

RAID Usage with FS Linux 9

Ed Himwich and Jonathan Quick

Table of Contents

INTRODUCTION.....	2
OVERVIEW.....	2
NORMAL OPERATION	4
RAID STATUS	5
DISK ROTATION.....	6
REFRESHING A STALE SECONDARY DISK.....	6
IF A DISK FALLS OUT OF THE RAID	7
RECOVERING FROM A DISK FAILURE	8
RECOVERING A LOST FILE OR FILES.....	8
APPLYING AN UPDATE	8
RECOVERING FROM A STALE SHELF DISK	9
REPLACEMENT DISKS	9
SCRIPTS	10
refresh_secondary.....	10
mdstat.....	11
APPENDIX A. DISK RESET RECIPE.....	11
APPENDIX B. TWO COMPUTER ROTATION.....	12
APPENDIX C. MULTIPLE COMPUTER NIC SUPPORT	15
APPENDIX D. CLONING DISKS	16
APPENDIX E. SPARE COMPUTER REFRESH SCRIPT.....	18

INTRODUCTION

This document, “FSL9_RAID”, describes how to use and maintain the software RAID system that is installed as standard in FS Linux 9. For RAID usage under FS Linux versions 6-8, please see the “RAID” document. 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 in a normal state, 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 an inactive shelf disk. 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 set-up of 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 for use in another computer. This appendix is referred to from **APPENDIX B**. The **APPENDIX E. SPARE COMPUTER REFRESH SCRIPT** section gives the set-up for 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

The standard disk configuration for an FSL9 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 at least 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 for three disk stations, the update is applied to only a single disk initially to allow a relatively easy recovery if one is necessary.

There are two basic partition schemes: DOS and GPT (GUID (Globally Unique Identifier) Partition Tables). All disks larger than 2 TB use GPT; smaller disks *generally* use DOS Tables. The type of partition table determines the device names used in Linux. In the rest of the document “DOS disks” and “GPT disks” are used to distinguish the type of partition table and hence the devices names of the partitions. The RAID devices, device names, and file systems for the different disk partitions schemes is given in the following table. The leading /dev is omitted from the device names.

RAID Devices	DOS Disk Devices	GPT Disk Devices	File Systems
md0	sda1, sdb1	sda1, sdb1	/
md1	sda5, sdb5	sda2, sdb2	(swap area)
md2	sda6, sdb6	sda3, sdb3	/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 “recovering” or being “refreshed”.

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 accidentally 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 (use `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. The standard set-up is for e-mail sent to `root` to be redirected to another user that logs 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 advisable to “hot swap” disks, even if in principle the controller chipset claims to support it.

NORMAL OPERATION

Normal operation requires two removable hard drives using device names `/dev/sda` for the first, typically in a slot labelled “Primary” (except for SCSI disks), and `/dev/sdb` for the second, typically labeled “Secondary” (except for SCSI disks). For systems with SATA disks,

the Primary disk is typically connected to the lowest numbered SATA interface; the Secondary to the next lowest numbered SATA interface. For IDE (PATA) disks, the Primary disk is on the master of primary IDE interface; the Secondary is on the master of the secondary IDE interface. For SCSI disks the Primary disk has the lowest SCSI ID; the Secondary, the next lowest. Thus for SCSI disks, the disk designation does not typically 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

If `mdstat` script has been installed on your machine, you can check on the RAID mirroring status by using the command:

```
mdstat
```

(The setup of this script is described in the [SCRIPTS](#) section later in this document.) Look 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.

Normally the `mdstat` output will show the string `[2/2]` in the listing of each MD device's status. This indicates that both disks 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 be investigated. The section [IF A DISK FALLS OUT OF THE RAID](#) contains suggestions on what to do if this situation occurs.

If the RAID was running normally and one disk has failed, you can detect which from the `mdstat` output, which will show “(F)” next to the partitions that have failed. However, if only

one disk was detected at boot time, it is necessary to observe the disk activity lights to determine which one it is.

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 (e.g., by changing 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. If the `refresh_secondary` script has been installed on your machine, you can do this with the command:

```
refresh_secondary
```

The setup of this script (and `mdstat`, used in this section also) is described in the **SCRIPTS** section later in this document. This script re-adds the partitions from the stale disk into the arrays and installs the boot strap loader. It prints status messages on the progress of the refresh. 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 while a recovery is underway. The status of the mirroring can be checked with:

```
mdstat
```

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

The RAID recovery process should ideally be allowed to run to completion before rebooting, and certainly before removing either of the disks.

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 be investigated. If this is the first time this has happened and the same disk has fallen out of all **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 partitions from different disks, the recovery will be more complicated, please contact Ed (Ed.Himwich@nasa.gov) for advice on how to deal with this situation.

If partitions from only one disk are missing from the **MD** devices that are missing partitions, you can attempt to add that disk back into the array at the next convenient opportunity. The approach used here is to always add the secondary disk back into the array. 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 as described above. 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 be 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. In any event, 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 accidentally 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 most recent 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** on the bad disks. 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 the procedure in **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 at least 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 netinst CD for your system according to the directions in `fsl9_DVD.txt`. You can skip the network set-up by selecting “Back” and the “Detect Disk”, which will take you directly to step 10. In steps 10-12 follow the single disk instructions, After setting up the partitions for RAID usage, you can stop at the point of creating the **MD** devices near the start of step 13 of the First Stage, Alternatively, if the disk has previously been set-up for use in a RAID using the `fsl9_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.

SCRIPTS

Two utility scripts are described here. The `refresh_secondary` script is used by the `root` user when adding a stale secondary 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 utility script `refresh_secondary` can be used by `root` to automate adding the partitions of a stale secondary into the RAID arrays for refreshing and installing the boot strap loader on the secondary. Different versions are needed for systems using DOS disks and GPT disks. Examples can be found in the `/usr2/fs/misc/` directory as `refresh_secondary.sd156` and `refresh_secondary.sd234` 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 DOS disks use:

```
cd /usr/local/sbin
cp -a /usr2/fs/misc/refresh_secondary.sd156 refresh_secondary
chown root.root refresh_secondary
chmod a+r,u+wx,go-wx refresh_secondary
```

For GPT disks use:

```
cd /usr/local/sbin
cp -a /usr2/fs/misc/refresh_secondary.sd234 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.

mdstat

The utility script `mdstat` can be used to check for mirroring activity. A copy of this script can be placed in the `/usr/local/bin` directory and set-up by using the following commands as root:

```
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.

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 for the appropriate type of disk (if step 6 is optionally used, only the first `dd` command for the appropriate type of disk should be used).

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.

1. Put the bad disk pair into both “Primary” and “Secondary” slots
2. Boot the `netinst` CD and proceed as you would for a standard install (instructions available at `ftp://ftp.hartrao.ac.za/pub/fs9x/fsl9_DVD.txt` 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 active **md** device, where X is the number of the device.

5. For DOS 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 GPT disks, at the shell prompt enter:

```
mdadm --zero-superblock /dev/sda2
mdadm --zero-superblock /dev/sda3
mdadm --zero-superblock /dev/sda4
mdadm --zero-superblock /dev/sdb2
mdadm --zero-superblock /dev/sdb3
mdadm --zero-superblock /dev/sdb4
```

6. For DOS disks, if you optionally want to delete the boot-strap loader from the MBRs, without affecting the partition table, use:

```
dd if=/dev/zero of=/dev/sda bs=446 count=1
dd if=/dev/zero of=/dev/sdb bs=446 count=1
```

If you wish to delete the entire MBR including the partition table, increase the block size (bs) to 512. If you do, the disk will have to be repartitioned before further use.

7. 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 actually used to control the equipment, which for non-Ethernet devices would usually imply having direct physical connections to the 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 be 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 should be moved and any needed direct connections to the equipment established. It should be 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, a disk 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. A new disk should be obtained for the Spare 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 chassis 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 `FSL9_Upgrade.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 `FSL9_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 both computers are set-up and 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 Operational and Spare computers. 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 is updated. A good approach might be to 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 has been verified:

```
cd /
fuser -k -m /usr2
umount /usr2
mkfs.ext4 /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.ext4 /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 Spare 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. 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` 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 `xs` 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 (possibly found by examining the same file on disks that already booted 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 accidentally 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. 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
mdadm --stop /dev/md1
mdadm --stop /dev/md2
```

as appropriate to stop each active RAID in turn.

FSL9 RAID Usage

6. Start up each RAID manually, asking that a new UUID be allocated. For DOS disks use:

```
mdadm --assemble /dev/md0 --update=uuid /dev/sda1
mdadm --assemble /dev/md1 --update=uuid /dev/sda5
mdadm --assemble /dev/md2 --update=uuid /dev/sda6
```

For GPT disks use:

```
mdadm --assemble /dev/md0 --update=uuid /dev/sda2
mdadm --assemble /dev/md1 --update=uuid /dev/sda3
mdadm --assemble /dev/md2 --update=uuid /dev/sda4
```

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 RAID. For DOS disks use:

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

For GPT disks, use:

```
mdadm /dev/md0 -a /dev/sdb2
mdadm /dev/md1 -a /dev/sdb3
mdadm /dev/md2 -a /dev/sdb4
```

11. Once all the RAID's 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 these three commands:

```
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 according.

```
grub-install /dev/sdb
```

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 using the directions in step 9 of `FSL9_End_User.txt`.

- B. Generate new ssh keys, typically:

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

but see `FSL9_End_User.txt` step 11 for more options.

- C. 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 `FSL9_End_User.txt` step 14 for details.

17. 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.

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

APPENDIX E. SPARE COMPUTER REFRESH SCRIPT

The section describes use of the example spare computer refresh script, `refresh_spare_usr2`, for stations with a spare computer as described in [APPENDIX B](#).

FSL9 RAID Usage

Normally the example would be available 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 other users are logged out before using this script. You must be logged in as root, do not use `ssh` or `su` to become root locally. Logging-in remotely as root is okay.

The Installation instructions are included in the script toward the end of the example script and are listed here as well. You should only install this script on a “spare” computer (as “root”):

1. Make sure you are working on the “spare” computer for a station.

2. Execute the commands:

```
cd /usr/local/sbin
cp -a /usr2/fs/misc/refresh_spare_usr2.9 refresh_spare_usr2
chown root.root refresh_spare_usr2
chmod a+r,u+wx,go-wx refresh_spare_usr2
```

3. Edit “`/usr/local/sbin/refresh_spare_usr2`”:

- a. Comment out the two `echo` commands (add leading “#”s) about the script needing to be customized.
- 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
from `#echo "Refreshing ..."` to `#echo "Done. ..."`
- d. Save the results

4. Test it the first time after a Spare computer disk rotation.