Root RAID HOWTO cookbook

Michael A. Robinton, *michael@bzs.org* v1.07, 25 marzo 1998

Questo documento è un "ricettario" per la creazione di un filesystem raid montato come root (root raid) e del suo compagno ideale, un sistema di recupero, usando initrd. Le istruzioni sono complete e date passo a passo, sia per dispositivi md0 raid1 che raid5. Per ciascun passo viene data una spiegazione di ciò che con esso si vuole ottenere. In questa revisione c'è anche un file generico **linuxrc** per initrd che può essere configurato con un singolo file di poche righe 4.13 (/etc/raidboot.conf) per configurazioni raid1 e raid5. La traduzione italiana è curata da Samuele Maretti *s.maretti@tiscalinet.it*

Contents

1 Introduzione

Si assume che il lettore abbia familiarità con i vari tipi di implementazione raid, i loro vantaggi ed effetti collaterali. Questo non è un tutorial, solo un insieme di istruzioni su come implementare un filesystem raid su un sistema linux. Tutte le informazioni necessarie per familiarizzare con raid su linux sono elencate qui direttamente o per riferimento: per favore leggetele prima di inviare domande per e-mail.

1.1 Dove ottenere copie aggiornate di questo documento.

Root-RAID-HOWTO

Disponibile in formato LaTeX (per DVI e PostScript), testo e HTML.

> *sunsite.unc.edu/mdw/HOWTO/*

Disponibile in formato SGML e HTML.

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

1.2 Bug

Al momento della stesura di questo documento il problema di fermare un dispositivo RAID montato come root non è ancora stato risolto in modo soddisfacente. Un metodo per eliminare il bisogno di eseguire ckraid (che richiede un sacco di tempo) ad ogni boot, proposto da Ed Welbon e implementato da Bohumil Chalupa si trova in questo documento. Senza tale metodo è necessario eseguire **ckraid** sul dispositivo **md** ogni volta che viene fatto un reboot del sistema. Su un array troppo grande questo può causare un grande scadimento delle prestazioni. Sul mio dispositivo RAID1 da 6 giga, installato su una macchina con processore Pentium 166 con 128 mega di ram ci vuole ben più di mezz'ora per eseguire ckraid :-(dopo ogni reboot. Sul mio array RAID5 da 13 giga con adattatore scsi da 20mb/sec ci vuole più di un'ora.

Il metodo consiste nel memorizzare, ad ogni shutdown, lo stato dell'array sul dispositivo di boot vero e proprio e confrontarlo con uno "stato di riferimento" là memorizzato quando il sistema è stato costruito la prima volta. Se al reboot i due stati coincidono il superblock dell'array viene ricostruito al boot successivo, altrimenti l'operatore viene avvisato dell'errore di stato e viene lasciato in esecuzione il sistema di recupero con tutti gli strumenti raid.

La ricostruzione del superblock fa sì che il sistema ignori che l'array è stato spento senza eseguire mdstop marcando tutti i drive come **OK**, come se nulla fosse accaduto. Questo funziona solo se tutti i drive sono OK al momento dello shutdown. Se uno dei drive dell'array ha dei problemi è necessario rimuovere tale drive prima di far ripartire il dispositivo md, o si potrebbe avere perdita di dati.

Niente di tutto questo si applica a raid0, su cui non deve essere eseguito mdstop prima dello shutdown.

Le soluzioni definitive proposte per questo problema includono l'introduzione di un file **finalrd** simile ad **initrd**, e un comando **mdrootstop** che, durante lo shutdown, scrive sull'array i flag **clean** quando è montato in sola lettura. Sono sicuro che ce ne sono altre.

Nel frattempo il problema è stato solo aggirato. Per favore, fatemi sapere quando verrà risolto in modo più pulito!!!

1.3 Riconoscimenti

Ciò che le persone seguenti hanno scritto o hanno inviato per e-mail mi ha aiutato a realizzare questo documento. Molte idee sono state *rubate* dal lavoro di altri, io ho solo provato a mettere tutto insieme nella forma di un **libro di ricette** in modo che fosse facile da usare. I miei ringraziamenti vanno a:

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

  per il RAID howto che mi ha spiegato la maggior parte delle cose.

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

  per aver risposto alle mie sciocche domande da principiante.

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

  per l'eccellente pacchetto ***initrd.md*** *che mi ha ispirato a scrivere questo.*

- *Bohumil Chalupa* `mailto:bochal@apollo.karlov.mff.cuni.cz`

  per aver implementato il metodo di cui ho parlato e che permette ai dispositivi raid montati come root di funzionare in modo produttivo.

- e molti altri che, in un modo o nell'altro, hanno contribuito a questo lavoro.

1.4 Informazioni di copyright (in lingua originale)

2 Di cosa hai bisogno PRIMA DI COMINCIARE

I pacchetti di cui hai bisogno e la documentazione che risponde alle domande più comuni sulla configurazione e l'utilizzo di raid sono elencati sotto. Leggi approfonditamente l'elenco.

2.1 Pacchetti necessari

Hai bisogno della versione più recente di questi pacchetti.

- un kernel che supporti raid, initrd e /dev/loopx

  > io ho usato *linux-2.0.33* `ftp://sunsite.unc.edu/pub/Linux/kernel/`
  > da sunsite

- *raid145-971022-2.0.31* `ftp://ftp.kernel.org/pub/linux/daemons/raid/`

  patch che aggiunge il supporto per raid1/4/5

- *raidtools-pre3-0.42* `ftp://ftp.kernel.org/pub/linux/daemons/raid/`

  strumenti per creare e mantenere dispositivi raid (c'è anche la documentazione).

- 12 (La patch di Gadi per il raid stop) nell'Appendice E.

- *linuxthreads-0.71* `ftp://ftp.inria.fr/INRIA/Projects/cristal/Xavier.Leroy`

  pacchetto threads necessario. Deve essere scaricato con ftp: i browser non funzionano. ftp.inria.fr/INRIA/Projects/cristal/Xavier.Leroy

- Una distribuzione Linux pronta da installare.

  Io ho usato
  Slackware-3.4 `ftp://ftp.cdrom.com/pub/linux`

Utile, ma non necessario

- *raidboot-0.01.tar.gz* `ftp://ftp.bizsystems.com/pub/raid/`

  sistema di recupero e boot raid precostruito.

Le istruzioni dettagliate che trovate in questo documento sono basate sui pacchetti citati. Se i pacchetti sono stati aggiornati o usate una distribuzione linux diversa potreste dover modificare le procedure che trovate qui.

Le patch, gli strumenti ecc... potrebbero essere diversi con i kernel 2.1. Controllate la documentazione più recente a:

ftp.kernel.org/pub/linux/daemons/raid/

2.2 Altre implementazioni simili.

Ho deciso di includere nel kernel tutte le parti necessarie in modo che tutto funzioni a partire dal boot senza caricare alcun modulo. L'immagine del kernel che ho ottenuto, compressa, è un po' più grande di 300k.

Date un'occhiata all'**initrd.md.tar.gz** di *Ed Welbon* `mailto:welbon@bga.com` dove trovate un altro modo per avere un dispositivo raid bootabile. Usa i moduli caricabili. Un'occhiata ai suoi script molto concisi vi mostreranno come fare se avete bisogno di un kernel molto piccolo con moduli.

http://www.realtime.net/˜welbon/initrd.md.tar.gz

2.3 Documentazione – Letture raccomandate

Dovreste leggere:

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

così come la documentazione e le pagine di manuale che accompagnano il pacchetto raidtools. In particolare leggete **man mdadd** e il documento **QuickStart.RAID** che sono inclusi in questo pacchetto.

Potrebbe essere interessante leggere anche:

- *BootPrompt-HOWTO* `http://sunsite.unc.edu/mdw/HOWTO/BootPrompt-HOWTO.html`
- **man lilo**
- **man lilo.conf**

2.4 Risorse RAID

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

Potete anche sottoscrivere le seguenti mailing list:

- *majordomo@nuclecu.unam.mx per iscrivervi* **subscribe raiddev**mandate un messaggio a: *raiddev@nuclecu.unam.mx*
- *majordomo@vger.rutgers.edu per iscrivervi* **subscribe linux-raid**mandate un messaggio a: *linux-raid@vger.rutgers.edu*

  (questa sembra essere la lista più attiva)

3 Partenza veloce con ROOT RAID

Se non vuoi compilare e fare il debug del sistema di recupero puoi prenderne uno generico incluso nella Slackware-3.4 all'indirizzo:

> *ftp.bizsystems.com/pub/raid/raidboot-0.01.tar.gz*

A questo punto segui i passi seguenti:

- Compila un kernel con raid attivo e con il supporto per i tuoi dischi (deve essere contenuto nel kernel, non un modulo)
- Controlla che l'array raid sia ben configurato e venga montato correttamente
- Installa il tuo sistema operativo sul sistema raid
- Correggi le voci in **fstab** in modo che **/dev/md0** sia il dispositivo di root. Assicurati anche che le partizioni che usi per il boot si trovino in **fstab**.
- Modifica i tuoi script di shutdown e reboot (i miei si trovano in /etc/rc.d/rc.6) come descritto in 4.12 (Modificare gli script rc per lo shutdown)
- Segui i passi seguenti: dal filesystem usato per i passaggi eseguiti finora (d'ora in poi "sistema di sviluppo") nel sistema di recupero e nel nuovo sistema raid

  ```
  cd /root/raidboot
  mkdir mnt
  gzip -d rescue.clean
  losetup /dev/loop0  rescue.clean
  mount /dev/loop0    mnt
  ```

  ```
  copia questi file
  ```

```
        cp -p /etc/*        mnt/etc
        cp -p /etc/rc.d/*   mnt/etc/rc.d
                {o nel modo più appropriato per il tuo sistema}
        cp -a /lib/modules/* mnt/lib/modules
```

Correggi le voci in **fstab** in modo che **/dev/md0** sia il dispositivo di root. Assicurati che le partizioni che usi per il boot appaiano in **fstab**.

Crea **/etc/raidboot.conf** che descriva la configurazione di boot raid. Questo file **NON** può contenere commenti nelle prime tre linee, dopo queste non ci sono problemi.

raidboot.conf

```
        /dev/sda1 /dev/sda2
        raidboot
        raid5.conf
# eventuali commenti possono essere messi *dopo* le tre
# linee di configurazione.
#
# Questo è 'raidboot.conf'
#
# linea uno, le partizioni contenenti il sistema di recupero raid 'initrd'
#       Non è necessario eseguire il boot da queste partizioni; comunque,
#       poiché il sistema di recupero non entra in un floppy, è necessario
#       sapere quali partizioni devono essere usate per caricarlo.
#
# linea due, il percorso delle informazioni di configurazione raid di boot
#       dove, a tempo di boot, si trovano shutdown, status ecc.
#       NON include le informazioni sul punto di mount, solo 'percorso':
#       /punto_di_mount/'percorso'
#
# linea tre, nome del file di configurazione raid
#       File contenente la configurazione raid corrente: raid1.conf, raid5.conf
```

Mancano poche altre cose ed il sistema raid è pronto ad eseguire il boot.

Crea 13 (rc.raidown), come descritto in appendice F, e copialo in /etc/rc.d sui sistemi di recupero, sviluppo e raid. Smonta il sistema di recupero e comprimilo.

```
        umount mnt
        losetup -d /dev/loop0
        mv rescue.clean rescue
        gzip rescue
```

Copia il file di recupero sulle partizioni di raid di boot.

```
        cp rescue.gz /mnt_point(1)/raidboot
        cp rescue.gz /mnt_point(2)/raidboot
```

Attiva l'array raid.

```
        mdadd -ar
```

Salva lo stato di riferimento **corretto** sulla partizione raid di boot.

```
cat /proc/mdstat | grep md0 > /mnt_point(1)/raidboot/raidgood.ref
cat /proc/mdstat | grep md0 > /mnt_point(1)/raidboot/raidgood.ref
```

Infine configura il programma di boot come indicato in 4.13 (Configurare RAIDBOOT - raidboot.conf) e riavvia il tuo sistema dall'array raid.

4 initrd - Ricette per il RAID montato come root

Questa è la procedura per creare un ramdisk 'initrd' con strumenti di recupero per il raid.

Nello specifico, questo documento si riferisce alle implementazioni RAID1 e RAID5.

4.1 Note sulla sicurezza

Il filesystem di recupero può essere usato da solo. Nel caso in cui ci fossero problemi a montare l'array raid vieni lasciato con il sistema di recupero montato e funzionante. PRENDI LE PRECAUZIONI DI SICUREZZA APPROPRIATE!!!

4.2 Compilare il Kernel e gli strumenti Raid

La prima cosa da fare è applicare una patch e compilare il kernel e familiarizzarsi con gli strumenti raid. Assicurati di usare anche 12 (La patch di Gadi per il raid stop) in appendice E. Configura, monta e prova i tuoi dispositivi raid. I dettagli su come questo deve essere fatto sono inclusi nel pacchetto **raidtools** e li vedremo in breve più avanti in questo stesso documento.

4.3 Costruire i filesystem *initrd* di recupero e di boot.

Ho usato la distribuzione **Slackware-3.4** per costruire i filesystem di recupero e boot e il filesystem per la macchina di produzione. Tutte le distribuzioni di linux dovrebbero andare bene. Se usi una distribuzione diversa, controlla le parti di questa procedura specifiche per Slackware e modificale in modo ad adattarle al tuo caso.

Uso loadlin per eseguire il boot dell'immagine del kernel da una partizione dos semplicemente perché nel mio sistema ci sono dei dispositivi strani che hanno bisogno di software di configurazione che gira solo sotto dos. Lilo funzionerà altrettanto bene e può essere usata una piccola partizione linux contenente solo i file di boot raid e il record di **lilo**.

Per il sistema di boot/recupero raid, ho deciso di creare un sistema ramdisk di minime dimensioni usando lo script 'setup' di Slackware seguito dall'installazione dei pacchetti 'linuxthreads' e 'raidtools' sulla Slackware sul mio ramdisk. Ho usato una procedura del tutto identica per costruire il sistema di produzione. Così i sistemi di recupero e produzione sono molto simili.

Questo processo di installazione dà, alla fine, un sistema piuttosto semplice (salva una copia del file) a cui sovrappongo

```
/lib/modules/2.x.x......
/etc .... con fstab, mdtab, raidX.conf, raidboot.conf modificati
/etc/rc.d
/dev/md*
```

dal sistema che uso in modo da personalizzarlo per il kernel e la macchina su cui verrà fatto girare.

Questo fa sì che il sistema di boot/ripristino sia lo stesso sistema installato sul dispositivo raid montato come root, solo un pochino "dimagrito" ma con le revisioni delle librerie ecc. sempre aggiornate.

4.4 Istruzioni PASSO dopo PASSO

Dalla directory home di root (/root):

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

Crea un punto di mount su cui lavorare

```
mkdir mnt
mkdir mnt2
```

Crea un file abbastanza grande da contenere il file system. Sarà un po' più grande del file system di recupero. Ho scelto 24 mega poiché 16 mega non sono sufficienti

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

associa il file con un dispositivo di loop e genera sul file un file system ext2

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

4.5 Installa la distribuzione - Specifico per Slackware

4.6 (...salta le operazioni specifiche per Slackware) e vai alla prossima sezione

Ora che un file system vuoto è stato creato e montato, esegui "setup".

```
Specifica                      /root/raidboot/mnt
```

come **'target'**. Come source puoi scegliere tutto quello che usi normalmente. Scegli i pacchetti che vuoi installare e procedi, ma **NON** eseguire la configurazione.

Scegli il modo 'EXPERT'

Io scelgo 'A', 'AP' e 'N', installando solo il minimo indispensabile per far funzionare il sistema più un editor che mi è familiare (vi, jed, joe) e che è ragionevolmente compatto.

```
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 mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
```

Dalla serie di pacchetti AP scelgo solo 'JOE', un editor che mi piace, e 'MC', un piccolo ed utile file manager. Puoi scegliere i programmi di cui avrai bisogno sul tuo sistema.

```
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 mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
```

Dal pacchetto 'N' ho scelto solo TCPIP. Nonostante non sia veramente necessario si rivela molto utile e permette l'accesso alla rete mentre si cerca di aggiustare qualche guaio o di effettuare un aggiornamento con l'array raid non montato. TCPIP contiene anche 'biff', che è usato da alcune applicazioni di 'A'. Anche se non installi 'N' probabilmente sarebbe il caso di installare il pacchetto biff ugualmente.

```
lqqqq SELECTING PACKAGES FROM SERIES N (NETWORK/NEWS/MAIL/UUCP) qqqqqk
x lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk 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 mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
```

Completata l'installazione rispondi no a tutto il resto (cioè rispondi no a tutte le richieste di configurazione) ed esci dallo script.

4.6 Installa linux pthreads

Adesso devi installare la libreria 'linuxthreads-0.71'. Ho incluso la patch che segue per il Makefile di linuxthreads invece di spiegare i dettagli dell'installazione. Salva il Makefile originale, applica la patch e poi:

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

```
-------------------diff Makefile.old  Makefile.raid-----------------
2a3,13
> # Se stai compilando "linuxthreads" per installarlo su un punto di mount
> # che non è la partizione di root, ridefinisci 'BUILDIR' in modo che sia
> # uguale al punto di mount che vuoi usare come directory "root"
> # Potresti doverlo fare se stai costruendo un 'ram disk iniziale' come
> # quello usato con i dispositivi root raid capaci di eseguire il boot.
> # RICHIEDE la versione 1.9.5 o superiori di ldconfig
> # per controllare esegui ldconfig -v
> #
> 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
```

4.7 Installazione degli strumenti Raid

Il passo successivo consiste nell'installazione degli strumenti raid. raidtools-0.42

Devi eseguire lo script "configure" facendolo puntare al Makefile nella directory con i sorgenti per i file ramdisk

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

La parte del Makefile riguardante l'installazione non funziona benissimo, quindi fai quello che segue per sistemare le cose. Questo problema verrò corretto nelle versioni future, così non sarà necessario effettuare nuovi link.

Correggere l'errore di make install

I link che sono specificati nel Makefile alla sezione 'LINKS' devono essere rimossi e rifatti correttamente in modo che tutto funzioni nel modo giusto.

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

4.8 Rimuovere directory e file non necessari dal nuovo filesystem.

Cancella le seguenti directory dal filesystem (ATTENZIONE NON CANCELLARLE DAL SISTEMA ATTUALMENTE IN ESECUZIONE) è un errore facile da commettere, indovina come me ne sono accorto!!!

```
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
```

4.9 Creare /dev/md$x$

L'ultimo passo consiste semplicemente nel copiare i dispositivi /dev/md* dal filesystem corrente sul filesystem di rescue. Potresti anche crearli con mknode.

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

4.10 Creare un filesystem utilizzabile per *initrd*

Adesso hai un filesystem pronto per essere personalizzato. Una volta personalizzato, questo filesystem può essere usato per recuperare il sistema nel caso in cui i dispositivi raid avessero problemi e ci fosse bisogno degli strumenti raid per rimediare. Sarà anche usato per eseguire il boot e montare come root il dispositivo raid usando il file linuxrc che verrà discusso di seguito.

Copia il filesystem su un dispositivo più piccolo per il file initrd, 16 mega dovrebbero essere sufficienti.

Crea il filesystem più piccolo e montalo

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

associa il file con un dispositivo di loop e genera un filesystem ext2 sul file

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

Copia il filesystem 'build' su 'bare.fs'

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

Salva il sistema 'bare.fs' prima di personalizzarlo in modo da rendere più semplici eventuali aggiornamenti futuri. Il filesystem 'build' non serve più e può essere cancellato.

```
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
```

4.10.1 Creare il filesystem di BOOT/RESCUE *initrd*

Adesso copia le cose dipendenti dal sistema che corrispondono al kernel dal sistema usato per lo sviluppo, in alternativa puoi modificare manualmente i file nel sistema di recupero in modo che corrispondano al sistema che stai costruendo.

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

Assicurati che nessun file `*~`, core o di log compaia nella tua directory etc. I 2 comandi che seguono danno dei messaggi di errore: ignorali.

```
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 <--- la tua versione 2.x.x
```

Modifica i file seguenti per adattarli al tuo sistema di recupero.

```
        cd mnt

Non-network
        etc/fstab
        etc/mdtab       dovrebbe funzionare correttamente
Network
        etc/hosts
        etc/resolv.conf
        etc/hosts.equiv         e file correlati
        etc/rc.d/rc.inet1       correggi ip#, mask, gateway, ecc...
        etc/rc.d/rc.S           rimuovi tutta la sezione riguardante lo
                                stato del filesystem,
                a partire da:
                        # Test to see if the root partition isread-only
                fino a, ma senza includerlo:
                        # remove /etc/mtab* so that mount will .....
                                Questo evita il fastidioso avvertimento
                                che il ramdisk è montato in lettura/scrittura.
        etc/rc.d/rc.xxxxx       altri se richiesto, vedi oltre
        root/.rhosts            se presente
        home/xxxx/xxxx          altri se richiesto

ATTENZIONE:     La procedura appena illustrata sposta i tuoi file
                password e shadow sul disco di recupero!!!!!

ATTENZIONE:     Forse non vuoi che questo accada per ragioni di sicurezza.
```

Crea tutte le directory che sono necessarie per montare /dev/disk... Questi sono i punti di mount per eseguire il boot del sistema (partizione di boot e partizione di boot di backup). Il mio sistema esegue il boot da dos usando **loadlin**, comunque anche partizioni linux e lilo funzioneranno bene. Il mio sistema usa:

```
        cd /root/raidboot/mnt           <--- initrd root
        mkdir dosa                      punto di mount della partizione dos
        mkdir dosb                      punto di mount del mirror dos
```

Il filesystem di recupero è completo!

Noterai, esaminando i file nel filesystem di recupero, che ci sono ancora molti file che potrebbero essere cancellati. Non l'ho fatto perché complicherebbe troppo la procedura e la maggior parte dei sistemi raid hanno memoria e spazio su disco a sufficienza. Se vuoi avere un filesystem ancora più piccolo, fallo!

4.11 Eseguire il boot del dispositivo RAID - linuxrc

Per far sì che il disco di recupero esegua il boot del dispositivo raid devi solo copiare il file di script:

linuxrc

nella directory root del dispositivo.

La teoria del funzionamento di questo file è discussa in 14 (Appendice G, teoria del funzionamento di linuxrc).

Un linuxrc (funzionante) molto semplice e più facile da capire si trova in 11 (Appendice D), *script linuxrc e di shutdown obsoleti*. Copia il testo seguente in un file **linuxrc** e salvalo nella tua area di sviluppo.

```
 -------------------- linuxrc ----------------------
#!/bin/sh
# ver 1.13 3-6-98
#
################# BEGIN 'linuxrc' ##################
#            DEFINIZIONE DELLE FUNZIONI            #
####################################################
# Definisce la funzione 'Fault' nel caso in cui
# qualcosa vada storto durante l'esecuzione di 'linuxrc'
#
FaultExit () {
# corregge fstab per mostrare '/dev/ram0/ per il sistema di recupero
    /bin/cat /etc/fstab | {
    while read Line
    do
        if [ -z "$( echo ${Line} | /usr/bin/grep md0 )" ]; then
            echo ${Line}
        else
            echo "/dev/ram0 / ext2 defaults 1 1"
        fi
    done
    } > /etc/tmp.$$
    /bin/mv /etc/tmp.$$ /etc/fstab
#       point root at /dev/ram0 (the rescue system)
        echo 0x100>/proc/sys/kernel/real-root-dev
        /bin/umount /proc
        exit
}

# Definisce una procedura 'Warning' per stampare un banner sul terminale di boot
#
Warning () {
    echo '*********************************'
    echo -e " $*"
    echo '*********************************'
}

# Definisce 'SplitKernelArg' che aiuta ad estrarre gli argomenti del kernel
# riguardanti 'Raid'
SplitKernelArg () { eval $1='$( IFS=,; echo $2)' }

# Definisce 'SplitConfArgs' che aiuta ad estrarre gli argomenti di
# configurazione del sistema
SplitConfArgs () {
    RaidBootType=$1
    RaidBootDevice=$2
    RaidConfigPath=$3
}
```

```
########################################################
################### MAIN linuxrc ######################
########################################################
# monta il filesystem proc
/bin/mount /proc

# Prende dalla linea di comando la partizione di boot e la posizione
# della configurazione
CMDLINE=`/bin/cat /proc/cmdline`
for Parameter in $CMDLINE; do
    Parameter=$( IFS='='; echo ${Parameter} )
    case $Parameter in
        Raid*) SplitKernelArg $Parameter;;
    esac
done

# controlla 'required raid boot'
if [ -z "${Raid_Conf}" ]; then
    Warning Kernel command line \'Raid_Conf\' missing
    FaultExit
fi
SplitConfArgs $Raid_Conf

# monta temporaneamente la partizione di boot
/bin/mount -t ${RaidBootType} ${RaidBootDevice} /mnt

# prende i file della directory etc dal sistema raid primario
pushd /etc

# questo verrà decompresso nella directory /etc (vedi rc.6)
if [ ! -f /mnt/${RaidConfigPath}/raidboot.etc ]; then
# cattive notizie, questo file dovrebbe esserci
    Warning required file \'raidboot.etc\' \
    missing from ${RaidBootDevice}/${RaidConfigPath} \\n \
    \\tUsing rescue system defaults
else
    /bin/tar -xf /mnt/${RaidConfigPath}/raidboot.etc
fi
# cerca il 'vero' dispositivo raidboot per questo boot
# il percorso dello stato e il nome di raidX.conf
if [ ! -f /mnt/${RaidConfigPath}/raidboot.cfg ]; then
# cattive notizie, questo file dovrebbe esserci
    Warning required file 'raidboot.cfg' \
    missing from ${RaidBootDevice}/${RaidConfigPath}\\n \
    \\tUsing rescue system defaults
# Prendi il primo nome di file raidX.conf in $RArg1
    RaidBootDevs=$RaidBootDevice
    RaidStatusPath=$RaidConfigPath
    for RaidConfigEtc in $( ls raid*.conf )
    do break; done
```

```
else
    {
    read RaidBootDevs
    read RaidStatusPath
    read RaidConfigEtc
    } < /mnt/${RaidConfigPath}/raidboot.cfg

fi
popd
/bin/umount /mnt

# Nel caso in cui non trovi il file di stato raid setta un flag
#
RAIDOWN="raidboot.ro not found"
RAIDREF="raidgood.ref not found"
echo "Reading md0 shutdown status."

# cerca lo stato di shutdown raid
for Device in ${RaidBootDevs}
do
# i tipi di questi filesystem dovrebbero essere in 'fstab'
#  le partizioni devono essere montate per avere uno shutdown raid pulito
    /bin/mount ${Device} /mnt
    if [ -f /mnt/${RaidStatusPath}/raidboot.ro ]; then
        RAIDOWN=`/bin/cat /mnt/${RaidStatusPath}/raidboot.ro`
        RAIDREF=`/bin/cat /mnt/${RaidStatusPath}/raidgood.ref`
        /bin/umount /mnt
        break
    fi
    /bin/umount /mnt
done
# Controlla che lo shutdown sia avvenuto in modo pulito
if [ "${RAIDOWN}" != "${RAIDREF}" ]; then
    Warning shutdown ERROR ${RAIDOWN}
    FaultExit
fi

# L'array raid è pulito, rimuovi i file di stato di shutdown
for Device in ${RaidBootDevs}
do
    /bin/mount ${Device} /mnt
    /bin/rm -f /mnt/${RaidStatusPath}/raidboot.ro
    /bin/umount /mnt
done

# Scrivi un superblock pulito su tutti i dispositivi raid

echo "write clean superblocks"
/sbin/mkraid -f --only-superblock /etc/${RaidConfigEtc}
```

```
# Attiva gli array raid
if [ -z "$Raid_ALT" ]; then
    /sbin/mdadd -ar
else
    /sbin/mdadd $Raid_ALT
fi

#  Se ci sono errori - ESCI e lascia funzionante il sistema di recupero
if [ $? -ne 0 ]; then
   Warning some RAID device has errors
   FaultExit
fi

# Tutto va bene, fai montare /dev/md0 al kernel
# e dì al kernel di assumere /dev/md0 come il dispsitivo root
# Il valore 0x900 è il numero di dispositivo calcolato come:
#  256*numero_major + numero_minor
echo "/dev/md0 mounted on root"
echo 0x900>/proc/sys/kernel/real-root-dev
# umount /proc to deallocate initrd device ram space
/bin/umount /proc
exit
#------------------ end linuxrc ----------------------
```

Aggiungi 'linuxrc' al device di boot initrd

```
cd /root/raidboot
chmod 777 linuxrc
cp -p linuxrc mnt
```

4.12 Modificare gli script rc per lo shutdown

Per completare l'installazione, modifica gli script rc in modo da salvare lo stato degli md sul vero dispositivo di root quando si verifica lo shutdown.

```
Nella slackware questi script sono rc.0 -> rc.6
Nella debian 'bo' sono sia 'halt' che 'reboot'

Se implementi questo in un'altra distribuzione, per favore invia per
e-mail le istruzioni e file di esempio in modo che possano essere inclusi qui.
```

Ho modificato leggermente il metodo per lo stop raid di Bohumil Chalupa. La sua soluzione originale è presentata in 8 (Appendice A).

Poiché non ci sono partizioni linux libere sul sistema di produzione tranne **md0**, le partizioni di boot sono usate per memorizzare lo stato **raidOK readonly**. Ho scelto di scrivere su ognuna copia della partizione di boot un file contenente lo stato dell'array md allo shutdown il cui significato è: il dispositivo md è stato rimontato in sola lettura. Questo fa sì che il sistema possa continuare a funzionare anche nel caso che uno dei drive muoia.

Lo script di shutdown è stato modificato in modo da richiamare 13 (rc.raidown), che salva le informazioni necessarie per effettuare effettuare il reboot e montare il dispositivo raid. Alcuni esempi di script di shutdown per varie distribuzioni di linux si trovano in 9 (Appendice B).

Per catturare lo stato allo shutdown dell'array raid inserisci una chiamata a 13 (rc.raidown) dopo tutte le istruzioni **case** (se presenti) ma prima dell'inizio dello shutdown vero e proprio (kill, salvataggi di stato ecc.) e prima che i filesystem siano smontati.

```
########### Salva informazioni di boot raid e stato #############
#
  if [ -x /etc/rc.d/rc.raidown ]; then
    /etc/rc.d/rc.raidown
  fi
################# fine del boot raid ########################
```

Una volta che tutti i file system sono smontati (il file system di root non verrà smontato) ma prima di qualsiasi controllo relativo allo stato della mancanza di corrente aggiungi:

```
############### per array raid ########################
# Ferma tutti gli array raid conosciuti (salvo root che non si ferma)
  if [ -x /sbin/mdstop ]; then
    echo "Stopping raid"
    /sbin/mdstop -a
  fi
##########################################################
```

Questo fermerà in modo pulito tutti i dispositivi di raid tranne il dispositivo di root. Il suo stato è passato in **raidstat.ro** al prossimo boot.

Copia il file rc sul tuo nuovo array raid, sul file system di recupero che è ancora montato come **/root/raidboot/mnt** e sul sistema usato per lo sviluppo se si trova sulla stessa macchina.

Modifica **etc/fstab** sul filesystem di recupero in modo opportuno ed assicurati che **mdtab** sul filesystem di recupero sia corretto.

Adesso copia il disco di recupero sulla tua partizione dos, e tutto dovrebbe essere pronto per eseguire il boot con il dispositivo raid come root.

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

Copia rescue.gz sulle tue partizioni di boot.

Tutto ciò che rimane è creare il file di configurazione **raidboot.conf** e controllare il nuovo file system eseguendo il reboot.

4.13 Configurare RAIDBOOT - raidboot.conf

I commenti che seguono il file di configurazione di esempio spiegano ciascuna delle tre linee. Questo file di esempio è relativo ad un array scsi raid5 con 4 drive con partizioni di boot duplicate sui drive sda1 e sdb1. Sostituisci invece i parametri relativi ai tuoi file system.

```
/dev/sda1 /dev/sdb1
linux
raid5.conf
```

```
# i commenti possono essere messi solo 'dopo' le tre
# linee di configurazione.
#
# Questo è 'raidboot.conf'
#
# linea uno, le partizioni contenenti il sistema raid-rescue 'initrd'
#       Non è necessario eseguire il boot da queste partizioni; comunque,
#       poiché il sistema di recupero non entra su un solo floppy, è
#       necessario sapere quali partizioni devono essere usate per caricare
#       il sistema di recupero
#
# linea due, il percorso del file di configurazione raidboot
#       La posizione dello stato di shutdown, ecc... a tempo di boot
#       NON include le informazioni sul punto di mount, solo il 'percorso'
#       /punto_di_mount/'percorso'
#
# linea tre, il nome del file di configurazione raid
#       Il file di configurazione raid corrente, ad esempio
#       raid1.conf, raid5.conf
```

4.14 Le variabili del kernel per RESCUE e RAID

Ci sono due variabili del kernel per il sistema di recupero e RAID, solo la prima deve essere specificata.

- Raid_Conf=msdos,/dev/sda1,raidboot

  Questa variabile punta al dispositivo raid di boot e al file di configurazione. Per il floppy di boot di recupero puoi specificarla alla linea di comando del kernel o nei file di boot di loadlin o di lilo.

  formato: `'filesystem-type,device,path-to-config-from-mountpoint'`

- Raid_ALT=-r,-p5,/dev/md0,/dev/sda3 /dev/sdb3 /dev/sdc3 /dev/sdd3

  Parametri alternativi per mdadd necessari quando si effettua il boot con array raid non ridondanti. Questi sono i parametri da linea di comando, separati da virgole, per **mdadd**. A meno che non siano necessari per avviare un array malfunzionante/non ridondante, COMMENTATELI O SPECIFICATELI CON UN 'NULL'.

  i.e. Raid_ALT=

Entrambi questi parametri possono essere specificati nel file di boot di lilo o loadlin o nella linea di comando del kernel di loadlin. Bisogna fare attenzione, comunque, a non eccedere la lunghezza massima di linea nel caso si usi la linea di comando (128 caratteri).

Eseguendo il boot con **lilo**, i parametri sono inclusi nel file di configurazione di lilo nella forma:

```
append="Raid_Conf=msdos,/dev/sda1,raidboot"
append="Raid_ALT=-r,-p5,/dev/md0,/dev/sda3 /dev/sdb3 /dev/sdc3 /dev/sdd3"
```

Vedi **man lilo.conf** per informazioni più dettagliate.

Poiché ho dell'hardware che richiede utility di configurazione DOS, ho una piccola partizione dos nel sistema. Quindi, uso loadlin per eseguire il boot del sistema raid5 dalla partizione dos con un mirror (una copia) del disco. Un metodo identico è usato per il sistema raid1. L'esempio sotto usa loadlin, ma la procedura è molto simile per lilo.

Il mio sistema dos contiene fra le utility un piccolo editor, così posso modificare i parametri di boot di loadlin se necessario, permettendomi di effettuare il reboot del sistema linux sul mio disco di swap mentre faccio dei test.

Il sistema dos contiene questo albero di directory per linux

```
c:\raidboot.bat
c:\raidboot\loadlin.exe
c:\raidboot\zimage
c:\raidboot\rescue.gz
c:\raidboot\raidboot.cfg
c:\raidboot\raidboot.etc
c:\raidboot\raidgood.ref
c:\raidboot\raidstat.ro (solo allo shutdown)
```

linux.bat contiene:

```
---------------------- linux.bat ---------------------------
echo "Start the LOADLIN process:"
c:\raidboot\loadlin @c:\raidboot\boot.par
-------------------- end linux.bat -------------------------
```

boot.par contiene:

```
# loadlin boot parameter file
#
# version 1.02 3-6-98

# immagine del kernel di linux
c:\linux\zimage

# dispositivo target di root
root=/dev/md0
#root=/dev/ram0
#root=/dev/sdc5

# monta il dispositivo di root come 'ro'
ro

# dimensione del ram disk
ramdisk_size=16384

# nome del file initrd
initrd=c:\raidboot\rescue.gz
#noinitrd

# la memoria finisce qui
mem=131072k
```

```
        # punta al dispositivo raid di boot, al file di configurazione
        # per il boot da floppy di recupero, puoi specificarlo
        # sulla linea di comando invece di qui
        # formato 'tipo-di-filesystem,dispositivo,percorso-di-config-frm_mntpnt'
        Raid_Conf=msdos,/dev/sda1,raidboot

        # Parametri alternativi di mdadd
        # necessari per il boot con raid non ridondante
        # altrimenti COMMENTARE O SPECIFICARE 'NULL'
        #Raid_ALT=-r,-p5,/dev/md0,/dev/sda3 /dev/sdb3 /dev/sdc3 /dev/sdd3

        # dispositivi ethernet
        ether=10,0x300,eth0

***** >> NOTA!! la sola differenza fra forzare l'esecuzione del sistema
        di recupero e montare il dispositivo raid è il parametro di loadlin

                root=/dev/ram0          per il sistema di recupero
                root=/dev/md0           per RAID

                Con root=/dev/ram0 il dispositivo RAID non sarà montato
                e il sistema di recupero partirà incondizionatamente.
```

Se l'array RAID fallisce, il sistema di recupero viene lasciato montato e funzionante.

5 Configurare il sistema RAID.

5.1 Specifiche di sistema. Sono stati configurati due sistemi con schede madri identiche.

```
                                   Raid-1          Raid-5
Motherboard:    Iwill P55TU      dual ide        adaptec scsi
Processore:     Intel P200
Dischi:                          2ea  7 gig      4 ea Segate 4.2 gig
                                 Maxtors         wide scsii
```

I drive sono indicati da linux con i nomi da 'sda' a 'sdd' sul sistema raid5 e da 'hda' a 'hdc' sul sistema raid1.

5.2 Partitionare i dischi rigidi.

Poiché testare un grosso array raid di root montabile è difficile a causa del problema di reboot causato da ckraid, ho ripartizionato il mio spazio di swap in modo da includere una partizione RAID più piccola a scopo di test, sda6, sdb6, sdc6, sdd6 e una piccola coppia di partizioni, root e /usr/src per sviluppare e testare il kernel raid e gli strumenti. Forse ti può essere utile.

```
        <bf/SISTEMA DI SVILUPPO - RAID5/
   Dispos.      Sistema         Dim.    Scopo

  /dev/sda1     dos boot        16 meg  partizione di boot
```

```
* /dev/sda2     extended        130 meg (vedi sotto)
  /dev/sda3     linux native    4 gig   raid5-1 primario
----------------------sda2-----------------------------
* /dev/sda5     linux swap      113 meg spazio di swap
* /dev/sda6     linux native    16 meg  test raid5-1
========================================================
  /dev/sdb1     dos boot        16 meg  copia partizione di boot
* /dev/sdb2     extended        130 meg (vedi sotto)
  /dev/sdb3     linux native    4 gig   raid5-2 primario
----------------------sdb2-----------------------------
* /dev/sdb5     linux swap      113 meg spazio di swap
* /dev/sdb6     linux native    16 meg  test raid5-2
========================================================
* /dev/sdc2     extended        146 meg (vedi sotto)
  /dev/sdc3     linux native    4 gig   raid5-3 primario
----------------------sdc2-----------------------------
* /dev/sdc5     linux swap      130 meg partiz. di root di sviluppo
* /dev/sdc6     linux native    16 meg  test raid5-3
========================================================
* /dev/sdd2     extended        146 meg (vedi sotto)
  /dev/sdd3     linux native    4 gig   raid5-4 primario
----------------------sdd2-----------------------------
* /dev/sdd5     linux swap      130 meg /usr/src di sviluppo
* /dev/sdd6     linux native    16 meg  test raid5-4


        <bf/SISTEMA DI SVILUPPO - RAID1/
   Dispos.      Sistema         Dim.    Scopo

  /dev/hda1     dos             16meg   partizione di boot
* /dev/hda2     extended        126m    (vedi sotto)
  /dev/hda3     linux           126m    partizione root di sviluppo
  /dev/hda4     linux           6+gig   raid1-1
----------------------hda2-----------------------------
* /dev/hda5     linux            26m    test raid1-1
* /dev/hda6     linux swap      100m
========================================================

  /dev/hdc1     è semplicemente una copia esatta di hda1 così che
                la partizione può essere resa attiva se hda fallisce
* /dev/hdc2     extended        126m    (vedi sotto)
  /dev/hdc3     linux           126m    /usr/src di sviluppo
  /dev/hdc4     linux           6+gig   raid1-2
----------------------hdc2-----------------------------
* /dev/hdc5     linux            26m    test raid1-2
* /dev/hdc6     linux swap      100m
```

Le partizioni sdx2 e hdx3 sono state definite 'swap' dopo lo sviluppo di questa utility. Avrei potuto farla su un'altra macchina, comunque, le librerie e i kernel sono vecchi di un anno o più sulle mie altre macchine linux, e ho preferito costruirla sulla macchina target.

Lo schema di partizionamento è stato scelto in modo tale che, nel caso che uno qualsiasi dei dischi fallisca catastroficamente, il sistema continui a funzionare e sia ancora possibile eseguire il boot con il minimo sforzo e senza perdita di dati.

- Se un qualsiasi drive fallisce, la procedura di boot terminerà e verrà fatto partire il sistema di recupero. L'esame dei messaggi su schermo o /dos$x$/raidboot/raidstat.ro informerà l'operatore sullo stato dell'array.
- Se sda1 (raid5) o hda1 (raid1) fallisce, la partizione di boot di backup del dos deve essere resa 'attiva' e il bios deve riconoscere la nuova partizione come dispositivo di boot o deve essere spostato fisicamente alla posizione $x$da.

  In alternativa, il sistema potrebbe essere fatto partire con un floppy disk usando l'immagine initrd sul boot drive di backup che rimane. Il sistema raid può poi essere reso attivo ancora con:

  ```
  "/sbin/mkraid /etc/raid<it/x/>.conf -f --only-superblock"
  ```

  per ricostruire i superblocchi rimanenti.
- Fatto questo,

  ```
  mdadd -ar
  ```

- Esamina lo stato dell'array per verificare che tutto sia OK poi rimpiazza il riferimento giusto all'array con lo stato corrente finché il disco mal funzionante può essere riparato o sostituito.

  ```
  cat /proc/mdstat | grep md0 > /dosx/raidboot/raidgood.ref

  shutdown -r now
  ```

  per fare un reboot pulito, e il sistema è ancora in funzione.

6 Costruire il file system RAID.

Questa è la descrizione dei miei sistemi RAID di cui parlo nelle specifiche di sistema. Il tuo sistema potrebbe avere un'architettura RAID diversa, perciò apporta le modifiche opportune. Leggi anche le pagine di manuale e QuickStart.RAID che è incluso nei raidtools-0.42.

6.1 /etc/raid5.conf

```
# raid-5: configurazione
raiddev                 /dev/md0
raid-level              5
nr-raid-disks           4
chunk-size              32

# Parity placement algorithm
parity-algorithm        left-symmetric

# Spare disks for hot reconstruction
#nr-spare-disks         0
```

```
device              /dev/sda3
raid-disk           0

device              /dev/sdb3
raid-disk           1

device              /dev/sdc3
raid-disk           2

device              /dev/sdd3
raid-disk           3
```

6.2 /etc/raid1.conf

```
# raid-1 configurazione
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
```

6.3 Procedure per la costruzione passo a passo di un file system RAID.

Per il mio sistema RAID5 ho fatto un'installazione completa di:

```
Slackware-3.4   qualsiasi distribuzione recente dovrebbe andare bene
linuxthreads-0.71
raidtools-0.42
linux-2.0.33 con la patch raid145 e la patch di Gadi
```

Crea e formatta il dispositivo raid.

```
mkraid /etc/raid5.conf
mdcreate raid5 /dev/md0 /dev/sda3 /dev/sdb3 /dev/sdc3 /dev/sdd3
mdadd -ar
mke2fs /dev/md0
mkdir /md
mount -t ext2 /dev/md0 /md
```

Crea i file di riferimento che verranno usati da reboot, ci potrebbero essere delle differenze sul tuo sistema.

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

Usa Slackware-3.4 o un'altra distribuzione per costruire il tuo SO

```
setup
```

Specifica '/md' come target, e il sorgente che usi normalmente. Scegli ed installa i diskset che ti interessano tranne il kernel. Configura il sistema, ma salta la sezione su lilo e sul boot del kernel. Esci da setup.

Installa 'pthreads'

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

modifica il Makefile e specifica

```
BUILDIR=/md

make
make install
```

Installa 'raidtools'

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

correggi l'errore dei raidtools che si verifica al make install

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

Crea /dev/mdx

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

Aggiungi la configurazione del sistema dal sistema corrente (ignora gli errori).

```
cp -dp /etc/* mnt/etc
cp -dp /etc/rc.d/* mnt/etc/rc.d         (include il nuovo rc.6)
mkdir  mnt/lib/modules
cp -a  /lib/modules/2.x.x mnt/lib/modules <--- il 2.x.x corrente
```

Adatta i file seguenti al tuo file system

```
cd /md
```

```
Non-network
        etc/fstab       inserisci i dispositivi di root e raid corretti.
        etc/mdtab       dovrebbe funzionare
Network
        etc/hosts
        etc/resolv.conf
        etc/hosts.equiv         e file correlati
        etc/rc.d/rc.inet1       correggi il numero ip#, mask, gateway, ecc...
```

```
etc/rc.d/rc.S           togli tutta la sezione sullo
                        stato del file system
        da:
                # Test to see if the root partition isread-only
        a, ma non incluso:
                # remove /etc/mtab* so that mount will .....
                        Questo evita il fastidioso avvertimento
                        sul fatto che il ramdisk è montato rw.
etc/rc.d/rc.xxxxx       altri se necessario
root/.rhosts            se presente
home/xxxx/xxxx          altri se necessario
```

```
ATTENZIONE:    La procedura sopra muove i tuoi file password e shadow
               sul nuovo file system!!!!!

ATTENZIONE:    Puoi non volerlo fare per ragioni di sicurezza.
```

Crea le directory necessarie a montare /dev/disk... Queste sono specifiche del sistema. Nel mio ho bisogno di:

```
cd /md          <--- nuovo root del file system
mkdir dosa                     punto di mount della partizione dos
mkdir dosb                     punto di mount del mirror dos
```

Il nuovo file system è completo. Assicurati di salvare lo stato di riferimento di md sul 'vero' dispositivo di root e sei pronto ad effettuare il boot.

monta le partizioni dos su dosa e dosb

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

mdstop /dev/md0
```

7 Un'ultima cosa.

Ricorda che un esperto è una persona che ne sa almeno l'1% più di te riguardo a un particolare soggetto. Tienilo in mente se decidi di chiedermi aiuto per e-mail. Proverò, ma ho fatto queste cose solo una volta per raid1 e una volta per raid5!

Michael Robinton *Michael@bzs.org*

8 Appendice A. - Lo shutdown per md0 di Bohumil Chalupa

Il post di Bohumil Chalupa sulla lista linux raid a proposito della soluzione del problema di mdstop per raid1 e 5. La sua soluzione non tiene conto della possibilità che il dispositivo raid venga corrotto allo shutdown. Così ho aggiunto un semplice confronto di stato effettuato al boot con un buono stato di riferimento. Questo permette all'operatore di intervenire se un disco nell'array ha dei problemi. Una descrizione di tutto ciò si trova nella parte principale di questo stesso documento.

```
> From: Bohumil Chalupa <bochal@apollo.karlov.mff.cuni.cz>
> Sono riuscito ad eseguire il boot di initrd e ad usare linuxrc per far
> partire l'array RAID1, per poi far diventare root /dev/md0.
>
> Non conosco, però, un modo per _fermare_ in modo pulito l'array.

Beh, dovrò rispondermi da solo :-)

> Date: Mon, 29 Dec 1997 02:21:38 -0600 (CST)
> From: Edward Welbon <welbon@bga.com>
> Subject: Re: smontare il dispositivo raid di root
>
> Per dispositivi md diversi da raid0, lo stato che deve essere salvato
> è noto probabilmente solo una volta che tutte le operazioni di scrittura
> sono state completate. Un tale stato non può essere naturalmente salvato
> sulla directory root una volta montata in sola lettura. In tal caso,
> dovresti poter montare un filesystem scrivibile "X" sul root in sola
> lettura e poter scrivere su "X" (ricordo di aver fatto una cosa del
> genere durante le operazioni di "recupero", ma non come procedura
> automatica)
>
> Il filesystem "X" sarà presumibilmente un dispositivo di boot da cui
> il raid (durante l'esecuzione di linuxrc via initrd) prenderà il suo
> stato iniziale. Fortunatamente raid0 non deve scrivere nessuno stato
> (anche se sarebbe piacevole poter scrivere i checksum su mdtab dopo
> un mdstop). Giocherò un po' con queste cose che non sembrano troppo
> difficili, anche se il "diavolo" è sempre nei "dettagli".

Già, è così.
Ho già avuto questa idea, ma non ho avuto tempo per provare. L'ho
fatto ieri, e funziona.

Con il mio RAID1 (mirror), non salvo nessun dato di checksum o
superblock raid.  Salvo solo un'informazione sulla "vera" partizione
di boot, che il volume di root md è stato rimontato in sola lettura
durante lo shutdown. Poi, durante il boot, lo script linuxrc esegue
mkraid --only-superblock se trova questa informazione, altrimenti
esegue ckraid.
Questo significa che le informazioni del superblock raid non vengono
aggiornate durante lo shutdown; sono aggiornate a tempo di boot.  Non
è una cosa molto pulita, temo, :-( ma funziona.

Uso Slackware e initrd.md di Edward Welbon per eseguire il boot del
dispositivo raid di root.
Per quanto mi ricordo ora, gli unici file modificati sono mkdisk e
linuxrc, e lo script di shutdown /etc/rc.d/rc.6 E lilo.conf,
naturalmente.

Seguono le parti più importanti.
```

```
Bohumil Chalupa

--------------- segue mio.linuxrc  -----------------
#!/bin/sh
# ci serve /proc
/bin/mount /proc
# avvia il dispositivo md0. Lascia fare il resto agli script /etc/rc.d
# Si dovrebbe fare il meno possibile qui
# ----------------------------------------
# shutdown test e ricreazione di root raid1
# /start deve essere creato sull'immagine rd in my.mkdisk
echo "preparing md0: mounting /start"
/bin/mount /dev/sda2 /start -t ext2
echo "reading saved md0 state from /start"
if [ -f /start/root.raid.ok ]; then
 echo "raid ok, modyfying superblock"
 rm /start/root.raid.ok
 /sbin/mkraid /etc/raid1.conf -f --only-superblock
else
 echo "raid not clean, runing ckraid --fix"
 /sbin/ckraid --fix /etc/raid1.conf
fi
echo "unmounting /start"
/bin/umount /start
# -----------------------------------------
#
echo "adding md0 for root file system"
/sbin/mdadd /dev/md0 /dev/sda1 /dev/sdb1
echo "starting md0"
/sbin/mdrun -p1 /dev/md0
# dì al kernel che vogliamo far diventare /dev/md0 il dispositivo di root
# arriviamo al valore 0x900 come 256*numero_di_dispositivo_major + numero_minor
echo "setting real-root-dev"
/bin/echo 0x900>/proc/sys/kernel/real-root-dev
#  smonta /proc in modo da poter deallocare il ram disk
echo "unmounting /proc"
/bin/umount /proc
/bin/echo "We are hopefully ready to mount /dev/md0 (major 9, minor 0) as
root"
exit
--------------- fine di mio.linuxrc -----------------------------------


----------- segue un estratto da /etc/rc.d/rc.6  -----------------
  # Disattiva lo swap poi smonta i file system locali.
  echo "Turning off swap."
  swapoff -a
  echo "Unmounting local file systems."
  umount -a -tnonfs
  # Non rimontare i volumi root UMSDOS:
```

```
  if [ ! "`mount | head -1 | cut -d ' ' -f 5`" = "umsdos" ]; then
    mount -n -o remount,ro /
  fi

  # Salva lo stato raid
  echo "Saving RAID state"
  /bin/mount -n /dev/sda2 /start -t ext2
  touch /start/root.raid.ok
  /bin/umount -n /start

-------------- fine dell'estratto da rc.6 ------------------------


------------------ segue una parte di mio.mkdisk ----------------------
#
#  adesso abbiamo un filesystem pronto da riempire, e dobbiamo metterci
#  alcune directory importanti. Ho avuto tantissimi guai finché non ho
#  creato un mtab intatto.  Nel mio caso, è conveniente sovrascrivere
#  /etc/mdtab, in questo modo posso attivare md con un semplice
#  "/sbin/mdadd -ar" in 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: servono mkraid e ckraid
cp -a $ROOT/usr/sbin/mkraid $ROOT/usr/sbin/ckraid $MOUNTPNT/sbin
2>>cp.stderr 1>>cp.stdout
# ---------------------------------------------------------------------
#  sembra che init non funzioni se non ha utmp. Forse ci si può lavorare
#  molto. Non dirò qual è il vero problema 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
#
#  poiché voglio cambiare il punto di mount, ho bisogno di questo
#  nonostante avessi potuto fare un "mkdir /proc" in linuxrc.
#
mkdir $MOUNTPNT/proc
chmod 555 $MOUNTPNT/proc
#
```

```
#  ----------------------------------------------------
#  monteremo il vero dispositivo di boot in /start temporaneamente
#  per verificare lo stato del root raid salvato durante lo shutdown
#
mkdir $MOUNTPNT/start
#  -----------------------------------------------------
#
#  ci serve linuxrc  (dopo tutto, è il punto focale di questo esercizio).
#
if [ -x ./my.linuxrc ]; then
  cp -a ./my.linuxrc $MOUNTPNT/linuxrc
  chmod 777 $MOUNTPNT/linuxrc
else
   ln -s /bin/sh $MOUNTPNT/linuxrc
fi
#
----------------- fine di parte di mio.mkdisk -----------------
```

9 Appendice B. - Script di SHUTDOWN di esempio

- 9.1 (Slackware)
- 9.2 (Debian)

9.1 Slackware - /etc/rc.d/rc.6

```
#! /bin/sh
#
# rc.6          This file is executed by init when it goes into runlevel
#               0 (halt) or runlevel 6 (reboot). It kills all processes,
#               unmounts file systems and then either halts or reboots.
#
# 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@bizsystems.com >
#               to add call to rc.raidown
  # Set the path.
  PATH=/sbin:/etc:/bin:/usr/bin

  # Set linefeed mode to avoid staircase effect.
  stty onlcr

  echo "Running shutdown script $0:"

  # Find out how we were called.
  case "$0" in
        *0)
```

```
                message="The system is halted."
                command="halt"
                ;;
        *6)
                message="Rebooting."
                command=reboot
                ;;
        *)
                echo "$0: call me as \"rc.0\" or \"rc.6\" please!"
                exit 1
                ;;
  esac

##### Salva informazioni sul boot e sullo stato raid #####
#
if [ -x /etc/rc.d/rc.raidown ]; then
   /etc/rc.d/rc.raidown
fi
################## fine raid boot ########################

  # Kill all processes.
  # INIT is supposed to handle this entirely now, but this didn't always
  # work correctly without this second pass at killing off the processes.
  # Since INIT already notified the user that processes were being killed,
  # we'll avoid echoing this info this time around.
  if [ "$1" != "fast" ]; then # shutdown did not already kill all processes
    killall5 -15
    killall5 -9
  fi

  # Try to turn off quota and accounting.
  if [ -x /usr/sbin/quotaoff ]
  then
        echo "Turning off quota."
        /usr/sbin/quotaoff -a
  fi
  if [ -x /sbin/accton ]
  then
        echo "Turning off accounting."
        /sbin/accton
  fi

  # Before unmounting file systems write a reboot or halt record to wtmp.
  $command -w

  # Save localtime
  [ -e /usr/lib/zoneinfo/localtime ] && cp /usr/lib/zoneinfo/localtime /etc

  # Asynchronously unmount any remote filesystems:
  echo "Unmounting remote filesystems."
```

```
  umount -a -tnfs &

  # Turn off swap, then unmount local file systems.
  echo "Turning off swap."
  swapoff -a
  echo "Unmounting local file systems."
  umount -a -tnonfs
  # Don't remount UMSDOS root volumes:
  if [ ! "`mount | head -1 | cut -d ' ' -f 5`" = "umsdos" ]; then
    mount -n -o remount,ro /
  fi

################ per gli array raid ########################
# Ferma tutti gli array raid conosciuti (tranne root)
if [ -x /sbin/mdstop ]; then
  echo "Stopping raid"
  /sbin/mdstop -a
fi
##########################################################

  # See if this is a powerfail situation.
  if [ -f /etc/powerstatus ]; then
    echo "Turning off UPS, bye."
    /sbin/powerd -q
    exit 1
  fi

  # Now halt or reboot.
  echo "$message"
  [ ! -f /etc/fastboot ] && echo "On the next boot fsck will be FORCED."
  $command -f
############### end rc.6 #################################
```

9.2 Debian bo - /etc/init.d/halt and /etc/init.d/reboot

Le modifiche mostrate di seguito per i file bo halt e reboot di Debian NON SONO TESTATE. Se le testate, per favore inviatemi una e-mail in modo che io possa togliere questo commento.

9.2.1 /etc/init.d/halt

```
#! /bin/sh
#
# halt          The commands in this script are executed as the last
#               step in runlevel 0, ie halt.
#
# Version:      @(#)halt  1.10  26-Apr-1997  miquels@cistron.nl
#

PATH=/sbin:/bin:/usr/sbin:/usr/bin
```

```
##### Salva informazioni sul boot e sullo stato raid #####
#
if [ -x /etc/rc.d/rc.raidown ]; then
   /etc/rc.d/rc.raidown
fi
################# fine raid boot #######################

# Kill all processes.
echo -n "Sending all processes the TERM signal... "
killall5 -15
echo "done."
sleep 5
echo -n "Sending all processes the KILL signal... "
killall5 -9
echo "done."

# Write a reboot record to /var/log/wtmp.
halt -w

# Save the random seed between reboots.
/etc/init.d/urandom stop

echo -n "Deactivating swap... "
swapoff -a
echo "done."

echo -n "Unmounting file systems... "
umount -a
echo "done."

mount -n -o remount,ro /

################ per gli array raid ########################
# Ferma tutti gli array raid conosciuti (tranne root)
if [ -x /sbin/mdstop ]; then
  echo "Stopping raid"
  /sbin/mdstop -a
fi
##########################################################

# See if we need to cut the power.
if [ -x /etc/init.d/ups-monitor ]
then
        /etc/init.d/ups-monitor poweroff
fi

halt -d -f
############# end halt ###################
```

9.2.2 /etc/init.d/reboot

```
#! /bin/sh
#
# reboot        The commands in this script are executed as the last
#               step in runlevel 6, ie reboot.
#
# Version:      @(#)reboot  1.9  02-Feb-1997  miquels@cistron.nl
#

PATH=/sbin:/bin:/usr/sbin:/usr/bin

##### Salva informazioni sul boot e sullo stato raid #####
#
if [ -x /etc/rc.d/rc.raidown ]; then
   /etc/rc.d/rc.raidown
fi
################## fine raid boot ########################

# Kill all processes.
echo -n "Sending all processes the TERM signal... "
killall5 -15
echo "done."
sleep 5
echo -n "Sending all processes the KILL signal... "
killall5 -9
echo "done."

# Write a reboot record to /var/log/wtmp.
halt -w

# Save the random seed between reboots.
/etc/init.d/urandom stop

echo -n "Deactivating swap... "
swapoff -a
echo "done."

echo -n "Unmounting file systems... "
umount -a
echo "done."

mount -n -o remount,ro /

################ per gli array raid ########################
# Ferma tutti gli array raid conosciuti (tranne root)
if [ -x /sbin/mdstop ]; then
  echo "Stopping raid"
  /sbin/mdstop -a
fi
```

```
##########################################################

echo -n "Rebooting... "
reboot -d -f -i
```

10 Appendice C. - altri file di setup

10.1 linuxrc 4.11 (linuxrc file)

10.2 loadlin – linux.bat file - boot.par 4.14 (linux.bat file - boot.par)

10.3 linuxthreads Makefile.diff 4.6 (linuxthreads Makefile.diff)

10.4 raid1.conf 6.2 (raid1.conf)

10.5 raid5.conf 6.1 (raid5.conf)

10.6 raidboot.conf 4.13 (raidboot.conf)

10.7 rc.raidown 13 (rc.raidown)

11 Appendice D. - script linuxrc e shutdown obsoleti

11.1 Lavoro obsoleto - linuxrc

Questo file linuxrc funziona benissimo con la procedura di shutdown mostrata di seguito.

```
 ---------------------- linuxrc --------------------
#!/bin/sh
# ver 1.07 2-12-98
# linuxrc - for raid1 using small dos partition and loadlin
#

# mount the proc file system
/bin/mount /proc

# This may vary for your system.
# Mount the dos partitions, try both
# in case one disk is dead
/bin/mount /dosa
/bin/mount /dosc

# Set a flag in case the raid status file is not found
# then check both drives for the status file
RAIDOWN="raidstat.ro not found"
/bin/echo "Reading md0 shutdown status."
if [ -f /dosa/raidboot/raidstat.ro ]; then
  RAIDOWN=‘/bin/cat /dosa/raidboot/raidstat.ro‘
  RAIDREF=‘/bin/cat /dosc/raidboot/raidgood.ref‘
```

```
else
  if [ -f /dosc/raidboot/raidstat.ro ]; then
    RAIDOWN=`/bin/cat /dosc/raidboot/raidstat.ro`
    RAIDREF=`/bin/cat /dosc/raidboot/raidgood.ref`
  fi
fi

# Test for a clean shutdown with all disks operational
if [ "${RAIDOWN} != ${RAIDREF}" ]; then
  echo "ERROR ${RAIDOWN}"
#  Use the next 2 lines to BAIL OUT and leave rescue running
   /bin/echo 0x100>/proc/sys/kernel/real-root-dev
   exit                 # leaving the error files in dosa/raidboot,etc...
fi

# The raid array is clean, proceed by removing
# status file and writing a clean superblock
/bin/rm /dosa/raidboot/raidstat.ro
/bin/rm /dosc/raidboot/raidstat.ro
/sbin/mkraid /etc/raid1.conf -f --only-superblock

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

# Mount raid array
echo "Mounting md0, root filesystem"
/sbin/mdadd -ar

#  If there are errors - BAIL OUT and leave rescue running
if [ $? -ne 0 ]; then
   echo "RAID device has errors"
#  Use the next 3 lines to BAIL OUT
   /bin/rm /etc/mtab            # remove bad mtab
   /bin/echo 0x100>/proc/sys/kernel/real-root-dev
   exit
fi

# else tell the kernel to switch to /dev/md0 as the /root device
# The 0x900 value the device number calculated by:
#  256*major_device_number + minor_device number
/bin/echo 0x900>/proc/sys/kernel/real-root-dev

# umount /proc to deallocate initrd device ram space
/bin/umount /proc
/bin/echo "/dev/md0 mounted as root"
exit
#------------------ end linuxrc ----------------------
```

11.2 Lavoro obsoleto - script di shutdown

Questa procedura di shutdown funziona benissimo con **linuxrc** indicato in precedenza

Per catturare lo stato di shutdown dell'array raid, inserisci subito prima che i file system vengano smontati:

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

Dopo che tutti i file system sono smontati (il file system di root non verrà smontato) aggiungi:

```
# il dispositivo di root rimane montato RO
# monta i file system dos RW
mount -n -o remount,ro /
echo "Writing RAID read-only boot FLAG(s)."
mount -n /dosa
mount -n /dosc
# create raid mounted RO flag in duplicate
# containing the shutdown status of the raid array
echo ${RAIDSTATUS} > /dosa/raidboot/raidstat.ro
echo ${RAIDSTATUS} > /dosc/raidboot/raidstat.ro

umount -n /dosa
umount -n /dosc

# Ferma tutti gli array raid (tranne root)
echo "Stopping raid"
mdstop -a
```

Questo fermerà in modo pulito tutti i dispositivi raid eccetto root. Lo stato di root viene passato al prossimo boot in **raidstat.ro**.

Quello che segue è l'intero script di shutdown preso dal mio vecchio sistema raid1 Slackware, ho aggiornato raid1 alla nuova procedura con il file /etc/raidboot.conf

```
#! /bin/sh
#
# rc.6          This file is executed by init when it goes into runlevel
#               0 (halt) or runlevel 6 (reboot). It kills all processes,
#               unmounts file systems and then either halts or reboots.
#
# 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

  # Set the path.
  PATH=/sbin:/etc:/bin:/usr/bin

  # Set linefeed mode to avoid staircase effect.
  stty onlcr
```

```
echo "Running shutdown script $0:"

# Find out how we were called.
case "$0" in
      *0)
              message="The system is halted."
              command="halt"
              ;;
      *6)
              message="Rebooting."
              command=reboot
              ;;
      *)
              echo "$0: call me as \"rc.0\" or \"rc.6\" please!"
              exit 1
              ;;
esac

# Kill all processes.
# INIT is supposed to handle this entirely now, but this didn't always
# work correctly without this second pass at killing off the processes.
# Since INIT already notified the user that processes were being killed,
# we'll avoid echoing this info this time around.
if [ "$1" != "fast" ]; then # shutdown did not already kill all processes
  killall5 -15
  killall5 -9
fi

# Try to turn off quota and accounting.
if [ -x /usr/sbin/quotaoff ]
then
      echo "Turning off quota."
      /usr/sbin/quotaoff -a
fi
if [ -x /sbin/accton ]
then
      echo "Turning off accounting."
      /sbin/accton
fi

# Before unmounting file systems write a reboot or halt record to wtmp.
$command -w

# Save localtime
[ -e /usr/lib/zoneinfo/localtime ] && cp /usr/lib/zoneinfo/localtime /etc

# Asynchronously unmount any remote filesystems:
echo "Unmounting remote filesystems."
umount -a -tnfs &
```

```
# you must have issued
# 'cat /proc/mdstat | grep md0 > {your boot vol}/raidboot/raidgood.ref'
# before linuxrc will execute properly with this info
RAIDSTATUS=`/bin/cat /proc/mdstat | /usr/bin/grep md0 # capture raid status`

# Turn off swap, then unmount local file systems.
# clearing mdtab as well
echo "Turning off swap."
swapoff -a
echo "Unmounting local file systems."
umount -a -tnonfs

# Don't remount UMSDOS root volumes:
if [ ! "`mount | head -1 | cut -d ' ' -f 5`" = "umsdos" ]; then
  mount -n -o remount,ro /
fi

# root device remains mounted
# mount dos file systems RW
echo "Writing RAID read-only boot FLAG(s)."
mount -n /dosa
mount -n /dosc
# create raid mounted RO flag in duplicate
# containing the shutdown status of the raid array
echo ${RAIDSTATUS} > /dosa/raidboot/raidstat.ro
echo ${RAIDSTATUS} > /dosc/raidboot/raidstat.ro

umount -n /dosa
umount -n /dosc

# Stop all the raid arrays (except root)
echo "Stopping raid"
mdstop -a

# See if this is a powerfail situation.
if [ -f /etc/power_is_failing ]; then
  echo "Turning off UPS, bye."
  /sbin/powerd -q
  exit 1
fi

# Now halt or reboot.
echo "$message"
[ ! -f /etc/fastboot ] && echo "On the next boot fsck will be FORCED."
$command -f
```

12 Appendice E. - La patch di Gadi per il raid stop per il kernel linux

```
--- linux/drivers/block/md.c.old       Fri Nov 21 13:37:11 1997
+++ linux/drivers/block/md.c    Sat Dec  6 13:34:28 1997
@@ -622,8 +622,13 @@
       return do_md_run (minor, (int) arg);

     case STOP_MD:
-      return do_md_stop (minor, inode);
-
+      err = do_md_stop(minor, inode);
+      if (err) {
+        printk("md: enabling auto mdstop for %s\n",
kdevname(inode->i_rdev));
+        md_dev[minor].auto_mdstop = 1;
+      }
+      return err;
+
     case BLKGETSIZE:   /* Return device size */
     if  (!arg)  return -EINVAL;
     err=verify_area (VERIFY_WRITE, (long *) arg, sizeof(long));
@@ -692,6 +697,10 @@

   sync_dev (inode->i_rdev);
   md_dev[minor].busy--;
+  if (!md_dev[minor].busy && md_dev[minor].auto_mdstop) {
+       do_md_stop(minor, inode);
+       md_dev[minor].auto_mdstop = 0;
+  }
 }

 static int md_read (struct inode *inode, struct file *file,
--- linux/include/linux/md.h~   Fri Nov 21 13:29:14 1997
+++ linux/include/linux/md.h    Fri Nov 21 13:29:14 1997
@@ -260,6 +260,7 @@
   int                 repartition;
   int                 busy;
   int                 nb_dev;
+  int                 auto_mdstop;
   void                *private;
 };
```

13 Appendice F. - rc.raidown

Copia il testo che segue nello script **rc.raidown** e salvalo in **/etc/rc.d**.

```
#! /bin/sh
#
```

```
# rc.raidown    This file is executed by init when it goes into runlevel
#               0 (halt) or runlevel 6 (reboot). It saves the status of
#               a root mounted raid array for subsequent re-boot
#
# Version:      1.08    3-25-98 Michael A. Robinton < michael@bizsystems.com >
#
############ Save raid boot and status info ##############
if [ -f /etc/raidboot.conf ]
then
  {
  read RaidBootDevs
  read RaidStatusPath
  read RaidConfigEtc
  } < /etc/raidboot.conf

# you must have issued
#       cat /proc/mdstat | grep md0 >
#               {your boot vol mnt(s)}/{RaidStatusPath}/raidgood.ref
# before linuxrc will execute properly with this info
#
#       capture raid status
  RAIDSTATUS=‘/bin/cat /proc/mdstat | /usr/bin/grep md0‘
  mkdir /tmp/raid$$
  echo "Writing RAID read-only boot FLAG(s)."
  for Device in ${RaidBootDevs}
  do
# get mount point for raid boot device or use tmp
    RBmount=$( cat /proc/mounts | /usr/bin/grep ${Device} )
    if [ -n ${RBmounts} ]; then
      RBmount=$( echo ${RBmount} | cut -f 2 -d ’ ’ )
    else
      RBmount="/tmp/raid$$"
      mount ${Device} ${RBmount}
    fi
  if [ -d ${RBmount}/${RaidStatusPath} ]; then
# Create raid mounted RO flag = shutdown status of raid array
    echo ${RAIDSTATUS} > ${RBmount}/${RaidStatusPath}/raidboot.ro
# Don’t propagate ’fstab’ from ramdisk
    if [ -f /linuxrc ]; then
      FSTAB=
    else
      FSTAB=fstab
    fi
    pushd /etc
# Save etc files for rescue system
    /bin/tar --ignore-failed-read \
        -cf ${RBmount}/${RaidStatusPath}/raidboot.etc \
        raid*.conf mdtab* ${FSTAB} lilo.conf
    popd
# Create new raidboot.cfg
```

```
    {
    /bin/echo ${RaidBootDevs}
    /bin/echo ${RaidStatusPath}
    /bin/echo ${RaidConfigEtc}
    } > ${RBmount}/${RaidStatusPath}/raidboot.cfg
    /bin/umount ${RBmount}
  fi
  done
  rmdir /tmp/raid$$
  echo "Raid boot armed"
fi
################# end raid boot ########################
```

14 Appendice G. - teoria del funzionamento di linuxrc

Questa è la forma complessa del file linuxrc per raid montato come root. Deve essere elaborato con 'bash' o una shell che riconosce le funzioni di shell.

Il vantaggio sta nel fatto che è generico e non dipende dai file di startup e dai parametri che si trovano nell'immagine **initrd**

Un parametro **Raid_Conf** passato a **linuxrc** dal kernel al boot da lilo o loadlin contiene un puntatore ai dispositivi di boot e alla posizione dei 2 file raidboot necessari per **linuxrc** (*raidboot.etc e raidboot.cfg scritti dallo script di shutdown*).

raidboot.etc contenente i file 'tar'-ati:

```
raid*
mdtab*
fstab
lilo.conf              ( se applicabile )
```

del sistema primario che vengono trasferiti sulla directory initrd **/etc** durante lo startup. Con cura, questo file può essere modificato, se necessario, quando il tuo sistema ha dei guai seri. **raidboot.cfg** contiene il nome della partizione di boot in uso e i backup applicabili, così come il percorso per il resto del file di startup di raid usato da **linuxrc**. Questo file viene normalmente creato dal file di shutdown e può essere creato manualmente se necessario.

raidboot.cfg è della forma: 3 linee - nessun commento

```
/dev/bootdev1 /dev/bootdev2 [/dev/bootdev3 ... e così via]
percorso_di/raid-status
nome_del_file_raidX.conf
```

percorso_di/raid-status non comprende il nome del punto di mount **nome_del_file_raidX.conf** è quello che si trova in /etc ed è normalmente usato per **ckraid** e **mkraid**.

I seguenti file addizionali si trovano sulle partizioni di boot raid permanenti. Questo è solitamente lo stesso di sopra, ma nelle situazioni di emergenza può essere caricato da dovunque sia disponibile, come da un floppy di boot.

- **raidgood.ref** creato dal comando cat /proc/mdstat | grep md0 > /{raid_status_path}/raidgood.ref

  vedi 4.12 (script di shutdown) per salvare questo file ed il prossimo

- **raidstat.ro** creato ad ogni shutdown dal file di shutdown rc, salvando lo stato di uscita dell'array raid.