 Épilogue

Yahou, et snif. Après avoir entassé 31 disques ZIP et un pack de batteries pour ZIP autonome, mon aventure suivante est de faire fonctionner le pcmcia sur mon transportable (NdT : Il faut bien que je m'amuse aussi).