RedHat Linux KickStart HOWTO

Martin Hamilton <martinh@gnu.org>
Traduction: Laurent Martin <l_martin@worldnet.fr>
v0.1, 28 septembre 1998

Ce HOWTO décrit brièvement comment utiliser le système *KickStart* de RedHat pour rapidement installer un grand nombre de systèmes Linux identiques. Pour les utilisateurs expérimentés, on décrit comment modifier la procédure d'installation de KickStart pour l'adapter à ses propres besoins et donne quelques indications pour créer des paquetages RPM.

Contents

1 Copyright **2**

2 Page sur la Toile **2**

3 Introduction **2**

4 Prérequis **3**

5 Préparation d'une disquette d'amorçage **3**

6 Configuration de BOOTP/DHCP et NFS **4**

7 Le fichier de configuration de KickStart **5**

7.1 Information système 6

7.2 Paquetages à installer 8

7.3 Commandes shell après l'installation 10

8 L'installation **11**

9 Montage des disquettes d'amorçage et supplémentaire **11**

10 Modifier l'installateur RedHat **12**

11 Créer vos propres RPM **13**

12 FAQ/Liste de voeux **15**

13 Crédits **16**

14 Annexe **16**

1 Copyright

2 Page sur la Toile

Si vous avez obtenu ce document dans le répertoire HOWTO d'un site Linux ou sur un CD-ROM, vous pouvez vérifier la page *KickStart HOWTO* `http://wwwcache.ja.net/dev/kickstart/`

pour voir s'il n'y a pas une nouvelle version de disponible.

3 Introduction

La version 5 de Linux RedHat est livrée avec un utilitaire peu connu (et jusqu'à aujourd'hui quasiment pas documenté) appelé *KickStart*. Il vous permet d'automatiser (presque) toute l'installation d'une distribution Linux RedHat et notamment:

- la sélection de la langue;
- la configuration réseau et la sélection des sources de la distribution;
- la sélection du clavier;
- l'installation de l'utilitaire de démarrage (ex: lilo);
- le partitionnement du disque et la création du système de fichiers;
- la sélection de la souris;
- la configuration du serveur X-Window;
- la sélection de la zone géographique;
- la sélection du mot de passe de l'utilisateur *root*;
- la sélection des paquetages à installer.

Les utilisateurs perspicaces d'une distribution RedHat auront probablement réalisé qu'il s'agit des principales étapes de l'installation manuelle d'une distribution RedHat. KickStart vous permet d'automatiser le processus d'installation en plaçant les informations que vous rentreriez normalement au clavier dans un fichier de configuration.

Mais attendez, il y a mieux!

Une fois le processus d'installation achevé, KickStart vous permet de spécifier une liste de commandes shell que vous souhaitez voir exécutées. Cela signifie que vous pouvez automatiquement installer des logiciels locaux qui ne font pas partie de la distribution RedHat (et oui, il existe bien d'autres logiciels libres que ceux fournis avec la distribution RedHat! Certains ne peuvent, pour des raisons légales, être distribués par RedHat, par exemple: les systèmes de cryptage `ssh` et `PGP`) et procéder aux derniers réglages nécessaires pour rendre votre nouveau système d'exploitation parfaitement opérationnel.

4 Prérequis

Il y a deux manières d'utiliser KickStart. La première est de copier le fichier de configuration de KickStart sur une disquette d'amorçage RedHat. La seconde est d'utiliser une disquette d'amorçage classique et de récupérer ce fichier de configuration via le réseau.

Dans les deux cas, vous aurez besoin:

1. de machines utilisant un processeur Intel (i386) - KickStart ne semble marcher que sur ces machines au moment où j'écris ces lignes;
2. du fichier de configuration de KickStart - nous en reparlerons dans la prochaine section;
3. d'une disquette d'amorçage RedHat - de préférence celle du répertoire *updates*, pour profiter des dernières mises à jour et corrections des gestionnaires de périphérique;
4. des entrées DNS pour les adresses IP que vous allez utiliser - optionnel, mais cela évitera que le processus d'installation ne vous demande sans cesse le domaine de votre machine;

Si vous souhaitez récupérer le fichier de configuration via le réseau, vous aurez également besoin:

1. d'un serveur BOOTP/DHCP pour le réseau sur lequel seront installées vos machines. Certains serveurs alloueront automatiquement les nouvelles adresses dans une plage donnée (par exemple: *le serveur BOOTP CMU* `ftp://ftp.ntplx.net/pub/networking/bootp/` avec les extensions d'adressage dynamique).
2. d'un serveur NFS sur la même machine que le serveur BOOTP avec une copie de la distribution RedHat montée dessus et le fichier de configuration de KickStart dans un répertoire */kickstart* exporté par NFS.

Il doit être possible de se passer du serveur BOOTP - cela est implicite dans la documentation de KickStart. Mais je n'ai pas essayé moi-même. De même, il doit être possible de procéder à l'installation depuis un CD-ROM plutôt que depuis un serveur NFS. Si vous essayez l'une de ces deux possibilités, faites moi savoir comment vous avez procéder pour que je puisse l'indiquer dans ce document.

Notez qu'il n'est pas absolument indispensable que le serveur NFS contienne la distribution RedHat et le fichier de configuration KickStart, cela rend juste les choses un petit peu plus simples de tout avoir à un seul endroit.

5 Préparation d'une disquette d'amorçage

Tout ce que vous avez à faire est de copier le fichier de configuration de KickStart sur la disquette d'amoçage RedHat sous le nom *ks.cfg*:

```
mcopy ks.cfg a:
```

La disquette est particulièrement remplie, et vous pourriez avoir besoin de détruire certains fichiers pour faire de la place. J'ai réussi à mettre mon fichier de configuration sur la disquette en supprimant tous les fichiers de messages qu'affiche normalement `SYSLINUX`:

```
mdel a:\*.msg
```

Vous pouvez également éditer le fichier de configuration de SYSLINUX: *syslinux.cfg*. Il est situé à la racine de la disquette d'amorçage. Le fichier *syslinux.cfg* suivant permet de passer immédiatement en mode KickStart lorsque la machine démarre:

```
default ks
prompt 0
label ks
  kernel vmlinuz
  append ks=floppy initrd=initrd.img
```

6 Configuration de BOOTP/DHCP et NFS

Si vous vous demandez ce que peuvent bien être ces BOOTP et DHCP, une description détaillée est disponible sur le *site DHCP* `http://www.dhcp.org/` . NFS est documenté en détail dans le NFS HOWTO.

Dans la configuration NFS + BOOTP/DHCP que nous considérons, le fichier de configuration de KickStart doit être montable par NFS par la machine que l'on installe à partir de */kickstart/IPADDR-kickstart* sur le serveur BOOTP/DHCP, où *IPADDR* est l'adresse IP de la nouvelle machine. Par exemple */kickstart/198.168.254.254-kickstart* pour la machine *198.168.254.254*.

En théorie, il doit être possible de modifier cet emplacement en renvoyant le paramètre `bf` (*boot file*) dans la réponse du serveur BOOTP/DHCP. Il doit même être possible d'avoir ces fichiers montés par NFS à partir d'une autre machine.

Pour exporter par NFS certains répertoires à partir d'une machine Linux existante, créez le fichier */etc/exports* avec un contenu ressemblant à:

```
/kickstart *.swedish-chef.org(ro,no_root_squash)
/mnt/cdrom *.swedish-chef.org(ro,no_root_squash)
```

Notez que si vous n'avez pas enregistrer les adresses IP que vous allez utiliser dans le DNS, le serveur NFS ou le portmapper RPC risque de vous rejeter. Vous pouvez probablement vous en sortir en indiquant les paires adresse IP/masque réseau dans les fichiers de configuration:

```
/kickstart 198.168.254.0/255.255.255.0(ro,no_root_squash)
```

et dans */etc/hosts.allow*:

```
ALL: 194.82.103.0/255.255.255.0: ALLOW
```

Soyez conscient que si vous indiquez le mot de passe de *root* dans le fichier de configuration de KickStart ou exportez par NFS des répertoires contenant des informations sensibles, vous devrez prendre soin de rendre ces informations accessibles à aussi peu de personnes que possible. Cela peut être fait en restreignant les permissions des répertoires exportés, par exemple en indiquant un hôte ou un sous-réseau particulier plutôt qu'un domaine entier.

La plupart des serveurs NFS requièrent que vous indiquiez à `mountd` et `nfsd` (sur certaines versions d'Unix, ils sont précédés du préfixe `rpc.`) que le fichier */etc/exports* a été modifié - habituellement en envoyant un `SIGHUP`. Il existe souvent un programme ou un script appelé `exportfs` qui fera cela pour vous, par exemple:

```
# exportfs -a
```

Si NFS ne fonctionne pas lorsque votre machine démarre, les répertoires pourront ne pas être exportés automatiquement. Essayez de redémarrer la machine ou lancer les programmes suivants sous *root*:

```
# portmap
# rpc.nfsd
# rpc.mountd
```

Comme mentionné précédemment, sur certains système le préfixe `rpc.` n'est pas utilisé. Dans les distributions Unix les plus récentes, ces programmes se trouvent dans le répertoire */usr/sbin* qui peut ne pas encore être dans votre variable PATH. Le programme `portmap` est parfois appelé `rpcbind`, sous Solaris par exemple.

Si vous utilisez le serveur BOOTP CMU avec DHCP et les extensions d'adressage dynamique évoqué plus haut, une entrée du fichier */etc/bootptab* (*/etc/bootptab* est l'emplacement normal du fichier de configuration de BOOTP/DHCP) devrait ressembler à cela:

```
.dynamic-1:ip=198.168.254.128:T254=0x30:T250="ds=198.168.254.2:
dn=swedish-chef.org:sm=255.255.255.0:gw=198.168.254.1:
dl=0xFFFFFFFF":
```

(passages à la ligne pour plus de clarté)

Cette ligne indique que les nouvelles machines se verront affecter dynamiquement une adresse commençant à *198.168.254.128* et continuant pour les 48 adresses (la valeur héxadécimale *30*) suivantes. Chaque client recevra en retour la valeur de *T250*. Dans notre exemple, cela donne:

- le serveur DNS (`ds`) à *198.168.254.2*;
- le nom de domaine (`dn`) à *swedish-chef.org*;
- le masque du sous-réseau (`sm`) à *255.255.255.0*;
- la passerelle par défaut (`gw`) à *198.168.254.1*;
- la durée de vie (`dl`) (combien de temps l'adresse sera valide) à « pour toujours ».

Il semble qu'un grand nombre de versions de ce serveur ne gèrent pas l'adressage dynamique. Pour celles-ci, vous devrez énumérer les adresses physiques (typiquement MAC Ethernet) de chacune des machines à installer dans */etc/booptab*. Ces entrées devraient ressembler à quelque chose comme:

```
bork.swedish-chef.org:\
  ip=198.168.254.128:\
  ha=0000E8188E56:\
  ds=198.168.254.2:\
  dn=swedish-chef.org:\
  sm=255.255.255.0:\
  gw=198.168.254.1:\
  dl=0xFFFFFFFF":
```

Notez que le paramètre `ha` correspond à l'adresse physique de la machine à installer.

7 Le fichier de configuration de KickStart

Le fichier de configuration se compose de trois sections principales:

1. informations sur le système, partionnement des disques et configuration réseau;
2. paquetages RedHat à installer;
3. commandes shells à exécuter après l'installation.

Il existe d'autres possibilités que nous n'aborderons pas ici mais qui **pourraient** marcher. Pour de plus amples informations, regardez l'exmple de fichier de configuration de KickStart dans *misc/src/install/ks.samp* et le fichier *doc/README.ks* dans le répertoire *i386* d'une distribution RedHat sur votre CD-ROM ou sur un site miroir de RedHat.

7.1 Information système

Les commandes que j'ai utilisées sont:

lang

Configuration de la langue, pour l'anglais:

```
lang en
```

network

Configuration du réseau, pour utiliser BOOTP/DHCP:

```
network --bootp
```

nfs

serveur NFS et répertoire à partir duquel l'installation doit avoir lieu:

```
nfs --server chicken.swedish-chef.org /mnt/cdrom
```

pour utiliser le serveur NFS *chicken.swedish-chef.org* et essayer de monter la distribution RedHat à partir du répertoire */mnt/cdrom*.

keyboard

Sélection du type de clavier, pour un clavier anglais:

```
keyboard uk
```

zerombr

Efface le secteur d'amorçage du disque (MBR) - enlève tous les programmes de lancement pouvant s'y trouver.

clearpart

Efface les partitions existantes, pour supprimer toutes les partitions disque avant l'installation:

```
clearpart -all
```

part

Partionne le disque, pour créer un système de fichier de 500Mo:

```
part / --size 500
```

install

Effectue une nouvelle installation de RedHat.

mouse

Définit la souris utilisée, pour une souris PS/2 ou compatible:

```
mouse ps/2
```

timezone

Définit le fuseau horaire, pour l'heure anglaise:

```
timezone --utc Europe/London
```

rootpw

Définit le mot de passe initial de *root*, basé sur un mot de passe déjà crypté:

```
rootpw --iscrypted XaacoeGPmf/A.
```

lilo

Installe le programme LILO, pour l'installer dans le secteur d'amorçage du disque (MBR):

```
lilo --location mbr
```

%packages

Paquetages à installer - voir ci-après.

%post

Commandes shells à lancer après l'installation - voir ci-après.

Notez que le répertoire dans lequel KickStart va chercher la distribution RedHat doit contenir un sous-répertoire *RedHat* qui contient la distribution RedHat pour la plate-forme considérée. Dans notre exemple, nous devrions avoir quelque chose comme:

```
/mnt/cdrom/RedHat
/mnt/cdrom/RedHat/base
/mnt/cdrom/RedHat/contents
/mnt/cdrom/RedHat/i386
/mnt/cdrom/RedHat/instimage
/mnt/cdrom/RedHat/RPMS
/mnt/cdrom/RPM-PGP-KEY
```

Si vous souhaitez créer vos propres mots de passe cryptés, il vous suffit d'utiliser Perl:

```
% perl -e 'print crypt("schmurrdegurr", "Xa") . "\n";'p
```

Autres options que je n'ai pas testées:

cdrom

Installe à partir d'un CD-ROM plutô que du réseau.

device

Déclare explicitement les détails d'un périphérique, par exemple:

```
device ethernet 3c509 --opts "io=0x330, irq=7"
```

D'autres valeurs de `device` sont possibles dont `scsi` pour les contrôleurs SCSI et `cdrom` pour les gestionnaires de CD-ROM propriétaires.

upgrade

Met à jour une installation existante au lieu d'en installer une nouvelle.

xconfig

Configure le serveur X-Window, la carte graphique et le moniteur, par exemple:

```
xconfig --server "Mach64" --monitor "tatung cm14uhe"
```

Je n'ai pas beaucoup creusé cette dernière option car je ne prévois pas d'utiliser X sur les machines installées avec KickStart. Si vous le faites, tenez moi au courant.

Voici à quoi ressemble maintenant la première partie du fichier de configuration de KickStart:

```
lang en
network --bootp
nfs --server chicken.swedish-chef.org /mnt/cdrom
keyboard uk
zerombr yes
clearpart --all
part / --size 500
part swap --size 120
install
mouse ps/2
timezone --utc Europe/London
rootpw --iscrypted XaacoeGPmf/A.
lilo --location mbr
```

7.2 Paquetages à installer

La partie du fichier de configuration de KickStart consacrée aux paquetages débute par une ligne avec la directive `%packages`. Elle est suivie par l'un des deux types de spécifications de paquetage: des paquetages peuvent être installés individuellement en donnant le nom de leur RPM (sans la version ni la plate-forme), des groupes de paquetages peuvent être installés en donnant le nom de leur groupe.

Voici un exemple de la section des paquetages d'un fichier de configuration de KickStart:

```
%packages
@ Base
netkit-base
bind-utils
ncftp
rdate
tcp_wrappers
traceroute
cmu-snmp
```

Bien, à quoi correspondent ces groupes? Il y a un grand nombre de groupes définis par défaut dans un fichier nommé *base/comps* dans le répertoire racine de la distribution RedHat. Voici ceux que l'on pouvait y trouver au moment où j'écris ces lignes:

- Base;
- Gestionnaire d'imprimante;

- Système X-Window;
- Outils Mail/WWW/News;
- Connexion DOS/Windows;
- Gestionnaires de fichiers;
- Manipulation graphique;
- Jeux sous X-Window;
- Jeux en mode console;
- Gestionnaires multimédia sous X-Window;
- Console Multimedia;
- Serveur d'impression;
- Station en réseau;
- Station en *dial-up*;
- Serveur de news;
- Serveur NFS;
- Connection SMB (Samba) ;
- Connexion IPX/Netware(tm);
- Serveur FTP anonyme/Gopher;
- Serveur Web;
- Serveur de noms DNS;
- Serveur Postgres (SQL);
- Gestion de réseau;
- TeX;
- Emacs;
- Emacs avec X-Window;
- Développement en C;
- Bibliothèques de développement;
- Développement en C++ ;
- Development sous X-Window;
- Documentation supplémentaire.

Vous noterez qu'ils correspondent aux différentes configurations qui vous sont proposées lors de l'installation manuelle. Notez également que certains paquetages sont présents dans plusieurs groupes, sans que cela pose de problème lors de l'installation. L'entrée d'un groupe dans la liste *comps* ressemble à quelque chose comme:

```
0 Extra Documentation
sag
lpg
howto
faq
man-pages
end
```

Il semble que les groupes dont le nom est précédé d'un *1* fasse partie de l'installation par défaut. Il semble donc possible de pousser un peu plus loin la personnalisation du processus d'installation en créant ses propres groupes ou en redéfinissant les groupes existant. Gardez moi au courant si vous essayez de le faire.

7.3 Commandes shell après l'installation

C'est probablement la fonctionnalité la plus intéressante et en tous cas celle qui n'a pas d'équivalent direct dans le processus d'installation manuel. Ce que nous pouvons faire ici est de définir un ensemble de commandes de niveau shell qui seront exécutées une fois l'installation terminée (partitionnement du disque, installation des paquetages, etc.)

Cette section débute par la directive `%post` dans le fichier de configuration de KickStart. Vous pouvez ensuite utiliser tous les utilitaires qui viennent d'être installés sur votre nouvelle machine Linux, par exemple:

```
%post
ln -s /etc/rc.d/init.d /etc/init.d
ln -s /etc/rc.d/rc.local /etc/rc.local
ln -s /usr/bin/md5sum /usr/bin/md5
ln -s /usr/bin/perl /usr/local/bin/perl
chmod ug-s /bin/linuxconf
mkdir /var/tmp/tmp
perl -spi -e 's!image=/boot/vmlinuz-.*!image=/boot/vmlinuz!' /etc/lilo.conf
rm /etc/rc.d/rc*.d/*sendmail
```

Vous pouvez également rediriger les flux standards:

```
cat <<EOF >>/etc/passwd
squid:*:102:3500:Squid Proxy:/usr/squid:/bin/bash
EOF

cat <<EOF >>/etc/group
cache:x:3500:
EOF
```

Modifier les scripts de lancement:

```
cat <<EOF >>/etc/rc.local
echo 8192 > /proc/sys/kernel/file-max
echo 32768 > /proc/sys/kernel/inode-max

[ -x /usr/sbin/sshd ] && /usr/sbin/sshd
[ -x /usr/sbin/cfd ] && /usr/sbin/cfd

EOF
```

Définir les entrée de *crontab*:

```
cat <<EOF >/tmp/crontab.root
# Keep the time up to date
0,15,30,45 * * * * /usr/sbin/ntpdate -s eggtimer 2>&1 >/dev/null
# Recycle Exim log files
1 0 * * * /usr/exim/bin/exicyclog
# Flush the Exim queue
0,15,30,45 * * * * /usr/exim/bin/exim -q
EOF

crontab /tmp/crontab.root
rm /tmp/crontab.root
```

Et même installer d'autres RPM que vous avez créés:

```
rpm -i ftp://chicken.swedish-chef.org/rpms/squid.rpm
rpm -i ftp://chicken.swedish-chef.org/rpms/ssh.rpm
rpm -i ftp://chicken.swedish-chef.org/rpms/exim.rpm
rpm -i ftp://chicken.swedish-chef.org/rpms/cfengine.rpm
rpm -i ftp://chicken.swedish-chef.org/rpms/linux.rpm

ssh-keygen -b 1024 -f /etc/ssh_host_key -N ""
depmod -a
```

8 L'installation

Démarrer la machine à installer à partir de la disquette d'amorçage RedHat comme d'habitude, mais au lieu de presser `ENTRÉE` à l'invite du programme `SYSLINUX`, tapez `linux ks`.

Si vous avez de la chance, c'est tout ce que vous aurez à faire!

Si vous avez modifié la disquette d'amorçage comme mentionné plus haut, vous n'avez même pas besoin de vous préoccuper de ce qui suit :-)

Comme nous ne faisons qu'automatiser les différentes étapes du processus d'installation RedHat, une boîte de dialogue peut apparaître au cas où KickStart n'arrive pas à déterminer ce qu'il doit faire. Le cas le plus probable est qu'il ne détecte pas automatiquement votre carte réseau, il vous demandera alors de lui fournir son IRQ et son adresse mémoire d'entrée/sortie.

9 Montage des disquettes d'amorçage et supplémentaire

La disquette d'amorçage RedHat `boot.img` est au format MS-DOS et utilise le programme `SYSLINUX` pour se lancer. La disquette supplémentaire `supp.img` est au format Linux ext2. Si votre noyau intègre la gestion des périphériques *loopback*, vous pouvez monter ces deux fichiers dans votre système de fichiers:

```
# mkdir -p /mnt/boot /mnt/supp
# mount -o loop -t msdos boot.img /mnt/boot
# mount -o loop supp.img /mnt/supp
```

Vous devriez maintenant être capable de voir et de manipuler les fichiers des disquettes d'amorçage et supplémentaire respectivement sous */mnt/boot* et */mnt/supp*. Ouf! Notez que d'anciennes versions de `mount` peuvent ne pas être capables de gérer l'option `-o loop`. Dans ce cas, vous devrez utiliser `losetup` pour configurer le périphérique *loopback* pour chacun des fichiers:

```
# losetup /dev/loop0 boot.img
# mount -t msdos /dev/loop0 /mnt/boot
```

Vous aurez peut-être également besoin d'utiliser explicitement l'option `-t ext2` lorsque vous monterez la disquette supplémentaire. Cependant, les personnes ayant une distribution Linux récente ne devraient pas avoir à se soucier de cela.

Bien sûr, si vous ne voulez pas prendre de risque , vous pouvez utiliser les véritables disquettes plutôt que leurs images. Si votre temps est précieux, vous préférerez probablement utiliser les périphériques *loopback* car vous pourrez alors utiliser des images des disquettes plutôt que de supporter les temps d'attente liés à la lecture de véritables disquettes.

10 Modifier l'installateur RedHat

Si vous voulez modifier la procédure d'installation elle-même, son code source se trouve sur le CD-ROM RedHat ou sur le site miroir RedHat le plus proche dans le répertoire *misc/src/install* à partir du répertoire racine *i386*.

Si vous examinez la disquette d'amorçage RedHat, vous verrez qu'en plus du noyau *vmlinuz*, il y a un gros fichier *initrd.img*:

```
-rwxr-xr-x   1 root     root         559 May 11 15:48 boot.msg
-rwxr-xr-x   1 root     root         668 May 11 15:48 expert.msg
-rwxr-xr-x   1 root     root         986 May 11 15:48 general.msg
-rwxr-xr-x   1 root     root      968842 May 11 15:48 initrd.img
-rwxr-xr-x   1 root     root        1120 May 11 15:48 kickit.msg
-r-xr-xr-x   1 root     root        5352 May 11 15:48 ldlinux.sys
-rwxr-xr-x   1 root     root         875 May 11 15:48 param.msg
-rwxr-xr-x   1 root     root        1239 May 11 15:48 rescue.msg
-rwxr-xr-x   1 root     root         402 May 11 15:48 syslinux.cfg
-rwxr-xr-x   1 root     root      444602 May 11 15:48 vmlinuz
```

Vous l'aurez deviné, il s'agit d'un autre sytème de fichiers au format ext2 enregistré comme un fichier - mais avec un truc en plus. Il est compressé! Vous pouvez le décompresser et le monter:

```
# gzip -dc /mnt/boot/initrd.img >/tmp/initrd.ext2
# mkdir /mnt/initrd
# mount -o loop /tmp/initrd.ext2 /mnt/initrd
```

La partie probablement la plus importante de ce système de fichiers est sa collection de modules chargeables par le noyau qui sont sur la disquette d'amorçage. Si vous souhaitez intégrer la nouvelle version d'un gestionnaire, vous devrez soit remplacer *vmlinuz* par un nouveau noyau dans lequel ce gestionnaire sera lié statiquement, soit remplacer ce gestionnaire dans la collection de modules. Que dire d'autre sinon que vous pouvez supprimer certains modules pour faire de la place sur la disquette!

La collection de modules est le fichier *modules/modules.cgz*. Devinez-vous de quoi il s'agit? Et bien croyez le ou non, c'est une archive `cpio` compressée! Voici comment l'utiliser:

```
# gzip -dc /mnt/initrd/modules/modules.cgz >/tmp/modules.cpio
```

```
# cpio -itv <modules.cpio >modules.listing
# mkdir modules
# cpio -idumv <../modules.cpio
```

Je ne crois pas qu'il existe actuellement sous Linux une façon d'accéder de manière transparente aux systèmes de fichiers compressés (en tous cas avec les distributions les plus courantes). Faites le moi savoir si vous avez des informations là-dessus!

Si vous modifier quelque chose, rappelez vous:

1. utilisez `cpio` pour recréer l'archive. La façon de procéder est laissée en exercice au lecteur ...
2. utilisez `gzip` pour compresser cette archive;
3. copiez la dans */mnt/initrd*, ou dans tout autre endroit où vous avez placé l'archive *initrd.img* décompressée;
4. démontez */mnt/initrd* (ou comme vous l'avez appelé);
5. compressez le nouvel *initrd.img* avec `gzip`;
6. copiez l'archive sur la disquette d'amorçage: */mnt/boot/initrd.img* dans notre exemple;
7. démonter la disquette d'amorçage: */mnt/boot*.

Vous pouvez maintenant créer de nouvelles disquettes d'amorçage avec:

```
# cat boot.img >/dev/fd0
```

11 Créer vos propres RPM

Le format des paquetages RPM est déjà abondamment documenté, notamment dans le livre *Maximum RPM* d'Ed Bailey que vous pouvez télécharger depuis le *site RPM* `http://www.rpm.org/` ou trouver dans toutes les bonnes librairies! Cette section présente quelques trucs pour les gens pressés.

Les paquetages RPM sont construits à partir d'un fichier de spécification. Il consiste (de la même manière que le fichier de configuration de KickStart) d'un ensemble d'étapes à accomplir pour construire le paquetage - on suppose que vous avez à le construire à partir des sources, potentiellement pour plusieurs plates-formes, et avez besoin d'y appliquer des corrections avant la compilation. Une fois construit et installé, un fichier RPM sera créé à partir des fichiers et des répertoires que vous avez spécifiés comme étant associés au paquetage. Il est important de noter que RPM n'a aucune idée des fichiers et répertoires liés à un paquetage donné - vous devez le lui dire.

Voici un exemple de spécification pour une version personnalisée du

du serveur Cache WWW Squid `http://squid.nlanr.net/` :

```
Summary: Squid Web Cache server
Name: squid
Version: 1.NOVM.22
Release: 1
Copyright: GPL/Harvest
Group: Networking/Daemons
Source: squid-1.NOVM.22-src.tar.gz
Patch: retry-1.NOVM.20.patch
```

```
%description
Juste une première tentative d'empaquetage d'un serveur Squid pour
l'installer facilement sur notre serveur RedHat Linux

%prep
%setup
%build
configure --prefix=/usr/squid
perl -spi -e 's!#( -DALLOW_HOSTNAME_UNDERSCORES)!$1!' src/Makefile
make

%install
make install

%files
/usr/squid
```

Voici comment construire ce RPM:

```
% mkdir -p SOURCES BUILD SRPMS RPMS/i386
% cp ~/squid-1.NOVM.22-src.tar.gz SOURCES
% cp ~/retry-1.NOVM.20.patch SOURCES
% rpm -ba squid-1.NOVM.22+retry-1.spec
```

Cela va créer automatiquement un sous-répertoire dans le répertoire *BUILD* dans lequel il va déballer le code source et lui appliquer les corrections (de nombreuses options concernant les corrections sont disponibles, voir le livre pour plus de détails). RPM va maintenant automatiquement construire le paquetage en lançant `configure` suivi de `make`, l'installer avec `make install` et prendre une “photo” des fichiers situés dans */usr/squid*. C'est cette dernière qui va constituer le binaire RPM du logiciel Squid.

Notez que l'on peut insérer des commandes shell au cours des phases de décompression, construction et d'installation, par exemple des appels en `perl` pour modifier des paramètres de compilation.

Le fichier RPM final sera placé dans le répertoire *RPMS* dans le sous-répertoire de la plate-forme correspondante *i386*. Dans notre exemple, il s'appelera *squid-1.NOVM.22-1.i386.rpm*. Notez que le nom du fichier est créé en collant les valeurs de certains des paramètres du fichier de spécification: `Name`, `Version` et `Release` suivi de la plate-forme, *i386* dans ce cas. Gardez cela en mémoire lorsque vous créerez des RPM afin d'éviter de leur donner des noms exagérément longs.

Il est également intéressant de savoir que l'on peut contruire des RPM sans avoir à reconstruire tout le paquetage, par exemple:

```
Summary: Linux 2.0.35 kernel + filehandle patch + serial console patch
Name: linux
Version: 2.0.35+filehandle+serial_console
Release: 1
Copyright: GPL
Group: Base/Kernel
Source: linux-2.0.35+filehandle+serial_console.tar.gz
%description
C'est juste une première tentative de créer un paquetage du noyau
Linux avec ses corrections pour l'installation de notre serveur RedHat
Linux.

%prep
```

```
echo

%setup
echo

%build
echo

%install
echo

%post
/sbin/lilo

%files
/lib/modules/2.0.35
/boot/vmlinuz
```

Dans ce cas, nous créons simplement un RPM composé du fichier */boot/vmlinuz* et du contenu du répertoire */lib/modules/2.0.35*, et exécutons */sbin/lilo* après que le paquetage a été installé sur une nouvelle machine. Si vous connaissez une meilleure façon d'écrire le fichier de spécification, faites le moi savoir.

12 FAQ/Liste de voeux

Q: Peut-on appliquer automatiquement toutes les corrections de RPM? Et comment?

R1: Copiez les RPMs que vous voulez installer dans le répertoire RPMS à partir duquel l'installation aura lieu, supprimez les anciens RPMs et mettez à jour le fichier */RedHat/base/hdlist* avec le détail des nouveaux RPMs. Voir ci-dessous un script d'Eric Doutreleau qui fait cela pour vous. Si vous le faites vous-même, souvenez-vous de lancer *genhdlist* après!

R2: Essayer ce script Perl: *patchup* `http://wwwcache.ja.net/dev/patchup/` . Il compare les RPMs que votre système a installé avec ceux présents dans un répertoire donné et fournit une liste de ceux qu'il pense vous devriez mettre à jour. Il peut même les installer pour vous si vous lui faites confiance.

R3: *rpm2hml* est une version bien plus puissante (12Mo de C contre une page de Perl!) de R2.

Q: Un unique fichier de configuration sur le serveur d'installation pour tous les clients, peut-être une solution de repli après avoir essayé *IPADDR-kickstart*?

R: ?

Q:Plus de souplesse lorsque les chose vont mal - par exemple demander un chemin alternatif si la distribution ne se trouve pas sur le CD-ROM.

R: ?

Q: Exclusion explicite de paquetages - par exemple touts sauf *sendmail*.

R: ?

Q:Choisir quels services sont lancés automatiquement au démarrage par les scripts dans */etc/rc.d/*.

R: L'utilitaire *chkconfig* vous permet de configurer les services qui doivent être lancés au démarrage. Vous pouvez l'utiliser parmi les scripts lancés après l'installation par exemple pour lancer *ypbind* dans les *runlevel* 3, 4 et 5:

```
chkconfig --level 345 ypbind on
```

et il va lancer ypbind sur le niveau 345.

Q: Quand les commandes shell de la section `%post` s'exécutent, envoyer leur sortie vers une autre console plutô que d'écrire sur l'écran principal. *Cela peut-il être fait dans la section des commandes shell en utilisant `open`?*.

R: Pas de problème! Il suffit de faire quelque chose comme:

```
exec >/dev/tty5
```

Q: Le code de création de système de fichiers vérifie-t-il existence de blocs défectueux?

R: Si vous passez sur la console sur laquelle les sorties de la création du système de fichiers sont affichées, vous ne verrez aucune mention indiquant que le test 'read-only' a été effectué.

13 Crédits

Remerciements à Eric Doutreleau pour ses informations sur *chkconfig*, l'adaptation du fichier de configuration de `SYSLINUX` et pour son script Perl d'actualisation des RPM.

14 Annexe

Voici le script d'Eric pour ajouter les RPM actualisés dans les répertoires de la distribution RedHat:

```
#!/usr/bin/perl
#
$redhatdir="/cdrom/i386";
$rpmdir="/cdrom/i386/RedHat/RPMS/";
$updatedir="/cdrom/updates/";
@OTHERDIR=($updatedir);
foreach $dir (@OTHERDIR)
  {
    print "update for $dir\n";
    system(" find $dir -name \"*.rpm\" -exec cp {} $rpmdir \\; ");
  }
chdir($contribdir) || die "peux pas aller dans $contribdir $!\n";
system("chmod -R 755 $redhatdir");
chdir($rpmdir) || die "problem to go in $rpmdir $!\n";
#
# remove the old file
#
opendir(DIR,'.');
@package=grep(/\.rpm$/,readdir(DIR));
foreach $file (@package)
  {
    $file =~ /(.*)\-([\d+|\.]+\w*)\-(\d+)\.[i386|noarch].*/;
    $nom=$1;
    $version=$2;
    $buildvers=$3;
    if ($NOM{$nom})
      {
        $version2=$VERSION{$nom};
        $buildver2=$BUILDVERS{$nom};
```

```
        $file2=$FILE{$nom};
        $nom2=$NOM{$nom};
        if ( $version2 gt $version )
          {
            print "$file2 is newer than $file\n";
            unlink($file);
          }
        else
          {
            if ( $version2 lt $version )
              {
                print "$file is newer than $file2\n";
                unlink($file2);
                $VERSION{$nom}=$version;
                $BUILDVERS{$nom}=$buildvers;
                $FILE{$nom}=$file;
                $NOM{$nom}=$nom;
              }
            else
              {
                # print "$file2 $file same version version\n";
                if ( $buildver2 > $buildvers )
                  {
                    print "$file2 : $buildver2 est mieux que $file : $buildvers\n";
                    unlink($file);
                  }
                else
                  {
                    print "$file2 : $buildver2 is older than $file : $buildvers\n";
                    unlink($file2);
                    $VERSION{$nom}=$version;
                    $BUILDVERS{$nom}=$buildvers;
                    $FILE{$nom}=$file;
                    $NOM{$nom}=$nom;
                  }
              }
          }
      }
    else
      {
        $VERSION{$nom}=$version;
        $BUILDVERS{$nom}=$buildvers;
        $FILE{$nom}=$file;
        $NOM{$nom}=$nom;
      }
  }

# we do the hard thing here
#
system("$redhatdir/misc/src/install/genhdlist $redhatdir");
```