Recettes pour le HOWTO Root RAID

Michael A. Robinton, *michael@bzs.org* `mailto:michael@bzs.org` , traduit par Olivier Tharan, <tharan@int-evry.fr>
v1.06, 12 février 1998

Ce document fournit quelques recettes pour créer un système de fichiers RAID monté à la racine et le système de secours utilisant initrd qui l'accompagne. Il y a des instructions complètes, pas à pas, pour le périphérique md0 en RAID 1. Chaque étape est accompagnée de l'explication de son but. Cette procédure peut être utilisée pour toutes les autres structures RAID avec des modifications mineures.

Contents

1 Introduction

Nous supposerons que le lecteur est familier avec les divers types d'implémentation RAID, leurs avantages et inconvénients. Ceci n'est pas un guide, juste une série d'instructions sur la manière de mettre en place du RAID sur la partition montée à la racine sur un système Linux. Toutes les informations nécessaires à une familiarisation avec RAID sous Linux sont listées ici directement ou par référence ; veuillez les lire avant d'envoyer des questions par e-mail.

1.1 Où trouver une copie à jour de ce document.

Root-RAID-HOWTO

Disponible en LaTeX (pour le DVI et le PostScript), texte pur et HTML.

> *sunsite.unc.edu/mdw/HOWTO/* `http://sunsite.unc.edu/mdw/HOWTO/`

Disponible en SGML et HTML.

> *ftp.bizsystems.com/pub/raid/* `ftp://ftp.bizsystems.com/pub/raid/`

1.2 Bogues

Au moment d'écrire ce document, le problème qui consiste à arrêter un périphérique RAID monté à la racine n'a pas encore été résolu de manière satisfaisante. Une solution de contournement proposée par Ed Welbon et implémentée par Bohumil Chalupa est incluse dans ce document et élimine le besoin de faire un ckraid qui prend du temps à chaque démarrage pour des périphériques RAID 1 et RAID 5. Sans cette solution, il est nécessaire de faire un **ckraid** sur le périphérique **md** à chaque fois que le système est redémarré. Sur une grande matrice ceci peut entraîner une dégradation importante de la performance pour la disponibilité du système. Sur mon périphérique RAID 1 de 6 Go tournant sur un Pentium 166 avec 128 Mo de RAM, il faut au moins une demi-heure pour faire le ckraid :-(après chaque redémarrage.

La solution de contournement stocke l'état de la matrice au moment de l'extinction sur le périphérique de démarrage **réel** et le compare à un état de référence placé là à la première construction du système. Si les états concordent au redémarrage, le superblock de la matrice est reconstruit, sinon l'administrateur est averti de l'erreur d'état et le système de secours reste en fonctionnement avec tous les outils RAID disponibles.

La reconstruction du superblock permet au système d'ignorer que la matrice a été éteinte sans mdstop en marquant tous les disques d'un **OK**, comme si rien ne s'était passé. Ceci ne fonctionne que si tous les disques sont OK à l'extinction. Si la matrice fonctionnait avec un disque défectueux, l'administrateur devrait enlever le disque défectueux avant de redémarrer le périphérique md ou des données pourraient être corrompues.

Ceci ne s'applique pas pour RAID 0 qui n'a pas besoin d'un mdstop avant l'extinction.

Des solutions finales proposées pour ce problème comprennent un **finalrd** similaire à **initrd**, et un **mdrootstop** qui écrit les drapeaux **clean** (propre) sur la matrice pendant l'extinction quand la matrice est montée en lecture seule. Je suis sûr qu'il y en a d'autres.

Pendant ce temps, le problème a été contourné pour l'instant. Veuillez me tenir au courant quand ce problème sera résolu de manière plus propre !

1.3 Remerciements

Les écrits et courriers électroniques des personnes suivantes m'ont aidés à réaliser ce document. Beaucoup d'idées ont été *volées* du travail utile d'autres personnes, j'ai simplement essayé de les mettre sous forme de **RECETTES** pour qu'elles soient rapides à utiliser. Mes remerciements à :

- *Linas Vepstas* `mailto:linas@linas.org`

  pour le Howto RAID qui m'a expliqué la plupart des choses ici.

- *Gadi Oxman* `mailto:gadio@netvision.net.il`

  pour avoir répondu à mes questions idiotes de "bleu".

- *Ed Welbon* `mailto:welbon@bga.com`

  *pour l'excellent paquetage **initrd.md** qui m'a inspiré à écrire ceci.*

- *Bohumil Chalupa* `mailto:bochal@apollo.karlov.mff.cuni.cz` *pour avoir implémenté la "solution de contournement" du redémarrage qui permet à du* **RAID monté sur la racine** *de fonctionner dans un environnement de production.*

- et bien d'autres qui ont contribué à ce travail d'une façon ou d'une autre.

1.4 Note de copyright

2 Ce dont vous avez besoin AVANT DE COMMENCER

Les paquetages dont vous avez besoin et la documentation qui répond aux questions les plus fréquentes sur la manière d'installer et de faire fonctionner RAID sont listés ci-dessous. Veuillez les consulter en détail.

2.1 Paquetages nécessaires

Vous devez obtenir les versions les plus récentes de ces paquetages :

- un noyau Linux qui supporte RAID, initrd et /dev/loopx

    J'ai utilisé *linux-2.0.32* `ftp://sunsite.unc.edu/pub/Linux/kernel/`
    de sunsite

- *raid145-971022-2.0.31* `ftp://ftp.kernel.org/pub/linux/daemons/raid/`
  patch qui ajoute le support pour RAID 1/4/5
- *raidtools-pre3-0.42* `ftp://ftp.kernel.org/pub/linux/daemons/raid/`
  outils pour créer et maintenir des périphériques RAID (et aussi de la documentation).
- *linuxthreads-0.71* `ftp://ftp.inria.fr/INRIA/Projects/cristal/Xavier.Leroy`
  paquetage threads obligatoire. Utilisez FTP, les navigateurs ne fonctionnent pas.
- Une distribution Linux, prête à installer.

    J'ai utilisé *Slackware-3.4* `ftp://ftp.cdrom.com/pub/linux`
    disponible partout.

Les instructions détaillées de ce document sont basées sur les paquetages ci-dessus. Si les paquetages ont été mis à jour ou si vous utilisez une distribution Linux différente, vous devrez peut-être modifier les procédures que vous trouverez ici.

Les patches, assortiments d'outils, etc. peuvent varier avec les noyaux 2.1. Veuillez vérifier la documentation la plus récente à :

> *ftp.kernel.org/pub/linux/daemons/raid/* `ftp://ftp.kernel.org/pub/linux/daemons/raid/`

2.2 Autres implémentations similaires.

J'ai choisi d'inclure dans le noyau tous les morceaux nécessaires au fonctionnement dès le démarrage sans charger aucun module. L'image de mon noyau fait un peu plus de 300 Ko compressée.

Jetez un coup d'oeil à **initrd.md.tar.gz** d' *Ed Welbon* `mailto:welbon@bga.com` pour un autre moyen de fabriquer un périphérique RAID sur lequel on peut démarrer. Il utilise des modules chargeables. Un coup d'oeil à ses scripts concis vous montrera comment il s'y prend si vous avez besoin d'un noyau très petit avec des modules.

> *http://www.realtime.net/~welbon/initrd.md.tar.gz* `http://www.realtime.net/~welbon/initrd.md.tar.gz`

2.3 Documentation – lectures recommandées

Veuillez lire :

> **/usr/src/linux/Documentation/initrd.txt**

ainsi que la documentation et les pages de manuel qui accompagnent l'ensemble raidtools. En particulier, lisez **man mdadd** ainsi que le document **QuickStart.RAID** inclus dans le paquetage raidtools.

2.4 Ressources sur RAID

- *sunsite.unc.edu/mdw/HOWTO/mini/Software-RAID* `http://sunsite.unc.edu/mdw/HOWTO/mini/Software-RAID`
- *www.ssc.com/lg/issue17/raid.html* `http://www.ssc.com/lg/issue17/raid.html`
- *linas.org/linux/raid.html* `http://linas.org/linux/raid.html`
- *ftp.kernel.org/pub/linux/daemons/raid/* `ftp://ftp.kernel.org/pub/linux/daemons/raid/`
- *www.realtime.net/˜welbon/initrd.md.tar.gz* `http://www.realtime.net/~welbon/initrd.md.tar.gz`
- *luthien.nuclecu.unam.mx/˜miguel/raid/* `http://luthien.nuclecu.unam.mx/~miguel/raid/`

Vous pouvez joindre des listes de distribution à :

- *majordomo@nuclecu.unam.mx* `mailto:majordomo@nuclecu.unam.mx` *envoyez un message à* **subscribe raiddev**envoyez le courrier à : *raiddev@nuclecu.unam.mx* `mailto:raiddev@nuclecu.unam.mx`
- *majordomo@vger.rutgers.edu* `mailto:majordomo@vger.rutgers.edu` *envoyez un message à* **subscribe linux-raid**envoyez le courrier à : *linux-raid@vger.rutgers.edu* `mailto:linux-raid@vger.rutgers.edu`

  (ceci semble être la liste la plus active)

3 *initrd* : recette pour du RAID monté sur la racine

Voici la procédure pour créer un disque RAM (ramdisk) 'initrd' avec des outils de secours pour le RAID.

Spécifiquement, ce document se rapporte à une implémentation RAID 1, cependant il est applicable de manière générale à n'importe quel arrangement RAID avec un périphérique RAID monté à la racine.

3.1 Rappel de sécurité

Le système de fichiers de secours peut être utilisé seul. S'il arrivait que votre matrice RAID ne puisse être montée, vous êtes réduit à utiliser le système de secours monté et en fonctionnement. PRENEZ LES PRÉCAUTIONS DE SÉCURITÉ APPROPRIÉES !

3.2 Construction du noyau et des outils RAID

La première chose à faire est de patcher et de construire votre noyau et de devenir familier avec les outils RAID. Configurez, montez et testez votre (vos) périphérique(s) RAID. Les détails sur la manière de procéder sont inclus dans le paquetage **raidtools** et revus en détails plus loin dans ce document.

3.3 Construction du système de fichier *initrd* de secours et de démarrage

J'ai utilisé la distribution **Slackware-3.4** pour construire à la fois le système de fichiers de secours/démarrage et le système de fichiers de la machine de production. N'importe quelle distribution Linux devrait fonctionner correctement. Si vous utilisez une distribution différente, relisez la partie de cette procédure spécifique à Slackware et modifiez-la pour l'adapter à vos besoins.

Vous pouvez charger la distribution Slackware à partir de :

ftp.cdrom.com/pub/linux/ `ftp://ftp.cdrom.com/pub/linux/`

Si vous disposez déjà de Slackware, vous n'avez besoin de charger que les nouveaux ensembles 'a', 'ap' et 'n'.

J'utilise Loadlin pour lancer l'image du noyau et un disque xRAM (ramdisk) à partir d'une partition DOS. J'ai choisi de créer un système de disque RAM minimal en utilisant le script d'installation 'setup' de Slackware suivi de l'installation des paquetages 'linuxthreads' et 'raidtools' sur l'installation propre de Slackware sur mon disque en RAM. J'ai utilisé une procédure ***identique*** pour construire le système de production. Par conséquent les systèmes de secours et de production sont très similaires.

Ce processus d'installation me donne un système 'nu' (sauvez une copie du fichier) sur lequel j'ajoute

```
/lib/modules/2.x.x......
/etc .... avec un fstab modifie
/etc/rc.d
/dev/md*
```

à partir de mon système en cours pour le personnaliser pour le noyau et la machine sur lesquels il tourne ou tournera.

Ceci rend le système de démarrage et de secours identique au système qui fonctionnera sur le périphérique RAID monté à la racine, simplement épuré un peu, tout en permettant de toujours garder à jour les versions de bibliothèques et autres.

3.4 Début des instructions pas à pas

À partir du répertoire de démarrage de root (/root) :

```
cd /root
mkdir raidboot
cd raidboot
```

Créez des points de montage sur lesquels travailler

```
mkdir mnt
mkdir mnt2
```

Créez un fichier suffisamment grand pour effectuer l'installation du système de fichiers. Il sera beaucoup plus grand que le système de fichiers de secours final. J'ai choisi 24 Mo car 16 Mo ne suffisent pas.

```
dd if=/dev/zero of=build bs=1024k count=24
```

Associez le fichier à un périphérique de boucle (loopback) et créez un système de fichiers ext2 sur le fichier :

```
losetup /dev/loop0 build
mke2fs -v -m0 -L initrd /dev/loop0
mount /dev/loop0 mnt
```

3.5 Installation de la distribution - spécifique à Slackware

Sautez la partie spécifique à la Slackware et allez à la section 3.6 (suivante).

Maintenant qu'un système de fichiers vide est créé et monté, lancez "setup".

```
Specifiez              /root/raidboot/mnt
```

pour la cible (**'target'**). La source est celle à partir de laquelle vous faites l'installation habituellement. Sélectionnez les paquetages que vous désirez installer et continuez mais **NE LANCEZ PAS** configure.

Choisissez le mode 'EXPERT'.

J'ai choisi les ensembles 'A', 'AP' et 'N' en n'installant que le minimum pour faire tourner le système plus un éditeur auquel je suis habitué (vi, jed, joe) qui soit suffisamment petit.

```
lqqqqqqqq SELECTING PACKAGES FROM SERIES A (BASE LINUX SYSTEM) qqqqqqqqk
x lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
x x   [X] aaa_base  Basic filesystem, shell, and utils - REQUIRED    x x
x x   [X] bash      GNU bash-1.14.7 shell - REQUIRED                 x x
x x   [X] devs      Device files found in /dev - REQUIRED            x x
x x   [X] etc       System config files & utilities - REQUIRED       x x
x x   [X] shadow    Shadow password suite - REQUIRED                 x x
x x   [ ] ide       Linux 2.0.30 no SCSI (YOU NEED 1 KERNEL)         x x
x x   [ ] scsi      Linux 2.0.30 with SCSI (YOU NEED 1 KERNEL)       x x
x x   [ ] modules   Modular Linux device drivers                     x x
x x   [ ] scsimods  Loadable SCSI device drivers                     x x
x x   [X] hdsetup   Slackware setup scripts - REQUIRED               x x
x x   [ ] lilo      Boots Linux (not UMSDOS), DOS, OS/2, etc.        x x
x x   [ ] bsdlpr    BSD lpr - printer spooling system                x x
x x   [ ] loadlin   Boots Linux (UMSDOS too!) from MS-DOS            x x
x x   [ ] pnp       Plug'n'Play configuration tool                   x x
x x   [ ] umsprogs  Utilities needed to use the UMSDOS filesystem    x x
x x   [X] sysvinit  System V-like INIT programs - REQUIRED           x x
x x   [X] bin       GNU fileutils 3.12, elvis, etc. - REQUIRED       x x
x x   [X] ldso      Dynamic linker/loader - REQUIRED                 x x
x x   [ ] ibcs2     Runs SCO/SysVr4 binaries                         x x
x x   [X] less      A text pager utility - REQUIRED                  x x
x x   [ ] pcmcia    PCMCIA card services support                     x x
x x   [ ] getty     Getty_ps 2.0.7e - OPTIONAL                       x x
x x   [X] gzip      The GNU zip compression - REQUIRED               x x
x x   [X] ps        Displays process info - REQUIRED                 x x
x x   [X] aoutlibs  a.out shared libs - RECOMMENDED                  x x
x x   [X] elflibs   The ELF shared C libraries - REQUIRED            x x
x x   [X] util      Util-linux utilities - REQUIRED                  x x
x x   [ ] minicom   Serial transfer and modem comm package           x x
x x   [ ] cpio      The GNU cpio backup/archiving utility            x x
x x   [X] e2fsbn    Utilities for the ext2 file system               x x
x x   [X] find      GNU findutils 4.1                                x x
x x   [X] grep      GNU grep 2.0                                     x x
x x   [ ] kbd       Change keyboard mappings                         x x
x x   [X] gpm       Cut and paste text with your mouse               x x
x x   [X] sh_utils  GNU sh-utils 1.16 - REQUIRED                     x x
```

```
x x   [X] sysklogd  Logs system and kernel messages                  x x
x x   [X] tar       GNU tar 1.12 - REQUIRED                          x x
x x   [ ] tcsh      Extended C shell version 6.07                    x x
x x   [X] txtutils  GNU textutils-1.22 - REQUIRED                    x x
x x   [ ] zoneinfo  Configures your time zone                        x x
x mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
```

Dans l'ensemble 'AP', je n'utilise que 'JOE' et l'éditeur que je préfère, et 'MC', petit outil utile de gestion de fichiers. Choisissez les outils dont vous aurez besoin sur votre système.

```
lqqqqqqqqq SELECTING PACKAGES FROM SERIES AP (APPLICATIONS) qqqqqqqqqk
x x     [ ] ispell    The International version of ispell         x x
x x     [ ] jove      Jonathan's Own Version of Emacs text editor  x x
x x     [ ] manpgs    More man pages (online documentation)        x x
x x     [ ] diff      GNU diffutils                                x x
x x     [ ] sudo      Allow special users limited root access      x x
x x     [ ] ghostscr  GNU Ghostscript version 3.33                 x x
x x     [ ] gsfonts1  Ghostscript fonts (part one)                 x x
x x     [ ] gsfonts2  Ghostscript fonts (part two)                 x x
x x     [ ] gsfonts3  Ghostscript fonts (part three)               x x
x x     [ ] jed       JED programmer's editor                      x x
x x     [X] joe       joe text editor, version 2.8                 x x
x x     [ ] jpeg      JPEG image compression utilities             x x
x x     [ ] bc        GNU bc - arbitrary precision math language   x x
x x     [ ] workbone  a text-based audio CD player                 x x
x x     [X] mc        The Midnight Commander file manager          x x
x x     [ ] mt_st     mt ported from BSD - controls tape drive     x x
x x     [ ] groff     GNU troff document formatting system         x x
x x     [ ] quota     User disk quota utilities                    x x
x x     [ ] sc        The 'sc' spreadsheet                         x x
x x     [ ] texinfo   GNU texinfo documentation system             x x
x x     [ ] vim       Improved vi clone                            x x
x x     [ ] ash       A small /bin/sh type shell - 62K             x x
x x     [ ] zsh       Zsh - a custom *nix shell                    x x
x mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
```

Dans l'ensemble 'N' je n'ai pris que TCP/IP. Ceci n'est pas vraiment nécessaire, mais est très utile et permet d'avoir accès au réseau pendant un travail de réparation ou de mise à jour avec la matrice RAID à la racine démontée. TCP/IP contient aussi 'biff' qui est utilisé par certaines applications de 'A'. Si vous n'installez pas 'N' vous voudrez quand même installer le paquetage biff.

```
lqqqqq SELECTING PACKAGES FROM SERIES N (NETWORK/NEWS/MAIL/UUCP) qqqqqk
x lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
x x    [ ] apache    Apache WWW (HTTP) server                      x x
x x    [ ] procmail  Mail delivery/filtering utility               x x
x x    [ ] dip       Handles SLIP/CSLIP connections                x x
x x    [ ] ppp       Point-to-point protocol                       x x
x x    [ ] mailx     The mailx mailer                              x x
x x    [X] tcpip     TCP/IP networking programs                    x x
x x    [ ] bind      Berkeley Internet Name Domain server          x x
x x    [ ] rdist     Remote file distribution utility              x x
```

```
x x    [ ] lynx      Text-based World Wide Web browser            x x
x x    [ ] uucp      Taylor UUCP 1.06.1 with HDB && Taylor configs x x
x x    [ ] elm       Menu-driven user mail program                x x
x x    [ ] pine      Pine menu-driven mail program                x x
x x    [ ] sendmail  The sendmail mail transport agent            x x
x x    [ ] metamail  Metamail multimedia mail extensions          x x
x x    [ ] smailcfg  Extra configuration files for sendmail       x x
x x    [ ] cnews     Spools and transmits Usenet news             x x
x x    [ ] inn       InterNetNews news transport system           x x
x x    [ ] tin       The 'tin' news reader (local or NNTP)        x x
x x    [ ] trn       'trn' for /var/spool/news                    x x
x x    [ ] trn-nntp  'trn' for NNTP (install 1 'trn' maximum)     x x
x x    [ ] nn-spool  'nn' for /var/spool/news                     x x
x x    [ ] nn-nntp   'nn' for NNTP (install 1 'nn' maximum)       x x
x x    [ ] netpipes  Network pipe utilities                       x x
x mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
```

À la fin de l'installation, dites non à tout le reste (non à toutes les demandes de configuration) et sortez du script.

3.6 Installation de Linux pthreads

Vous devez maintenant installer la bibliothèque 'linuxthreads-0.71'. J'ai inclus un patch pour le Makefile de linuxthreads plutôt que d'expliquer les détails de l'installation à la main. Sauvez le Makefile d'origine, appliquez le patch et ensuite :

```
        cd /usr/src/linuxthreads-0.71
  patch
        make
        make install


-------------------diff Makefile.old  Makefile.raid-----------------
2a3,13
> # If you are building "linuxthreads" for installation on a mount
> # point which is not the "root" partition, redefine 'BUILDIR' to
> # the mount point to use as the "root" directory
> # You may wish to do this if you are building an 'initial ram disk'
> # such as used with bootable root raid devices.
> # REQUIRES ldconfig version 1.9.5 or better
> # do ldconfig -v to check
> #
> BUILDIR=/root/raidboot/mnt
> #BUILDIR=
>
81,82c92,93
<       install pthread.h $(INCLUDEDIR)/pthread.h
<       install semaphore.h $(INCLUDEDIR)/semaphore.h
---
>       install pthread.h $(BUILDIR)$(INCLUDEDIR)/pthread.h
>       install semaphore.h $(BUILDIR)$(INCLUDEDIR)/semaphore.h
```

```
84c95
<       test -f /usr/include/sched.h || install sched.h $(INCLUDEDIR)/sched.h
---
>       test -f $(BUILDIR)/usr/include/sched.h || install sched.h $(BUILDIR)$(INCLUDEDIR)/sched.h
86,89c97,103
<       install $(LIB) $(LIBDIR)/$(LIB)
<       install $(SHLIB) $(SHAREDLIBDIR)/$(SHLIB)
<       rm -f $(LIBDIR)/$(SHLIB0)
<       ln -s $(SHAREDLIBDIR)/$(SHLIB) $(LIBDIR)/$(SHLIB0)
---
>       install $(LIB) $(BUILDIR)$(LIBDIR)/$(LIB)
>       install $(SHLIB) $(BUILDIR)$(SHAREDLIBDIR)/$(SHLIB)
>       rm -f $(BUILDIR)$(LIBDIR)/$(SHLIB0)
>       ln -s $(SHAREDLIBDIR)/$(SHLIB) $(BUILDIR)$(LIBDIR)/$(SHLIB0)
> ifneq ($(BUILDIR),)
>       ldconfig -r ${BUILDIR} -n $(SHAREDLIBDIR)
> else
91c105,106
<       cd man; $(MAKE) MANDIR=$(MANDIR) install
---
> endif
>       cd man; $(MAKE) MANDIR=$(BUILDIR)$(MANDIR) install
```

3.7 Installation des outils pour le RAID

L'étape suivante est l'installation des outils RAID : raidtools-0.42.

Vous devez lancer le script "configure" pour faire pointer le Makefile vers le répertoire de construction des fichiers ramdisk.

```
cd /usr/src/raidtools-0.42
configure --sbindir=/root/raidboot/mnt/sbin --prefix=/root/raidboot/mnt/usr
make
make install
```

Maintenant ! Le Makefile pour l'installation n'est pas tout à fait correct, effectuez les opérations suivantes pour le nettoyer. Ceci sera modifié dans les versions futures pour que la liaison des liens ne soit pas à nouveau nécessaire.

Réparer l'erreur du make install

Les liens des fichiers spécifiés dans le Makefile à 'LINKS' doivent être enlevés et re-liés pour fonctionner correctement.

```
cd /root/raidboot/mnt/sbin
ln -fs mdadd mdrun
ln -fs mdadd mdstop
```

3.8 Effacement des répertoires et fichiers non nécessaires dans le nouveau système de fichiers

Effacez les répertoires suivants du système de fichiers (ATTENTION NE LES EFFACEZ PAS DE VOTRE SYSTÈME EN FONCTIONNEMENT), c'est facile à faire, devinez comment j'ai trouvé !!!

```
cd /root/raidboot/mnt
rm -r home/ftp/*
rm -r lost+found
rm -r usr/doc
rm -r usr/info
rm -r usr/local/man
rm -r usr/man
rm -r usr/openwin
rm -r usr/share/locale
rm -r usr/X*
rm -r var/man
rm -r var/log/packages
rm -r var/log/setup
rm -r var/log/disk_contents
```

3.9 Création de /dev/md*x*

La dernière étape copie simplement les périphériques /dev/md* du système de fichiers courant sur le système de fichiers de secours. Vous pouvez les créer avec mknod.

```
cp -a /dev/md* /root/raidboot/mnt/dev
```

3.10 Création d'un système de fichiers nu convenant à *initrd*

Vous avez maintenant un système de fichiers propre et réutilisable, prêt à être personnalisé. Une fois personnalisé, ce système de fichiers pourra être utilisé comme secours, si jamais le(s) périphérique(s) RAID, ainsi que les outils RAID nécessaires à la réparation, devenaient corrompus. Il sera aussi utilisé pour démarrer et monter à la racine le périphérique RAID en ajoutant le fichier linuxrc dont nous parlerons bientôt.

Copiez le système de fichiers sur un périphérique plus petit pour le fichier initrd, 16 Mo devraient être suffisants.

Créez le système de fichiers plus petit et montez-le

```
cd /root/raidboot
dd if=/dev/zero of=bare.fs bs=1024k count=16
```

associez le fichier à un périphérique de boucle et générez un système de fichiers ext2 sur ce fichier

```
losetup /dev/loop1 bare.fs
mke2fs -v -m0 -L initrd /dev/loop1
mount /dev/loop1 mnt2
```

Copiez le système de fichiers 'build' sur 'bare.fs'

```
cp -a mnt/* mnt2
```

Sauvez le système 'bare.fs' avant personnalisation pour que les mises à jour ultérieures soient facilitées. Le système de fichiers 'build' n'est plus nécessaire et peut être effacé.

```
cd /root/raidboot
umount mnt
umount mnt2
losetup -d /dev/loop0
losetup -d /dev/loop1
rm build
cp bare.fs rescue
gzip -9 bare.fs
```

3.10.1 Création du système de fichiers *initrd* de démarrage/secours

Copiez maintenant les parties dépendantes du système qui correspondent au noyau de la plate-forme de développement, ou bien vous pouvez modifier à la main les fichiers du système de fichiers de secours pour correspondre à votre système cible.

```
losetup /dev/loop0 rescue
mount /dev/loop0 mnt
```

Assurez-vous que votre répertoire etc est propre des fichiers `*~`, core et log. Les deux commandes qui suivent créent des messages d'avertissement, ignorez-les.

```
cp -dp /etc/* mnt/etc
cp -dp /etc/rc.d/* mnt/etc/rc.d

mkdir  mnt/lib/modules
cp -a  /lib/modules/2.x.x mnt/lib/modules <--- votre noyau 2.x.x en cours
```

Éditez les fichiers suivantes pour les corriger pour votre système de secours.

```
        cd mnt

Non reseau
        etc/fstab       mettez en commentaire le montage des peripheriques root et raid
        etc/mdtab       devrait fonctionner
Reseau
        etc/hosts
        etc/resolv.conf
        etc/hosts.equiv         et fichiers lies
        etc/rc.d/rc.inet1       adresses IP, masque, passerelle, etc. correctes
        etc/rc.d/rc.S           enlevez entierement la section sur l'etat du systeme de fichiers
                de :
                        # Test to see if the root partition isread-only
                a, mais n'incluant pas :
                        # remove /etc/mtab* so that mount will .....
                                ceci evite l'avertissement ennuyeux que
                                le disque RAM est monte en lecture/ecriture
        etc/rc.d/rc.xxxxx       d'autres si necessaire, voir plus bas
        root/.rhosts            si present
```

```
        home/xxxx/xxxx          d'autres si necessaire

    ATTENTION : la procedure ci-dessus deplace vos fichiers de mot de
                passe et shadow sur le disque de secours !!!

    ATTENTION : vous ne voudrez pas faire ceci pour des raisons de securite
```

Créez tous les répertoires pour monter /dev/dsk... tels qu'ils peuvent être nécessaires et uniques à votre système. Le mien nécessite :

```
        cd /root/raidboot/mnt           <--- racine initrd
        mkdir dosa                      point de montage partition DOS
        mkdir dosc                      point de montage miroir DOS
```

Le système de fichiers de secours est terminé !

Vous noterez en examinant les fichiers du système de fichiers de secours qu'il y a encore de nombreux fichiers que l'on pourrait effacer. Je ne l'ai pas fait car cela compliquerait totalement cette procédure et la plupart des systèmes RAID ont suffisamment de disque et de mémoire. Si vous voulez encore diminuer la taille du système de fichiers, allez-y !

3.10.2 Faire qu'*initrd* démarre le périphérique RAID - linuxrc

Pour que la disquette de démarrage de secours puisse démarrer le périphérique RAID, vous n'avez besoin que de copier le fichier script exécutable :

linuxrc

à la racine du périphérique.

```
--------------------- linuxrc --------------------
#!/bin/sh
# ver 1.07 2-12-98
# monter le systeme de fichiers proc
/bin/mount /proc

# Ceci peut varier pour votre systeme.
# Monter les partitions DOS, essayer les deux
# au cas ou l'un des disques est mort
/bin/mount /dosa
/bin/mount /dosc

# Positionner un drapeau au cas ou le fichier d'etat RAID n'est pas
# trouve puis chercher le fichier d'etat sur les deux disques
RAIDOWN="raidstat.ro not found"
/bin/echo "Reading md0 shutdown status."
if [ -f /dosa/linux/raidstat.ro ]; then
  RAIDOWN=`/bin/cat /dosa/linux/raidstat.ro`
  RAIDREF=`/bin/cat /dosc/linux/raidgood.ref`
else
  if [ -f /dosc/linux/raidstat.ro ]; then
```

```
    RAIDOWN=‘/bin/cat /dosc/linux/raidstat.ro‘
    RAIDREF=‘/bin/cat /dosc/linux/raidgood.ref‘
  fi
fi

# Tester si l’extinction a ete propre avec des disques en etat de
# marche
if [ "${RAIDOWN} != ${RAIDREF}" ]; then
  echo "ERROR ${RAIDOWN}"
# Utilisez les deux lignes suivantes pour SORTIR et laisser le secours
# tourner
   /bin/echo 0x100>/proc/sys/kernel/real-root-dev
   exit                 # laissant les fichier d’erreurs dans dosa/linux, etc.
fi

# La matrice RAID est propre, continuer en enlevant
# le fichier d’etat et en ecrivant un superblock propre
/bin/rm /dosa/linux/raidstat.ro
/bin/rm /dosc/linux/raidstat.ro
/sbin/mkraid /etc/raid1.conf -f --only-superblock

/bin/umount /dosa
/bin/umount /dosc

# Monter la matrice RAID
echo "Mounting md0, root filesystem"
/sbin/mdadd -ar

# S’il y a des erreurs, SORTIR et laisser le secours tourner
if [ $? -ne 0 ]; then
   echo "RAID device has errors"
# Utilisez les trois lignes suivantes pour SORTIR
   /bin/rm /etc/mtab            # enlever le mtab mauvais
   /bin/echo 0x100>/proc/sys/kernel/real-root-dev
   exit
fi

# sinon annoncer au noyau de passer a /dev/md0 comme peripherique
# /root. La valeur 0x900 est le numero de peripherique calcule par :
#  256*numero_periph_majeur + numero_periph_mineur
/bin/echo 0x900>/proc/sys/kernel/real-root-dev

# demonter /proc pour liberer l’espace memoire du peripherique initrd
/bin/umount /proc
/bin/echo "/dev/md0 mounted as root"
exit
#------------------ end linuxrc ----------------------
```

Ajoutez ’linuxrc’ au périphérique de démarrage initrd.

```
        cd /root/raidboot
```

```
chmod 777 linuxrc
cp -p linuxrc mnt
```

3.11 Modification des scripts rc pour l'extinction

Pour terminer l'installation, modifiez les scripts rc pour sauvegarder l'état md sur le vrai périphérique racine quand l'extinction du système arrive.

Dans la Slackware, c'est rc.0 -> rc.6

J'ai modifié légèrement le contournement de Bohumil Chalupa pour stopper le RAID. Sa solution à l'origine est présentée dans l'annexe 7 (A).

Puisqu'il ne reste plus de partition Linux libre sur le système de production exceptée **md0**, les partitions DOS sont utilisées pour stocker l'état **raidOK en lecture seule**. J'ai choisi d'écrire un fichier sur chaque partition DOS contenant l'état de la matrice md à l'extinction signifiant que le périphérique md a été remonté en lecture seule. Ceci évite la panne du système si l'un des disques meurt.

J'ai modifié mon script rc.6 pour tenter le démontage de la matrice racine en RAID 1 et tout autre périphériques RAID de mdtab. Vous aurez besoin de scripts légèrement différents, mais les éléments de base devraient être les mêmes. Le fichier rc.6 complet est montré dans l'annexe 8 (B).

Pour capturer l'état à l'extinction de la matrice RAID, insérez juste avant que les systèmes de fichiers soient démontés :

```
RAIDSTATUS=`/bin/cat /proc/mdstat | /usr/bin/grep md0`
```

Après que tous les systèmes de fichiers soient démontés (le système de fichiers racine ne se démontera pas), ajoutez :

```
# le peripheriques racine reste monte en lecture seule
# les systemes de fichiers DOS sont en lecture/ecriture
mount -n -o remount,ro /
echo "Writing RAID read-only boot FLAG(s)."
mount -n /dosa
mount -n /dosc
# creer un drapeau RAID monte en lecture seule en double
# contenant l'etat d'extinction de la matrice RAID
echo ${RAIDSTATUS} > /dosa/linux/raidstat.ro
echo ${RAIDSTATUS} > /dosc/linux/raidstat.ro

umount -n /dosa
umount -n /dosc

# stoppe toutes les matrices RAID (sauf la racine)
echo "Stopping raid"
mdstop -a
```

Ceci arrêtera proprement tous les périphériques RAID sauf la racine. L'état de la racine est passé au démarrage suivant dans **raidstat.ro**.

Copiez le fichier rc sur votre nouvelle matrice RAID, le système de fichiers de secours qui est encore monté sur **/root/raidboot/mnt** et le système de développement s'il est sur la même machine.

Modifiez le **etc/fstab** de secours si nécessaire et assurez-vous que le **mdtab** de secours est correct.

Copiez maintenant la disquette de secours sur votre partition DOS et tout devrait être prêt pour démarrer le périphérique RAID à la racine.

```
umount mnt
losetup -d /dev/loop0
gzip -9 rescue
```

Copiez rescue.gz sur votre partition DOS.

Tout ce qui reste est de tester le nouveau système de fichiers en redémarrant. Voyez les paramètres loadlin sur le fichier DOS **linux.bat** dans ce qui suit.

3.12 Mise en place du démarrage avec loadlin pour le secours et le RAID

Les disques que j'ai choisis pour mon système sont bien plus grands que ceux que LILO peut gérer. Par conséquent, j'ai utilisé loadlin pour démarrer le système à partir d'une petite partition DOS avec un miroir (copie) sur le disque accompagnateur.

Mon système DOS de démarrage contient un petit éditeur parmi les utilitaires pour que je puisse modifier les paramètres de démarrage de loadlin si nécessaire, ce qui me permet de redémarrer le système Linux sur mon disque de dépannage pendant que je fais des tests.

Le système DOS contient cette arborescence pour Linux :

```
c:\linux.bat
c:\linux\loadlin.exe
c:\linux\zimage
c:\linux\rescue.gz
c:\linux\raidgood.ref
c:\linux\raidstat.ro    (seulement a l'extinction)
```

linux.bat contient :

```
---------------------- linux.bat ---------------------------
rem  fichier batch DOS d'exemple pour demarrer Linux
rem  Demarrage du processus LOADLIN :

rem c:\linux\loadlin c:\linux\zimage root=/dev/ram0 ro ramdisk_size=16384 initrd=c:\linux\rescue.gz
c:\linux\loadlin c:\linux\zimage root=/dev/md0 ro ramdisk_size=16384 initrd=c:\linux\rescue.gz mem=1

rem -- ceci est mon systeme de developpement -- il s'en va plus tard
rem c:\linux\loadlin c:\linux\zimage root=/dev/hda3 ro noinitrd mem=131072k
------------------------------------------------------------

***** >> NOTE !! la seule difference entre forcer le lancement du
        systeme de secours et le montage du peripherique RAID est le
        parametre de loadlin.

                root=/dev/ram0          pour le systeme de secours
                root=/dev/md0           pour RAID

                Avec root=/dev/ram0 les peripheriques RAID ne seront
```

```
                pas montes et le systeme de secours tournera sans
                conditions
```

Si la matrice RAID tombe en panne, le système de secours reste monté et tourne (ceci semble ne pas fonctionner de temps en temps, je ne sais pas pourquoi, cela fonctionne quand le bouton reset est appuyé mais ne fonctionne pas avec 'shutdown -r now').

4 Configuration du système de production en RAID

4.1 Spécifications du système

```
Carte mere :    Iwill P55TU     double ide + adaptec scsi
Processeur :    Intel P200
Disques:                2 ea. Maxtor 7 gig eide
```

Les disques sont désignés par Linux comme 'hda' et 'hdc'

4.2 Partitionnement des disques durs

Puisque le test d'une grande matrice RAID qu'on peut monter à la racine est difficile à cause du problème du redémarrage, j'ai repartitionné mon espace d'échange (swap) pour inclure une partition RAID plus petite à des fins de tests. Vous pourrez trouver ceci utile.

```
        <bf/SYSTEME DE DEVELOPPEMENT/

  /dev/hda1     dos             16meg
* /dev/hda2     extended        126m
  /dev/hda3     linux           126m    partition racine pendant le developpement
  /dev/hda4     linux           6+gig   RAID 1
* /dev/hda5     linux            26m    RAID 1 de test
* /dev/hda6     linux swap      100m

  /dev/hdc1     est simplement une copie exacte de hda1 pour que cette
                partition puisse etre activee si hda tombe en panne
* /dev/hdc2     extended        126m
  /dev/hdc3     linux           126m    /usr/src pendant le developpement
  /dev/hdc4     linux           6+gig   miroir RAID 1
* /dev/hdc5     linux            26m    miroir RAID 1 de test
* /dev/hdc6     linux swap      100m

        <bf/SYSTEME DE PRODUCTION/

/dev/hda1       dos             16meg
/dev/hda2       linux swap      126m
/dev/hda3       linux swap      126m
/dev/hda4       linux           6+gig   RAID 1

/dev/hdc1       est simplement une copie exacte de hda1
/dev/hdc2       linux swap      126m
```

```
/dev/hdc3       linux swap      126m
/dev/hdc4       linux           6+gig   miroir RAID 1
```

Les partitions hdx3 ont été changées en 'swap' après avoir développé cet utilitaire. J'aurais pu le faire sur une autre machine, mais les bibliothèques et les noyaux ont tous à peu près un an de retard sur mes autres machines Linux et j'ai préféré le construire sur la machine cible.

J'ai choisi de partitionner de cette façon et d'utiliser loadlin plutôt que LILO parce que :

1. la partition principale (6 Go) est trop grande pour pouvoir démarrer avec LILO seul et aurait demandé une partition supplémentaire plus petite située à l'intérieur des 1024 premières adresses du disque ;
2. Si jamais ce disque tombe en panne de manière catastrophique, le système doit continuer à tourner et être démarrable avec un minimum d'efforts et AUCUNE perte de données.
   - Si l'un des disques durs tombe en panne, le démarrage s'arrêtera, et le système de secours se mettra en marche. L'examen du message à l'écran ou de /dos$x$/linux/raidstat.ro indiquera à l'administrateur l'état de la matrice défaillante.
   - Si hda tombe en panne, la partition DOS sur hdc doit être marquée 'active' et le BIOS doit reconnaître hdc comme périphérique de démarrage ou bien il doit être déplacé physiquement en position hda en le recâblant. On peut réactiver à nouveau le système RAID en enlevant le disque défectueux et en tapant :

     ```
     "/sbin/mkraid /etc/raid1.conf -f --only-superblock"
     ```

     pour reconstruire le superblock restant.
   - Une fois ceci fait, alors

     ```
     mdadd -ar
     ```

   - Examinez l'état de la matrice pour vérifier que tout est en ordre, puis remplacez la référence de la matrice en bon état par l'état courant jusqu'à ce que le disque défectueux puisse être réparé et remplacé.

     ```
     cat /proc/mdstat | grep md0 > /dosa/linux/raidgood.ref

     shutdown -r now
     ```

     pour effectuer un redémarrage propre, et le système sera à nouveau disponible.

5 Construction du système de fichiers RAID

Cette description convient à mon système RAID 1 décrit dans les spécifications du système. Votre système peut avoir une architecture RAID différente, vous ferez les modifications appropriées. Veuillez lire les pages de manuel et le QuickStart.RAID livré avec les raidtools-0.42. Mon /etc/raid1.conf contient :

```
# configuration RAID 1
raiddev                 /dev/md0
raid-level              1
nr-raid-disks           2
nr-spare-disks          0

device                  /dev/hda4
```

```
raid-disk               0

device                  /dev/hdc4
raid-disk               1
```

5.1 Procédures pas à pas pour la construction d'un système de fichiers RAID de production

Pour mon système RAID 1, j'ai fait une installation complète de :

```
Slackware-3.4
linuxthreads-0.71
raidtools-0.42
linux-2.0.32 avec le patch raid145
```

Créez et formatez le périphérique RAID.

```
mkraid /etc/raid1.conf
mdcreate raid1 /dev/md0 /dev/hda4 /dev/hdc4
mdadd -ar
mke2fs /dev/md0
mkdir /md
mount -t ext2 /dev/md0 /md
```

Créez les fichiers de référence que reboot utilisera, ils peuvent être différents sur votre système.

```
cat /proc/mdstat | grep md0 > /dosa/linux/raidgood.ref
cat /proc/mdstat | grep md0 > /dosc/linux/raidgood.ref
```

Utilisez Slackware 3.4 ou une autre distribution pour construire votre système d'exploitation.

```
setup
```

Spécifiez 'md' comme cible, et comme source ce que vous utilisez habituellement. Sélectionnez et installez les ensembles de disques intéressants sauf pour le noyau. Configurez le système, mais passez les sections sur LILO et le démarrage du noyau. Sortez de setup.

Installez les 'pthreads'

```
cd /usr/src/linuxthreads-0.71
```

Éditez le Makefile et spécifiez

```
BUILDIR=/md

make
make install
```

Installez les 'raidtools'

```
cd /usr/src/raidtools-0.42
configure --sbindir=/md/sbin --prefix=/md/usr
```

Réparez l'erreur du make install des raidtools

```
cd /md/sbin
rm mdrun
rm mdstop
ln -s mdadd mdrun
ln -s mdadd mdstop
```

Créez /dev/mdx

```
cp -a /dev/md* /md/dev
```

Ajoutez la configuration système du système en cours (ignorez les erreurs).

```
cp -dp /etc/* mnt/etc
cp -dp /etc/rc.d/* mnt/etc/rc.d         (incluez le nouveau rc.6)
mkdir  mnt/lib/modules
cp -a  /lib/modules/2.x.x mnt/lib/modules <--- votre 2.x.x en cours
```

Éditez les fichiers suivants pour les corriger pour votre système de fichiers

```
cd /md
```

```
Non reseau
        etc/fstab       correct pour les peripheriques racine reelle et RAID
        etc/mdtab       devrait etre correct
Reseau
        etc/hosts
        etc/resolv.conf
        etc/hosts.equiv         et fichiers associes
        etc/rc.d/rc.inet1       adresses IP, masque, passerelle, etc. correctes
        etc/rc.d/rc.S           enlevez la section entiere sur l'etat du systeme de fichiers
                a partir de :
                        # Test to see if the root partition isread-only
                jusqu'a mais ne comprenant pas :
                        # remove /etc/mtab* so that mount will .....
                                Ceci evite l'avertissement ennuyeux que
                                le disque RAM est monte en lecture/ecriture
        etc/rc.d/rc.xxxxx       d'autres si necessaire
        root/.rhosts            s'il existe
        home/xxxx/xxxx          d'autres si necessaire

    ATTTENTION :        La procedure ci-dessus deplace vos fichiers de
                        mot de passe et shadow sur le nouveau systeme
                        de fichiers !!!

    ATTENTION :         Vous n'aurez pas envie de faire ceci pour des
                        raisons de securite
```

Créez tous les répertoires pour monter /dev/disk... qui pourraient être nécessaire compte-tenu de votre système. Le mien nécessite :

```
cd /md          <--- nouvelle racine du systeme de fichiers
mkdir dosa              point de montage de la partition DOS
mkdir dosc              point de montage du miroir DOS
```

Le nouveau système de fichiers est terminé. Assurez-vous de sauvegarder l'état de référence de md sur le 'véritable' périphérique racine et vous êtes prêt à démarrer.

montez les partitions DOS sur dosa et dosc

```
cat /proc/mdstat | grep md0 > /dosa/linux/raidgood.ref
cat /proc/mdstat | grep md0 > /dosc/linux/raidgood.ref

mdstop /dev/md0
```

6 Une dernière pensée

Souvenez-vous qu'un expert est quelqu'un qui en connait au moins 1% de plus que vous sur un sujet. Gardez ceci à l'esprit quand vous m'enverrez un e-mail demandant de l'aide. J'essaierai, mais je n'ai fait ceci qu'une fois !

Michael Robinton *Michael@bzs.org* `mailto:michael@bzs.org`

7 Annexe A. - extinction de md0 chez Bohumil Chalupa

Voici le post de Bohumil Chalupa sur la liste Linux RAID concernant le contournement du problème de mdstop avec RAID 1 + 5. Sa solution ne considère pas la possibilité d'une corruption du périphérique RAID à l'extinction. J'ai donc ajouté une simple comparaison d'état avec un état de référence correct et connu au démarrage. Ceci permet à l'administrateur d'intervenir si quelque chose ne va pas bien avec un disque dans une matrice. La description de ceci est dans le corps principal de ce document.

(NdT : j'ai traduit le message parce qu'il est long et intéressant, celui en anglais se trouve bien évidemment dans la version d'origine de ce HOWTO.)

```
> From: Bohumil Chalupa <bochal@apollo.karlov.mff.cuni.cz>
>
> Je peux maintenant demarrer initrd et utiliser linuxrc pour demarrer
> la matrice RAID 1, puis changer la racine en /dev/md0 avec succes.
>
> Je ne connais pas, cependant, de manière d'_arreter_ proprement la
> matrice.

Bon. Je dois repondre moi-meme :-)

> Date: Mon, 29 Dec 1997 02:21:38 -0600 (CST)
> From: Edward Welbon <welbon@bga.com>
> Subject: Re: dismounting root raid device
>
> Pour les peripheriques md autres que RAID 0, il y a surement un etat
> a sauvegarder qui n'est connu qu'une fois que toutes les ecritures
> sont terminees. Un tel etat ne peut bien sur etre sauve a la racine
```

```
> une fois qu'elle est montee en lecture seule. Dans ce cas, on
> devrait pouvoir monter un systeme de fichiers "X" en ecriture sur la
> racine en lecture seule et pouvoir ecrire sur "X" (je me rappelle
> l'avoir fait pendant des operations de "secours", mais pas dans une
> procedure automatisee).
>
> Le systeme de fichiers "X" serait je pense un peripherique de
> demarrage a partir duquel le RAID (pendant l'execution de linuxrc à
> travers initrd) irait prendre son etat initial. Heureusement RAID 0
> ne demande pas d'ecrire un quelconque etat (bien qu'il serait utile
> de pouvoir ecrire les sommes de vérification (checksums) dans mdtab
> apres un mdstop). En fin de compte, je bricolerai ceci mais ca ne
> parait pas difficile bien que le "diable" se trouve toujours dans
> les "details".

Oui, c'est ca.
J'ai deja eu cette idee a l'esprit depuis quelques temps, mais je n'ai
pas eu le temps de l'essayer. Je l'ai fait hier, et ca marche.

Avec mon RAID 1 (miroir), je ne sauvegarde aucune somme de
verification ni donnees de superblock RAID. Je ne sauvegarde une
information que sur la partition de demarrage "reelle", celle sur
laquelle le volume md racine etait remonte en lecture seule pendant
l'extinction. Alors, pendant le demarrage, le script linuxrc lance
mkraid --only-superblock quand il trouve cette information ; sinon, il
lance ckraid. Ceci veut dire que l'information du superblock RAID
n'est pas mise a jour pendant l'extinction ; elle est mise a jour au
demarrage. Ce n'est pas tres propre, j'en ai peur, :-( mais ca
fonctionne.

J'utilise Slackware et initrd.md d'Edward Welbon pour demarrer le
peripherique RAID a la racine.
Autant que je puisse m'en souvenir, les seuls fichiers modifies sont
mkdisk et linuxrc, et le script d'extinction /etc/rc.d/rc.6. Et
lilo.conf, bien sur.

J'annexe les parties importantes.

Bohumil Chalupa

--------------- mon.linuxrc suit -----------------
#!/bin/sh
# nous avons besoin de /proc
/bin/mount /proc
# demarrer le périphérique md0. Laisser les scripts /etc/rc.d
s'occuper du reste. Nous devrions en faire le moins possible ici.
# ----------------------------------------
# test d'extinction du RAID 1 racine et recreation
# /start doit etre cree sur l'image rd dans mon.mkdisk
echo "preparation de md0: montage de /start"
```

```
/bin/mount /dev/sda2 /start -t ext2
echo "lecture de l'etat sauvegarde de md0 dans /start"
if [ -f /start/root.raid.ok ]; then
 echo "RAID ok, modification du superblock"
 rm /start/root.raid.ok
 /sbin/mkraid /etc/raid1.conf -f --only-superblock
else
 echo "RAID pas propre, lancement de ckraid --fix"
 /sbin/ckraid --fix /etc/raid1.conf
fi
echo "demontage de /start"
/bin/umount /start
# ----------------------------------------
#
echo "ajout de md0 pour le systeme de fichiers racine"
/sbin/mdadd /dev/md0 /dev/sda1 /dev/sdb1
echo "lancement de md0"
/sbin/mdrun -p1 /dev/md0
# dire au noyau que nous voulons passer /dev/md0 comme peripherique
# racine, la valeur 0x900 vient de 256*numero_periph_majeur +
# numero_periph_mineur.
echo "mise en place du vrai-periph-racine (real-root-dev)"
/bin/echo 0x900>/proc/sys/kernel/real-root-dev
# demontage de /proc pour que le disque RAM puisse etre enleve de la
# memoire
echo "unmounting /proc"
/bin/umount /proc
/bin/echo "On peut normalement monter /dev/md0 (majeur 9, mineur 0) a la racine"
exit
-------------- fin de mon.linuxrc ----------------------------------


----------- extrait de /etc/rc.d/rc.6 suit -----------------
  # arreter la swap, puis demonter les systemes de fichiers locaux
  echo "Extinction de la swap."
  swapoff -a
  echo "Demontage des systemes de fichiers locaux."
  umount -a -tnonfs
  # Ne pas remonter les volumes UMSDOS racine :
  if [ ! "`mount | head -1 | cut -d ' ' -f 5`" = "umsdos" ]; then
    mount -n -o remount,ro /
  fi

  # Sauvegarder l'etat RAID
  echo "Sauvegarde de l'etat RAID"
  /bin/mount -n /dev/sda2 /start -t ext2
  touch /start/root.raid.ok
  /bin/umount -n /start

------------- fin de l'extrait de rc.6 -----------------------
```

```
------------------ une partie de mon.mkdisk suit ----------------------
#
#  maintenant que le systeme de fichiers est pret a etre rempli, nous
#  devons obtenir quelques repertoires importants. J'ai eu des
#  problemes sans fin jusqu'a ce que je cree un fichier mtab tout
#  neuf. Dans mon cas, il est pratique de recopier /etc/mdtab, de
#  cette maniere je peux activer md avec un simple "/sbin/mdadd -ar"
#  dans linuxrc.
#
cp -a $ROOT/etc $MOUNTPNT 2>cp.stderr 1>cp.stdout
rm -rf $MOUNTPNT/etc/mtab
rm -rf $MOUNTPNT/etc/ppp*
rm -rf $MOUNTPNT/etc/termcap
rm -rf $MOUNTPNT/etc/sendmail*
rm -rf $MOUNTPNT/etc/rc.d
rm -rf $MOUNTPNT/etc/dos*
cp -a $ROOT/sbin $ROOT/dev $ROOT/lib $ROOT/bin $MOUNTPNT 2>>cp.stderr
1>>cp.stdout
# ---------------------------------------------------------------------
#  RAID: aura besoin de mkraid et ckraid
cp -a $ROOT/usr/sbin/mkraid $ROOT/usr/sbin/ckraid $MOUNTPNT/sbin
2>>cp.stderr 1>>cp.stdout
#
---------------------------------------------------------------------
#  il semble qu'init ne voudra pas jouer sans utmp. on peut probablement
#  reduire ceci de beaucoup. aucune idee de ce que le vrai bug était 8-).
#
mkdir $MOUNTPNT/var $MOUNTPNT/var/log $MOUNTPNT/var/run $MOUNTPNT/initrd
touch $MOUNTPNT/var/run/utmp $MOUNTPNT/etc/mtab
chmod a+r $MOUNTPNT/var/run/utmp $MOUNTPNT/etc/mtab
ln -s /var/run/utmp $MOUNTPNT/var/log/utmp
ln -s /var/log/utmp $MOUNTPNT/etc/utmp
ls -lstrd $MOUNTPNT/etc/utmp $MOUNTPNT/var/log/utmp $MOUNTPNT/var/run/utmp
#
#  puisque je voulais changer le point de montage, j'avais besoin de
#  ceci bien que je suppose que j'aurais pu faire un "mkdir /proc"
#  dans linuxrc.
#
mkdir $MOUNTPNT/proc
chmod 555 $MOUNTPNT/proc
#
#  -------------------------------------------------------
#  nous monterons le vrai peripherique de demarrage sur /start de
#  maniere temporaire pour verifier l'etat du RAID racine sauve au
#  moment de l'extinction
#
mkdir $MOUNTPNT/start
#  -------------------------------------------------------
```

```
#
#  besoin de linuxrc (c'est, apres tout, le but de l'exercice).
#
if [ -x ./my.linuxrc ]; then
  cp -a ./my.linuxrc $MOUNTPNT/linuxrc
  chmod 777 $MOUNTPNT/linuxrc
else
   ln -s /bin/sh $MOUNTPNT/linuxrc
fi
#
----------------- fin de la partie de mon.mkdisk -----------------
```

8 Annexe B. - fichier rc.0 - rc.6 complet

```
#! /bin/sh
#
# rc.6          Ce fichier est execute par init quand il passe au
#               runlevel (niveau d'execution) 0 (halt) ou runlevel 6
#               (redemarrage). Il tue tous les processus, demonte les
#               systemes de fichiers et soit stoppe soit redemarre.
#
# Version:      @(#)/etc/rc.d/rc.6      1.50    1994-01-15
#
# Author:       Miquel van Smoorenburg <miquels@drinkel.nl.mugnet.org>
# Modified by:  Patrick J. Volkerding, <volkerdi@ftp.cdrom.com>
# Modified by:  Michael A. Robinton, <michael@bzs.org> for RAID shutdown

  # Positionner le chemin.
  PATH=/sbin:/etc:/bin:/usr/bin

  # Mettre en mode saut de ligne pour eviter l'effet d'escalier.
  stty onlcr

  echo "Lancement du script d'extinction $0:"

  # Trouver comment nous sommes appeles.
  case "$0" in
        *0)
                message="Le systeme est stoppe."
                command="halt"
                ;;
        *6)
                message="Redemarrage."
                command=reboot
                ;;
        *)
                echo "$0: veuillez m'appeler en tant que \"rc.0\" or \"rc.6\" !"
                exit 1
                ;;
```

```
esac

# Tuer tous les processus.
# INIT est sense s'occuper entierement de tout ceci, mais il n'a pas
# toujours fonctionne correctement sans cette seconde passe de
# tuerie de processus. Puisqu'INIT a déjà averti les utilisateurs
# que les processus etaient tues, nous eviterons d'afficher cette
# information cette fois-ci.
if [ "$1" != "fast" ]; then # shutdown n'a pas encore tue tous les processus
  killall5 -15
  killall5 -9
fi

# Essayer d'arreter les quotas et la comptabilite.
if [ -x /usr/sbin/quotaoff ]
then
      echo "Arret des quotas."
      /usr/sbin/quotaoff -a
fi
if [ -x /sbin/accton ]
then
      echo "Arret de la comptabilite."
      /sbin/accton
fi

# Avant de demonter les systemes de fichiers ecrire un
# enregistrement reboot ou halt dans wtmp.
$command -w

# Sauvegarde de l'heure locale
[ -e /usr/lib/zoneinfo/localtime ] && cp /usr/lib/zoneinfo/localtime /etc

# Demontage asynchrone de tout systeme de fichiers distant
echo "Demontage des systemes de fichiers distants."
umount -a -tnfs &

# vous devez avoir lance
# 'cat /proc/mdstat | grep md0 > {your boot vol}/linux/raidgood.ref'
# avant que linuxrc s'execute proprement avec cette information
RAIDSTATUS=`/bin/cat /proc/mdstat | /usr/bin/grep md0 # capture raid status`

# Arret de la swap, puis demontage des systemes de fichiers locaux.
# effacement de mdtab par la meme occasion
echo "Arret de la swap."
swapoff -a
echo "Demontage des systemes de fichiers locaux."
umount -a -tnonfs

# Ne pas remonter les volumes UMSDOS racine :
if [ ! "`mount | head -1 | cut -d ' ' -f 5`" = "umsdos" ]; then
```

```
  mount -n -o remount,ro /
fi

# le peripherique racine reste monte
# monter les systemes de fichiers DOS en lecture/ecriture
echo "Ecriture des drapeaux RAID de demarrage en lecture seule."
mount -n /dosa
mount -n /dosc
# creation du drapeau RAID monte en lecture seule en double
# contenant l'etat a l'extinction de la matrice RAID
echo ${RAIDSTATUS} > /dosa/linux/raidstat.ro
echo ${RAIDSTATUS} > /dosc/linux/raidstat.ro

umount -n /dosa
umount -n /dosc

# Arreter toutes les matrices RAID (sauf la racine)
echo "Arret du RAID"
mdstop -a

# Voir si c'est une situation de coupure de courant.
if [ -f /etc/power_is_failing ]; then
  echo "Extinction de l'UPS, a+."
  /sbin/powerd -q
  exit 1
fi

# Maintenant arret ou redemarrage.
echo "$message"
[ ! -f /etc/fastboot ] && echo "Au prochain demarrage fsck sera FORCE."
$command -f
```