RAID Usage with FS Linux 6 (sarge), 7 (etch), & 8 (lenny)

Ed Himwich and Jonathan Quick

Table of Contents

INTRODUCTION

This document describes how to use and maintain the software RAID system that is installed as standard in FS Linux 6, 7, & 8. The document is divided into several small sections. The discussion starts with an **OVERVIEW** section that describes the system and the rationale for its use. The disk layout is described as well. The **NORMAL OPERATION** section describes how the system is intended to be used for normal operations and defines some nomenclature. The **RAID STATUS** section explains how to determine if the RAID system is still in process of refreshing the disks or if a fault has occurred. The **DISK ROTATION** section explains how to rotate the disks to update the back-ups. The **REFRESHING A STALE SECONDARY DISK** section explains how to re-introduce an old disk back into the RAID system to bring it back up to date. The **IF A DISK FALLS OUT OF THE RAID** section describes how to deal with a report from the RAID system that one of the disks is no longer in the RAID. The **RECOVERING FROM A DISK FAILURE** section explains how to handle a disk failure. The **RECOVERING A LOST FILE OR FILES** section explains how to recover data from an inactive disk. The **APPLYING AN UPDATE** section describes a method to use for updates so that it is relatively easy to recover if the update fails for some reason. The **RECOVERING FROM A STALE SHELF DISK** section explains how to return the RAID system back to the state stored on an inactive disk. The **REPLACEMENT DISKS** section covers considerations for buying replacement disks and procedures for initializing them. The **SCRIPTS** section documents two useful scripts. The **APPENDIX A. DISK RESET RECIPE** section describes how to reset the mount count on a stale disk so that a back-up can be used to refresh it without having to reboot the back-up many times to make it "less stale". The **APPENDIX B. TWO COMPUTER ROTATION** section recommends a scheme for preparation and handling of computer, and disk, rotation if you have bought two identical FS computers for an increased level of reliability. The **APPENDIX C. MULTIPLE COMPUTER NIC SUPPORT** section describes how to modify the system set-up to allow one removable disk to be used in two computers that are the same except for the MAC address of the NIC. This appendix is used in configuring the two computer rotation described in **APPENDIX B**, but can also be useful for other systems where a disk may be used in more than one identical computer. The **APPENDIX D. CLONING DISKS** section describes how to clone a disk. This appendix is referred to from **APPENDIX B**. The **APPENDIX E. SPARE COMPUTER REFRESH SCRIPT** section gives an example of a script for refreshing the `/usr2` partition of a spare computer for stations that have one. This appendix is referred to from **APPENDIX B**.

OVERVIEW

Beginning with FS Linux 6, the standard disk configuration for a FS computer is a software RAID1 scheme. In this approach two drives are normally inserted all the time. The operating system (this approach does NOT use a RAID hardware controller) maintains these as exact mirrors of each other. If one disk should fail, even during normal operations, there should be no loss of data or system functionality. This also provides a continuous automatic back-up of the

operational software. The robustness and continuous backup are the primary advantages of using the RAID1 approach. The primary disadvantage is that because the back-up occurs automatically and continuously, the back-up disk cannot be used to recover a file that was accidentally deleted or recover from a change that was made since the last back-up. However, stations that have three disks can still maintain a "Shelf" back-up disk that can be used to recover from such errors, provided the back-up is recent enough. It is recommended that all stations have three disks.

Stations with only two disks are also recommended to run in the RAID1 configuration. However, neither disk rotation, nor any other operation in this document that utilizes a third disk is applicable to these stations. The **APPLYING AN UPDATE** section covers two disk stations as special case, but as with three disk stations, the update is applied to only a single disk initially to allow a relatively easy recovery if necessary.

For IDE (PATA) disk, the partition layout on the disks is:

RAID Device	**Disk devices**	**File System**
`md0`	`hda1, hdc1`	`/`
`md1`	`hda5, hdc5`	(swap area)
`md2`	`hda6, hdc6`	`/usr2`

For SCSI and SATA disks, the partition layout on the disks is:

RAID Device	**Disk devices**	**File System**
`md0`	`sda1, sdb1`	`/`
`md1`	`sda5, sdb5`	(swap area)
`md2`	`sda6, sdb6`	`/usr2`

The RAID devices are identified as **MD** (multi-disk) devices: `md1`, `md2`, and `md3`.

The RAID system automatically mounts and maintains both disks in a mirrored state as long as they are booted together. If one disk is removed before a boot it becomes "stale". The RAID system notes this and will not automatically re-use it. It must be "added" back into the array before it can be used again. This is true of any "stale" disk (mounted fewer times than the other or others). The process of adding a disk back in is described in **REFRESHING A STALE SECONDARY DISK**. When a stale disk is being brought back up to date it is referred to as a "recovery" taking place.

The system can be shutdown with no special consideration of the RAID as long as which disks are installed does not change before the next boot. If the disks are to be changed after a shutdown, it should be verified that there is no recovery ongoing before shutting down. The procedure for checking this is described in the **RAID STATUS** section.

When the system is shutdown some of the last display messages may say something to the effect that stopping one or more of the **MD** disks "failed (busy)". These errors can safely be ignored as long there was no recovery in progress right before the shutdown.

The system can be shutdown both accidently and on purpose in many ways. The most graceful way is to use the `shutdown` command as `root`, but other ways include using `Ctrl-Alt-Del` on `vt1-vt6` (and remove the power during the POST of the computer during the boot), by a power failure, or by a hard re-boot, e.g., cycling the power switch to force a re-boot. After any shutdown and with the same disks installed on reboot the RAID system will recover automatically. It is recommended to avoid accidental shutdowns if possible and to avoid power failures and hard reboots in general. An inexpensive un-interruptible power supply (UPS) is a relatively cheap way to avoid accidental shutdowns due to short power glitches.

If the system was shutdown in anyway including accidents, like power failures, without verifying that no recovery was occurring, the disks should not be changed until the system has been rebooted and it has been verified that there is no recovery in progress. The procedure for checking this is described in the **RAID STATUS** section.

The `mdadm` utility (`man mdadm` for more information) has various options for managing the RAID devices. However, the `refresh_secondary` script described elsewhere in this document is the main tool you will typically need to use.

The **MD** system monitors the status of the disks and sends an e-mail to the `root` user if any problem is detected. It is important that the root user's mail be check frequently for these messages. In FSL7 and later, the standard set-up is for e-mail sent to `root` to be redirected to another user that logins in more frequently, typically `oper`. The **IF A DISK FALLS OUT OF THE RAID** section describes how to deal with a report from the **MD** system that one of the disks is no longer in the RAID.

Disks from different computers should not be mixed, nor should disks from one computer be cloned for use in another, unless the procedure in **APPENDIX D. CLONING DISKS** is used. Each RAID (which can be thought of as set of disks set-up as a RAID) has its own identity. If disks from different computers with the same RAID identity are mixed the results can be difficult to deal with and possibly catastrophic. For this reason, a RAID should be created from scratch for each computer according to the procedure in **APPENDIX D** and then not mixed with disks from another computer. Note that can you add a disk from one computer to the RAID on another computer if the disk to be added is at least as big as the smallest disk in the RAID and you re-initialize the disk to be added first. Please follow the procedure in the **REPLACEMENT DISKS** section to set it up properly.

Disks should be only be inserted or removed when the computer power is off. It is not possible to “hot swap” disks.

NORMAL OPERATION

Normal operation of a FS Linux 6, 7, or 8 system requires two hard drives. One installed as the master on the primary IDE (PATA) interface, `hda` (the slot is usually labeled "Primary"), and a second on the master of the secondary IDE interface, hdc (the slot is usually labeled "Secondary"). For SATA disks, the "Primary" is in the slot connected to lowest numbered SATA interface, `sda`; the "Secondary", the next lowest, `sdb`. For SCSI disks the Primary disk has the lowest SCSI ID, `sda`; the Secondary, the next lowest, `sdb`. Thus for SCSI disks, the disk designation does not depend on the actual slots used for the disks.

The Primary and Secondary disks will normally be running in a RAID1 (mirroring) configuration. A third disk which contains a mirror image of the system in a working state from an earlier date is kept on the shelf as a spare and is referred to as the "Shelf" disk. Periodically, the disks should be "rotated" so a more recent copy of the working system can be stored on the shelf and the disk that had been on the shelf can be exercised. A period of no more than two months is recommended for disk rotation. A rotation is also recommended before any significant or non-reversible update in order to make recovery easier in case of problems.

The `root` user's (or user that the `root` user's e-mail is redirected to, typically `oper`) e-mail should be checked at least weekly, but preferably more often. If any messages about the RAID system are received, they need to be followed up on promptly or the utility and safety provided by the system may be degraded.

The disks are usually labeled "1", "2", and "3" to help keep track of which is which, but any suitable labeling scheme (distinct from the slot labels of "Primary" and "Secondary") will do. For SCSI disk you may want reusable labels that say "Primary" and "Secondary" that can be moved between disks "1", "2" and "3" as their roles change.

RAID STATUS

You can check on the RAID mirroring status by using the command:

```
cat /proc/mdstat
```

and looking for messages about expected completion time. When none of the RAID devices are showing a completion time (i.e. when `mdstat` shows no `recovery=`), mirroring is up to date. A script `mdstat`, described in the **SCRIPTS** section, can also be used to check the status and produces more readable output than the raw `cat /proc/mdstat` command if the case of FSL6.

Normally each **MD** device will show the string `[2/2]` in the listing of its status. This indicates that both disks are in the array are active. If instead the status shows `[2/1]` for one or more of the **MD** devices, one of the disks had fallen out of one or more of the **MD** devices and the situation needs to investigated. The disks in the array are listed. By implication, you can

determine which are missing. The section **IF A DISK FALLS OUT OF THE RAID** contains suggestions on what to do if this situation occurs.

It is possible to get more detailed information about the status of a RAID device using (as `root`):

```
mdadm --detail /dev/mdX
```

where `X` is the number of the RAID device (`0`, `1`, or `2`).

The log file `/var/log/syslog` can be examined (as `root`) for information on the status of the disk system and clues to what happened if something has gone wrong. Old versions of `syslog`: `syslog.1`, `syslog2.gz`, etc. may also be useful.

DISK ROTATION

When rotating the disks, the system should be checked to make sure that the RAID system is not currently recovering (see **RAID STATUS**) before shutting down. When the system is quiet, shut it down gracefully, e.g., with:

```
shutdown -h now
```

When the system has stopped, turn it off (this may have happened automatically), remove the disk in the "Primary" slot, affix a label to it with the current date and place it on the shelf. Move the disk in the "Secondary" slot to the "Primary" slot. Take the previous "Shelf" disk and place it in the "Secondary" slot. Then follow the directions below in **REFRESHING A STALE SECONDARY DISK**.

For SCSI disks, it may be necessary to manipulate the SCSI IDs (change the jumpers) of the disks so that the new Primary has a lower SCSI ID than the new Secondary.

REFRESHING A STALE SECONDARY DISK

Make sure both slot key latches are turned "on" and then power up the system. Since the new "Secondary" disk is stale (mounted fewer times since it was last current) compared to the new "Primary" drive, the system should boot up using only the "Primary" disk (which you can confirm by watching the disk activity lights.) If the Secondary disk is booted, you can use the procedure in **APPENDIX A. DISK RESET RECIPE** to reset the Secondary disk so that it will not be selected in preference to the Primary disk and try again.

Once it has booted, log in as `root` and add the "Secondary" disk back into the array using the commands for IDE (PATA) disks:

```
mdadm /dev/md0 -a /dev/hdc1
mdadm /dev/md1 -a /dev/hdc5
mdadm /dev/md2 -a /dev/hdc6
```

or for SCSI and SATA disks:

```
mdadm /dev/md0 -a /dev/sdb1
mdadm /dev/md1 -a /dev/sdb5
mdadm /dev/md2 -a /dev/sdb6
```

These commands can alternately be executing using the script `refresh_secondary` described in the **SCRIPTS** section at the end of this document. (Using the script in the **SCRIPTS** section is less error prone and it also includes additional commands to make it more reliable.) These commands re-add the stale disk into the array. Once this is accomplished, the RAID system will work to bring the "Secondary" disk up to date, i.e., recover it. This may take an hour or two depending on your disk sizes. If convenient, it can be started before leaving for the evening or the weekend so that it doesn't have to be waited for. The system, including the FS, can be used as normal when a recovery is underway. The status of the mirroring can be checked with:

```
cat /proc/mdstat
```

or the script:

```
mdstat
```

as described in the **RAID STATUS** section.

The RAID recovery process should be allowed to run to completion before rebooting.

IF A DISK FALLS OUT OF THE RAID

If an e-mail is received from the **MD** system indicating that a disk has fallen out of the RAID or it is discovered incidentally by noting that the `mdstat` script returns `[2/1]` for one or more of the **MD** devices, the situation needs to investigated. If this is the first time this has happened and the same disk has fallen out of any **MD** devices that are missing a disk, it may be sufficient to merely re-introduce the disk into the array. You should be able to tell which disk has fallen out of each **MD** device from the e-mail messages and/or the output of `mdstat` and/or output from `mdadm --detail ...` commands as described in the **RAID STATUS** section. If different **MD** devices are missing different disks, the recovery will be more complicated, please contact Ed (Ed.Himwich@nasa.gov) for advice or suggestions on resources to deal with the situation.

If the same disk is missing from all **MD** devices that are missing disks, you can attempt to add it back into the array at the next convenient opportunity. At that time, if it is the Primary disk that has fallen out of the array, shut the system down gracefully, swap the disks (adjust IDs as appropriate if you have SCSI disks), and reboot. At this point, the disk that fell out of the array is the Secondary disk either because it started off that way or because you swapped the disks. You can then follow the directions in the section on **REFRESHING A STALE SECONDARY DISK** above to recover.

If this is either a recurring problem or there is an opportunity to be more thorough, the log file `/var/log/syslog` can be examined (as `root`) for clues to what happened. Old versions of `syslog`: `syslog.1`, `syslog2.gz`, etc. may also be useful. If the disk has clearly failed or attempts to add it back into the array are unsuccessful, you can refer to the **RECOVERING FROM A DISK FAILURE** section below for advice.

RECOVERING FROM A DISK FAILURE

If one of the disks in the RAID array fails during normal operation (which should be reported via e-mail, to the user designated to receive `root` e-mail, typically `oper`, by the **MD** monitoring daemon), it can removed at the next convenient opportunity when the system is not otherwise in use. At that time, shut the system down:

```
shutdown -h now
```

When the system has stopped, turn it off (this may have happened automatically), remove the failed disk (which should have shown no activity during the shutdown process). If the failed disk was the Primary disk, put the still working disk in the Primary slot. Place the "Shelf" disk in secondary slot. (Adjust IDs as appropriate if you have SCSI disks.) Then follow the directions in the section on **REFRESHING A STALE SECONDARY DISK** above. Please see the **REPLACEMENT DISKS** section for information on obtaining a new third disk.

RECOVERING A LOST FILE OR FILES

You can use the RAID system to recover from accidently losing one or more files if the "Shelf" disk still contains the file you want to recover.

To recover a lost file, shut the system down and boot with only the "Shelf" disk installed in the "Primary" slot. The "Secondary" slot should be empty. Once the system boots, copy the file or files you wish to recover to another media (floppy, CD, DVD, another computer's hard disk via the network etc.) Then reboot with the normal disks in their original positions. Copy the file or files from the other media back to the operational system.

APPLYING AN UPDATE

To make it easier to recovery from an error in a significant and/or irreversible update to the system, it is recommended that a disk rotation be performed just before the update is applied. Once the rotation has refreshed the previously stale "Shelf" disk, shut the system down and place the new "Secondary" disk on the shelf along with the old "Secondary" (now "Shelf") disk. In case you need to undo the update, this gives you a second disk to recover from in case one of them fails for some reason. Then boot the system with only the "Primary" disk installed. Perform the update on this disk only.

If the update is successful, shutdown the system and re-insert the new "Secondary" disk you just placed on the shelf and follow the directions above for **REFRESHING A STALE SECONDARY DISK**.

If the update is not successful, after shutting down you can remove the now bad "Primary" disk. Put the new "Secondary" disk from the shelf in the "Primary" slot and the "Shelf" (old "Secondary") disk in the Secondary slot and reboot. Then follow the directions for **REFRESHING A STALE SECONDARY DISK**. This will return you to your working system. Then use the procedure in **APPENDIX A. DISK RESET RECIPE** to reset the boot count on the disk with the failed update. Note that now the disk with the failed update, i.e., the "Shelf" disk, is bad. Be sure to do another disk rotation to produce an up-to-date "Shelf" disk before trying the update again.

For stations with two disks, no disk rotation is possible of course, but updates should still be applied with only one disk in the RAID array active. This will allow recovery from the other disk in case of problems. If it should become necessary to recover from the "Secondary" disk, reset the boot count on the disk with the failed update (old "Primary") using the procedure in **APPENDIX A. DISK RESET RECIPE**. Then place the old Secondary disk in the Primary slot and the old Primary disk in the Secondary slot and refresh the new Secondary.

RECOVERING FROM A STALE SHELF DISK

If for some reason both your active disks are corrupted (i.e. "bad"), you can still recover from your "Shelf" disk. Before doing this, you should consider your situation carefully. If both active disks become corrupt during normal operations, it may be that the computer is at fault and putting the "Shelf" disk in the computer will put it at risk. On the other hand, if the corruption occurred due to miscopying or other operational errors, the computer is not a priori suspect.

If you are going to recover from the "Shelf" disk, you will need to convince the RAID system that the "Shelf" disk is not the stale disk. To do this use the procedure described in **APPENDIX A. DISK RESET RECIPE**. Once this has been accomplished the "Shelf" disk can be booted as the "Primary" with either of the bad disks as the "Secondary". In this configuration, you should then perform REFRESHING **A STALE SECONDARY DISK** and then do a **DISK ROTATION** to bring the system back to full normal redundancy.

REPLACEMENT DISKS

If you have a disk failure, you should buy a replacement disk that is exactly the same size or bigger than the smallest remaining disk. Make sure to initialize it to have each partition at least as big as the same partition on your smallest disk. This will assure that all the disks can then be used interchangeably. Note that the RAID size was determined at build time to fit within the smallest of the corresponding partitions on the then included disks. Note that it is possible to enlarge the RAID sizes to take advantage of any additional space on the smallest of the current set of disks - contact Ed (Ed.Himwich@nasa.gov) for more details.

You can partition the disk using the installer DVD for your system according to the directions in, e.g., `fs8linux_DVD.txt`. After setting up the partitions for RAID usage, you can stop at the point of creating the **MD** devices (using `fs8linux_DVD.txt`, follow the directions for ending the installation near the start of step 10 of phase one). Alternatively, if the disk has previously been set-up for use in a RAID using the `fs8linux_DVD.txt` instructions, you can instead use the procedure in **APPENDIX A: DISK RESET RECIPE** to reset the disks.

Once the disk has been initialized or reset, it should be added into array as the secondary disk using the directions in the **REFRESHING A STALE SECONDARY DISK** section. After this is done for the first time, `grub` should be installed in the master boot record according the directions at the end of the installation document (using `fs8linux_DVD.txt`, following step 12 in phase two). Note the `refresh_secondary` script does this automatically.

SCRIPTS

Two utility scripts are described here. The `refresh_secondary` script is used by the `root` user when adding a stale disk into the RAID array. The `mdstat` script is intended for use by any user to check whether the mirroring is currently recovering. More flexibility can be achieved of course by using low-level commands.

refresh_secondary

The following utility script `refresh_secondary` can be used by `root` to automate adding the disks of a stale secondary into the disk RAID array for refreshing. One version is provided for IDE (PATA) disk systems and another for SCSI or SATA disks systems; examples can be found in the `/usr2/fs/misc/` directory as `refresh_secondary.hdc` and `refresh_secondary.sdb`, respectively. A copy of the appropriate script can be placed in the `/usr/local/sbin` directory and set-up by using the following commands as `root`, for IDE (PATA) disks:

```
cd /usr/local/sbin
cp -a /usr2/fs/misc/refresh_secondary.hdc refresh_secondary
chown root.root refresh_secondary
chmod a+r,u+wx,go-wx refresh_secondary
```

or for SCSI or SATA disks:

```
cd /usr/local/sbin
cp -a /usr2/fs/misc/refresh_secondary.sdb refresh_secondary
chown root.root refresh_secondary
chmod a+r,u+wx,go-wx refresh_secondary
```

Thereafter it can be executed by `root` (only) as a normal command. The contents of the script for IDE (PATA) disks, `refresh_secondary.hdc`, are:

```
#!/bin/sh
# for IDE (PATA) RAID arrays, adds a "stale" secondary disk into the array
if [[ -n `mdadm --detail /dev/md0 /dev/md2 | grep /dev/hdc` ]]; then
        echo "ERROR: \"Secondary\" disk /dev/hdc is already active !!"
        exit 1
elif [[ -n `mdadm --detail /dev/md1 | grep /dev/hdc5` ]]; then
    # Sometimes swap partitions don't detect staleness
        echo "NOTE: \"Secondary\" disk swap partition /dev/hdc5 was active"
    mdadm /dev/md1 -f /dev/hdc5
    mdadm /dev/md1 -r /dev/hdc5
fi
echo "Adding \"Secondary\" disk /dev/hdc to RAID array /dev/md0 ... "
mdadm /dev/md0 -a /dev/hdc1
while [[ -n `grep recovery /proc/mdstat` ]]; do
    recovery=`grep recovery /proc/mdstat | cut -c32-`
    echo -e -n '\r'$recovery
    sleep 1;
done
echo -e "\ndone."
echo "Installing GRUB bootloader on \"Secondary\" disk /dev/hdc ... "
/usr/sbin/grub --batch << EOT
device (hd0) /dev/hdc
root (hd0,0)
setup (hd0)
quit
EOT
echo "done."
echo "Adding \"Secondary\" disk /dev/hdc to RAID array /dev/md1 ... "
mdadm /dev/md1 -a /dev/hdc5
while [[ -n `grep recovery /proc/mdstat` ]]; do
    recovery=`grep recovery /proc/mdstat | cut -c32-`
    echo -e -n '\r'$recovery
    sleep 1;
done
echo -e "\ndone."
echo "Adding \"Secondary\" disk /dev/hdc to RAID array /dev/md2 ... "
echo "(This may safely be interrupted with 'Control-C' if you don't want to wait)"
mdadm /dev/md2 -a /dev/hdc6
while [[ -n `grep recovery /proc/mdstat` ]]; do
    recovery=`grep recovery /proc/mdstat | cut -c32-`
    echo -e -n '\r'$recovery
    sleep 1;
done
echo -e "\ndone."
```

The contents of the script for SCSI or SATA disks, `refresh_secondary.sdb`, are:

```
#!/bin/sh
# for IDE (SATA) or SCSI RAID arrays, adds a "stale" secondary disk into the array
if [[ -n `mdadm --detail /dev/md0 /dev/md2 | grep /dev/sdb` ]]; then
        echo "ERROR: \"Secondary\" disk /dev/sdb is already active !!"
        exit 1
elif [[ -n `mdadm --detail /dev/md1 | grep /dev/sdb5` ]]; then
    # Sometimes swap partitions don't detect staleness
        echo "NOTE: \"Secondary\" disk swap partition /dev/sdb5 was active"
    mdadm /dev/md1 -f /dev/sdb5
    mdadm /dev/md1 -r /dev/sdb5
fi
echo "Adding \"Secondary\" disk /dev/sdb to RAID array /dev/md0 ... "
mdadm /dev/md0 -a /dev/sdb1
while [[ -n `grep recovery /proc/mdstat` ]]; do
    recovery=`grep recovery /proc/mdstat | cut -c32-`
    echo -e -n '\r'$recovery
    sleep 1;
done
echo -e "\ndone."
echo "Installing GRUB bootloader on \"Secondary\" disk /dev/sdb ... "
/usr/sbin/grub --batch << EOT
device (hd0) /dev/sdb
root (hd0,0)
setup (hd0)
quit
EOT
echo "done."
echo "Adding \"Secondary\" disk /dev/sdb to RAID array /dev/md1 ... "
mdadm /dev/md1 -a /dev/sdb5
while [[ -n `grep recovery /proc/mdstat` ]]; do
    recovery=`grep recovery /proc/mdstat | cut -c32-`
    echo -e -n '\r'$recovery
    sleep 1;
done
echo -e "\ndone."
echo "Adding \"Secondary\" disk /dev/sdb to RAID array /dev/md2 ... "
echo "(This may safely be interrupted with 'Control-C' if you don't want to wait)"
mdadm /dev/md2 -a /dev/sdb6
while [[ -n `grep recovery /proc/mdstat` ]]; do
    recovery=`grep recovery /proc/mdstat | cut -c32-`
    echo -e -n '\r'$recovery
    sleep 1;
done
echo -e "\ndone."
```

mdstat

The following utility script `mdstat` can be used to check for mirroring activity. For FSL6, it provides easier to read output than the raw `cat /proc/mdstat` command. A copy of this script can be placed in the `/usr/local/bin` directory and set-up by using the following commands as `root`, FSL6 IDE (PATA) disks only:

```
cd /usr/local/bin
cp -a /usr2/fs/misc/mdstat.6 mdstat
chown root.root mdstat
chmod a+rx,u+w,go-w mdstat
```

or for FSL7 and FSL8:

```
cd /usr/local/bin
cp -a /usr2/fs/misc/mdstat.7 mdstat
chown root.root mdstat
chmod a+rx,u+w,go-w mdstat
```

Thereafter it can be executed by any user as a normal command. The contents of the script for FSL6 IDE (PATA) disks, `mdstat.6`, are (note the third and fourth lines below are continuations of the second):

```
#!/bin/sh
cat /proc/mdstat | sed -e
's/ide\/host0\/bus0\/target0\/lun0\/part/hda/' | sed -e
's/ide\/host0\/bus1\/target0\/lun0\/part/hdc/'
```

The contents of the script for FSL7 & FSL8, `mdstat.7`, are simply:

```
#!/bin/sh
cat /proc/mdstat
```

This version does nothing to improve the readability of the output, just reduces the typing involved slightly.

APPENDIX A. DISK RESET RECIPE

This is a recipe for resetting two bad disks in preparation for reloading them from an older spare. It will remove the difficulty of having to reboot the good disk enough times so that the RAID system thinks it is newer. This procedure can be modified to reset a single bad disk by changing step 1 to insert only that one disk in the Primary drive and changing step 5 to use only the first three `mdadm` commands.

Please note that the use of `mdam --zero-superblock` probably irreversibly deletes the RAID copy on the partition, so it should be used with only bad disks installed. Unfortunately using the FSL6 installer forces the use of the horrible (`devfs`) file names as shown, though fortunately tab-completion does help you with typing them in. However, you can use any version of the installer, FSL6, 7, or 8, regardless of what system you have installed on the hard disks. So using the FSL7 or FSL8 installer will allow you avoid the horrible file names.

1. Put the bad disk pair into both "Primary" and "Secondary" slots

2. Boot the Binary-1 DVD and proceed as you would for a standard install (instructions available at `ftp.hartrao.ac.za` in `/pub/fs6x/fs6linux_DVD.txt`, `/pub/fs7x/fs7linux_DVD.txt`, or `/pub/fs/8x/fs8linux_DVD.txt`), as appropriate, using anonymous `ftp`) up to the point where it prompts you to start entering the network configuration information, select `<Go Back>` to get to the main menu. Select `Detect disks`. When it prompts you to select the partitioning method, select `<Go Back>` to get to the main menu again.

3. Select `Execute a shell` and press `<Enter>`

4. Check to make sure no **md** devices are active and "stop" them if they are. To check, use the command:

   ```
   cat /proc/mdstat
   ```

   If any are active, issue the command:

   ```
   mdadm --stop /dev/mdX
   ```

   for each **md** device, where `X` is the number of the device.

5. For FSL7 or FSL8 installer, and SCSI or SATA disks, at the shell prompt enter:

   ```
   mdadm --zero-superblock /dev/sda1
   mdadm --zero-superblock /dev/sda5
   mdadm --zero-superblock /dev/sda6
   mdadm --zero-superblock /dev/sdb1
   mdadm --zero-superblock /dev/sdb5
   mdadm --zero-superblock /dev/sdb6
   ```

For FSL7 or FSL8 installer, and IDE disks, at the shell prompt enter:

```
mdadm --zero-superblock /dev/hda1
mdadm --zero-superblock /dev/hda5
mdadm --zero-superblock /dev/hda6
mdadm --zero-superblock /dev/hdb1
mdadm --zero-superblock /dev/hdb5
mdadm --zero-superblock /dev/hdb6
```

For FSL6 installer and IDE disks, at the shell prompt enter:

```
mdadm --zero-superblock /dev/ide/host0/bus0/target0/lun0/part1
mdadm --zero-superblock /dev/ide/host0/bus0/target0/lun0/part5
mdadm --zero-superblock /dev/ide/host0/bus0/target0/lun0/part6
mdadm --zero-superblock /dev/ide/host0/bus1/target0/lun0/part1
mdadm --zero-superblock /dev/ide/host0/bus1/target0/lun0/part5
mdadm --zero-superblock /dev/ide/host0/bus1/target0/lun0/part6
```

6. Then shutdown by entering:

```
shutdown -h now
```

APPENDIX B. TWO COMPUTER ROTATION

This section describes how to handle computer, and disk, rotation if you purchased two identical FS computers in order to have a higher level of reliability. This approach provides a larger pool of spare disks and a spare computer. For this discussion, the set of disks for one computer are designated the "Operational" disks; the disks for the other, the "Spare" disks. In this approach the "Operational" computer is whichever one is connected to the VLBI equipment. Normally, it would have the Operational disks installed. The Spare computer is the other one and is probably only connected to the network and not the VLBI equipment.

The basic idea of the two computer rotation scheme is that each computer, with its disks, is kept as a separate system except when the computers are swapped or a disk from the Spare set needs to pressed into duty in the Operational computer. The Spare computer has its own IP address and is kept on the network. This allows it to be exercised a bit while it is not in use as the Operational computer to make sure it is still working. The Spare disks are set-up so that they contain a complete working system, including the FS, as a "deep" back-up for the Operational system. The Spare computer should keep the same disk rotation pattern as the operational computer.

Periodically, about every six months and just after a successful disk rotation, the computers should be swapped, disks and all. The old Spare computer with the Spare disks is moved and connected to the VLBI equipment. It is tested briefly to verify that the system on Spare disks can still communicate successfully with the VLBI equipment. Afterwards the disks are swapped between machines so that the original Operational disks go into the machine (newly) connected to the VLBI equipment and the Spare disks go into the other machine, which is now the

"Spare". Swapping the computers in this way periodically provides a more rigorous proof that the Spare computer and disks are really good spares. There are three failure cases where this rotation scheme would be interrupted:

(1) One of the installed Operational disks fails. After pressing the Operational computer Shelf disk into service, some disks from the Spare computer can then be re-initialized and refreshed (follow the procedure in **APPENDIX B: DISK RESET RECIPE** and then use the procedure in the **REFRESHING A STALE SECONDARY** section) for use in the Operational computer.

(2) The Operational computer fails. In this case, the computers can be swapped as would occur at a normal computer rotation. If some, but not all, of the operational disks failed at the same time you would need to add in some the Spare disks, as in (1) above. Don't put the Spare disks in the failed computer since it might damage them.

(3) Both installed Operational disks fail. In the case, it is likely that there is some problem with the Operational computer. The safe course here would be to swap computers with their disks so that now you are using the (old) Spare Computer and its disks as the (new) Operational computer and disks. The (old) Operational computer can be debugged off-line. You should keep the (old) Operational Shelf disk safely on the shelf to use for restoring your system once the problem has solved. However in a pinch (the Spare disks are too out of date), the (old) Operational computer's Shelf disk could be used as the Primary disk in the (new) Operational computer and one of the (old) Spare computers disks re-initialized and "refreshed" as the Secondary disk (follow the procedure in **APPENDIX B: DISK RESET RECIPE** and then use the procedure in the **REFRESHING A STALE SECONDARY** section).

If a failure happens during an observation, you would not need to completely recover the system right away, just enough so that you can observe again. In case (1), the system should be safe to run on the remaining disk until the observation is over, at which point the recovery can be started.

When you have a failure, you should obtain any replacement hardware needed to reconstruct your full system.

In order to set-up this rotation scheme, you will need to pick one computer and its set of disks to be the operational ones. The choice can be arbitrary, maybe just the computer you start setting up first. Note that for the computer, "Operational" is only a temporary designation, since the chassises will get swapped from time-to-time. For the disks "Operational" is a more long-lived designation; in principle, it would never change unless all the Operational disks failed and you had to switch to the Spares.

Both the Operational and Spare computers should be fully checked out in the normal way (for example see `FSL8_PC_Checkout.txt`) with a complete set-up. This will verify that all aspects of the computer are working and properly labeled. Then both the Operational computer and spare computer should be set-up for operational use (see `FSL8_End_User.txt`); each

computer should have its own name and IP address, otherwise everything can be the same. As an alternative, some work can be saved by doing the full check-out and set-up on the Operational computer and then cloning the disk for use in the Spare computer as described in **APPENDIX D. CLONING DISKS**. However after the cloning, it is still important to perform the testing and labeling steps in the full check-out procedure for the Spare computer to verify that all components are working and properly labeled. Using the disk cloning procedure can help avoid subtle differences in the set-ups of the two computers and is the recommended approach, but requires additional steps and care.

Once the computers are set-up and both verified to work with the equipment and if the cloning procedure was not used, the network configuration in the disks on both computers will need to be modified to support the NICs on both computers. Please refer to **APPENDIX C. MULTIPLE COMPUTER NIC SUPPORT**.

After cloned disk testing or the NIC support modifications, the machines should now be ready for use as an Operational and a Spare computer. Please be sure to apply the weekly updates to both machines. In case you don't log into the spare computer very often to check its e-mail, please establish your own reminder system to make sure that the Spare gets updated when the primary. A good approach might to be use the Spare computer as a test-bed for the updates in order to not affect the Operational disks if something goes wrong.

If you want to be able to handle case (3) above more gracefully, you might consider re-copying the contents of the `/usr2` system to the Spare computer after each FS update on the Operational computer, so that the FS installation on the Spare disks doesn't get too out of date (you may not have all operating system modifications on the Spare disks that you have on the Operational disks unless you have been very meticulous in making any changes in parallel, but this is probably a minor concern; hopefully you will have been doing the weekly updates right along, which are very important). To copy the `/usr2` contents to the Spare disks, you can execute the following commands (as `root`) on the Spare computer once the update as been verified:

```
cd /
fuser -k -m /usr2
umount /usr2
mkfs.ext3 /dev/md2
mount /usr2
ssh root@operational "(cd /usr2; tar --one-file-system -cf - .)" | (cd /usr2; tar xpf -)
```

where “operational” is the node name (a fully qualified host name may be needed) or IP address of the Operational computer. These commands should be put in script, checked carefully, and only executed from the script because the step that re-initializes `/usr2` (`mkfs.ext3 /dev/md3`) can cause serious problems if there is an error in the form of the command. It would probably be a good idea if the script included a warning about the `/usr2` partition on the Spare computer being overwritten and requiring confirmation before proceeding. An example script is given in **APPENDIX E. SPARE COMPUTER REFRESH SCRIPT**. The first use of the script should be after a Spare computer disk rotation so that it would be possible to recover from a problem using the Shelf disk. Note you will need to have `ssh` connections enabled and the `/etc/hosts.allow` file on the Operational computer set-

up to allow the Spare computer to connect with `ssh`. Typically the latter is done by adding a line like:

```
sshd: spare
```

where `spare` is the node name (a fully qualified host name may be needed) or IP address of the Operational computer.

APPENDIX C. MULTIPLE COMPUTER NIC SUPPORT

This section describes how to modify the system set-up to allow one removable disk to be used in two computers that are the same except for the MAC address of the NIC. The result is that node name and IP address of the system will move with the disks. This is used in configuring the two computer rotation described in **APPENDIX B**, but can also be useful for other systems where a disk may be used in more than one identical computer. Please note that FSL6 does not require this step. The following procedure can be done in parallel if you already have separate RAIDs that boot on each computer.

Start by examining the file `/etc/udev/rules.d/70-persistent-net.rules` (in FSL8, for FSL7, the file is `z25_persistent-net.rules`) on the disks (possibly a single non-RAID disk, but referred to here as plural because for a RAID, the complete active RAID array should be kept together when disks are swapped) that you want to use in the other computer. Please note the MAC address (of the form `xx:xx:xx:xx:xx:xx`, where the `x`'s are hex digits) of the interface you are using, typically `eth0`. Edit the file as `root` and duplicate the line for the interface you are using. On the duplicated line, leave everything else as it is, but change the MAC address to that of the one on the other machine (possible found by examining the same file on disks that already boot in the other machine or examining the NIC), then save the file. Now shutdown the machine and swap the disks to other machine and boot it. If everything was done correctly, the other machine should have access to the network with the same IP address and node as used before on the original computer. Once this is verified, you can switch the disks back to their original machine.

APPENDIX D. CLONING DISKS

The section describes how to set-up a pair of disks for use in a second computer by cloning one of the disks already in use in another computer, referred to here as the original computer. This is only useful if you have multiple computers with identical hardware. The key points are that after the actual cloning: (1) the UUID of the disks for the second computer array must be updated to avoid possible conflicts if disks from the two computers are ever accidently mixed, and (2) the IP and node information for the second computer need to be modified so there is no address conflict on the network.

1. Initialize two new disks (to be used for the clone) using the procedure given in the **REPLACEMENT DISKS** section and the second set of hardware.

2. On the original machine, boot with one of its RAID pair in the "Primary" slot and one of the new disks in the "Secondary" slot and "refresh" it using the procedure in the **REFRESHING A STALE SECONDARY DISK**.

3. Once the RAIDs have been refreshed, install grub on the secondary disk (`refresh_secondary` did this automatically if you used it, but see `fs8linux_DVD.txt`, Second Stage, Step 12 for manual steps), shutdown the system, move the new disk to the second machine Primary slot and the other new disk in the Secondary slot, and boot the installer in `rescue` mode (under `Advanced Options`).

4. At the point where you are asked to select a root filesystem, back out to the main menu and start a shell.

5. Check the RAID setup using `cat /proc/mdstat` and then use:

   ```
   mdadm --stop /dev/md0
   ```

   etc. to stop each RAID in turn.

6. Start up each RAID manually, asking that a new UUID be allocated using:

   ```
   mdadm --assemble /dev/md0 --update=uuid /dev/hda1
   mdadm --assemble /dev/md1 --update=uuid /dev/hda5
   mdadm --assemble /dev/md2 --update=uuid /dev/hda6
   ```

   etc. (or `/dev/sda1`, etc as appropriate)

7. Exit the shell using:

   ```
   exit
   ```

8. In the menu select Detect disks.

9. Again at the point where you are asked to select a root filesystem, back out to the main menu and start a shell.

10. Now add the blank disk partitions to the appropriate RAIDs using:

```
mdadm /dev/md0 -a /dev/hdc1
mdadm /dev/md1 -a /dev/hdc5
mdadm /dev/md2 -a /dev/hdc6
```

    etc. (or `/dev/sdb1`, etc. as appropriate)

11. Once all the RAIDs have synchronized (check on this using `cat /proc/mdstat`), exit the shell and from the main menu start rescue mode again and select `/dev/md0` as the root filesystem.

12. Start a shell in the environment of the root filesystem and update the RAID configuration stored there using:

```
mv /etc/mdadm/mdadm.conf /etc/mdadm/mdadm.conf.old
/usr/share/mdadm/mkconf > /etc/mdadm/mdadm.conf
update-initramfs -u -k all
```

13. Exit the shell, disconnect the network cable, and reboot the system to the hard disk.

14. Install `grub` on the second disk

15. Change the MAC address in the file that holds it as described in **APPENDIX C** above (as `root`). Note that unless you plan to use the resulting disk in both the original machine and the second machine (such as described in **APPENDIX B**), you don't need to preserve the original MAC address of the original machine and can just replace the MAC address of the interface you are using with the MAC address in the second machine.

16. Update the configuration for the new node name as `root`:

    A. Update the network configuration on the second computer to use the node name and IP address that you want (or need) it to have:

```
cd /root/fsadapt
./netconfig
```

    answering the questions asked by the `netconfig` script.

    B. If you changed domain name of the system (to the right of the dot after the nodename) and the `/etc/hesiod.conf` exists on your system, you should update the `rhs=` line in that file.

    C. Reboot.

D. If you have modified the user names in `/etc/passwd` file on the original computer to reflect the name of the system (for `root` and `prog` as opposed to the site name (for `oper`), you may want to update the user names on the second computer to reflect its name.

E. Update the mail configuration using:

```
dpkg-reconfigure exim4-config
```

If you added re-write rules on the original system, you should update them on this system for the new node name. Please check `FSL8_End_User.txt` step 20 for details.

F. Generate new `ssh` keys, typically:

```
cd /etc/ssh
rm *key *key.pub
dpkg-reconfigure openssh-server
```

but see `FSL8_End_User.txt` step 13 for more options.

G. For completeness, you can update the `emacs` configuration:

```
dpkg-reconfigure emacs22
```

H. If there are any other places you made customizations that depend on the node or domain name, you may want to update them as well. For the node name, you might, for example, try:

```
rgrep old_node_name /etc
```

and change all appropriate instances into the `new_node_name`. You could do something similar for the domain name if you changed it.

17. Connect the network cable and reboot to make sure the system is running normally and works on the network.

18. Once it is confirmed that the second computer is operating normally, you can prepare a third disk for it using the procedure in the **REPLACEMENT DISKS** section and there after follow the usual disk rotation.

19. Be sure to add the normal second disk back into the original computer and “refresh” it.

APPENDIX E. SPARE COMPUTER REFRESH SCRIPT

The section gives an example spare computer refresh script, `refresh_spare_usr2`, for stations with a spare computer as described in **APPENDIX B**. Normally the example would reside in the `/usr/fs/misc` directory, but if not available in your FS version, you can obtain a copy from Ed (Ed.Himwich@nasa.gov).

The script must only be executed on the spare computer because it erases the `/usr2` partition. Please note that it will stop any processes that are using `/usr2`. If there are such processes, you should reboot when the script finishes to restart them. Please make sure all otrher users are logged before using this script. You must be logged in as root, do not use ssh or su or become root locally. Logging-in remotely as root is okay.

The script, named `refresh_spare_usr2`, is listed below. Installation instructions are included in the script toward the end.

```
#!/bin/sh
echo "*********************************************************************"
echo "*  This is a script to refresh the contents of the spare computer *"
echo "*    '/usr2' partition if you have both an Operational and Spare  *"
echo "*    computer. Please refer to 'usr2/fs/misc/RAID.txt' for more   *"
echo "*    information.                                                 *"
echo "*********************************************************************"
echo ""
echo "CAUTION: This script will automatically overwrite everything on the"
echo "         /usr2 partition of the computer it is run on.  Use it only"
echo "         on the Spare computer.  It will stop any processes that"
echo "         are using the /usr2 partition.  Rebooting afterwards is"
echo "         recommended."
echo ""
echo "This script must be executed by 'root' on the Spare computer. "
echo "You must be logged-in as 'root', do not use 'su' or 'ssh' to"
echo "become 'root' locally. Remote login via 'ssh' as 'root' is okay."
echo "Make sure all other users are logged-out before proceeding."
echo ""
echo -e "Do you wish to continue with the REFRESH? (y=yes, n=no) : \c"
badans=true
while [ "$badans" = "true" ]
do
  read ans
  case "$ans" in
    y|yes) echo -e "\nO.K. Continuing with refresh ... "
           badans=false
           ;;
    n|no)  echo -e "\nO.K. Exiting."
           exit
           ;;
    *)     echo -e "\nPlease answer with y=yes or n=no : \c"
  esac
done
#
echo "This script must be customized before first use, and it hasn't been."
echo "See script for details."
#To customize this script (as 'root') on the Spare computer:
#   1. Execute the commands:
#        cd /usr/local/sbin
#        cp -a /usr2/fs/misc/refresh_spare_usr2 refresh_spare_usr2
```

```
#        chown root.root refresh_spare_usr2
#        chmod a+r,u+wx,go-wx refresh_spare_usr2
#   2. Edit '/usr/local/sbin/refresh_spare_usr2':
#        A. comment out the two echo commands above (add leading '#'s).
#        B  change the 'operational' on the ssh command below to the node name
#           (a fully qualified hostname may be needed) or IP address of
#           the operational machine
#        C. uncomment (delete the leading '#'s) the 15 commands
#            below (#echo "Refreshing ..." ... #echo "Done. ...").
#        D. Save the results
#   3. Test it the first time after a Spare computer disk rotation.
#echo "Refreshing spare computer disk /usr2 partition ..."
#fuser -k -m /usr2
#umount /usr2
#mkfs.ext3 /dev/md2
#mount /usr2
#echo ""
#echo "You may be prompted for your operational machine's root password, then"
#echo "there will be a long pause without any output unless there are errors."
#echo "If there are errors, please note them and report them to Ed.Himwich@nasa.gov."
#echo "The refresh is finished when you see a line that starts: 'Done. ...',"
#echo "and get a prompt. Please follow the directions on the 'Done. ...' line."
#echo ""
#ssh root@operational "(cd /usr2; tar --one-file-system -cf - .)" | (cd /usr2; tar xpf -)
#echo ""
#echo "Done. Wait until 'mdstat' does not show an active recovery and then reboot."
```