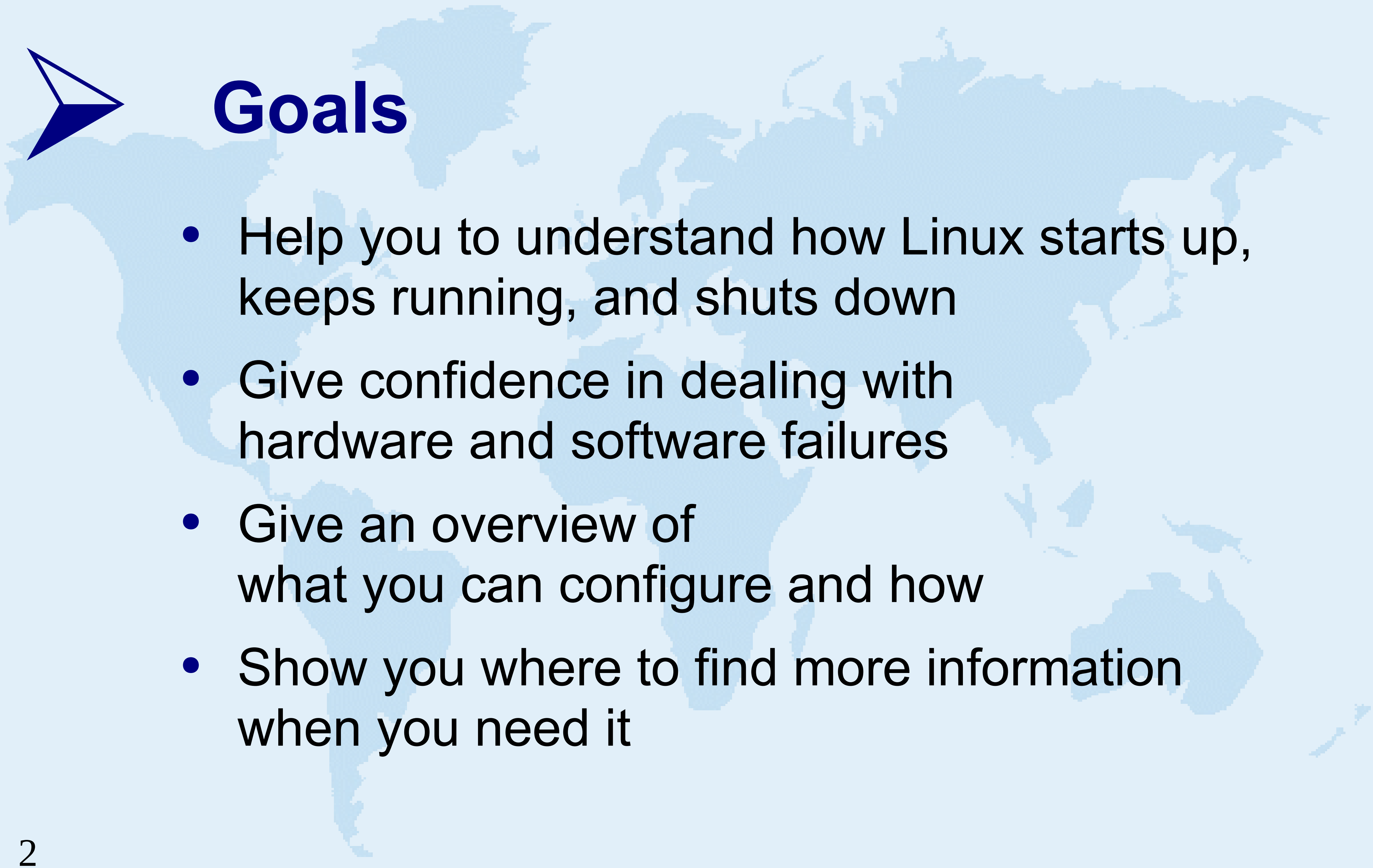
Linux System Administration

Jonathan Quick, Hartebeesthoek Radio Astronomy Observatory

Linux Startup and Shutdown
Managing Hard Disks, Partitions, Backups
Rescueing a Failing PC / System
Modifying Configuration
Adding/Removing Packages



Goals

- Help you to understand how Linux starts up, keeps running, and shuts down
- Give confidence in dealing with hardware and software failures
- Give an overview of
  what you can configure and how
- Show you where to find more information when you need it

Basic Linux Concepts

- Linux Kernel
  - Base monolithic kernel + loadable modules
  - Gives standardized access to underlying hardware
- Linux System / "Distribution"
  - Kernel + lots of software
  - Adds both system and application level software to the system
- Background processes ("daemons")

Logging in as 'root'

- In order to do any system-wide changes you usually have to be logged in as 'root'
- You can change to a virtual console (Ctrl-Alt-F1) and login normally or use 'su -'
- 'root' can override all permissions, start and stop anything, erase hard drives,...
  - So please be careful with disk names and similar!
  - You can browse and check many (if not most of the) things as a normal user (like 'oper').

Getting System Information

- ps axf, top; kill, kill -9
- free
- df, mount
- netstat -an, ifconfig, route -n
- w, who
- cat /proc/cpuinfo (and others)

Linux PC-Level Startup

- PC ROM BIOS initializes hardware and boots a Master Boot Record (MBR)
  - From a floppy, hard disk, CD-ROM, ...
- That MBR contains LILO, the Linux Loader
  - Shows LILO prompt, uses BIOS disk routines to load Linux kernel into memory
- Linux kernel starts, checks hardware
- Kernel attempts to locate the "root partition"
  - This becomes the '/' root file system

Linux LILO Prompt

- To get the prompt, keep any of Shift / Ctrl / Alt keys down when the word "LILO" appears
- 'LILO boot:' (TAB to see choices)
  - linux single
  - linux root=/dev/sdb1
- Boot floppies present the same LILO prompt
  - rescue root=/dev/sda1

Recovering from LILO Failures

- LILO depends on the kernel file remaining at the same physical CHS addresses on disk
  - Copying the file, restoring a backup, changing BIOS setup (CHS) can disrupt this and LILO boot fails
- You need to load and start a kernel from a boot floppy (or a bootable CD-ROM)
  - LILO boot: rescue root=/dev/sda1or root=/dev/hda1
- Reinstall LILO (as 'root'):
  - lilo

Linux Kernel-Level Startup

- Once '/' has been mounted (read-only), the kernel starts '/sbin/init'
  - As process #1, the "grandmother" of all processes
- The 'init' process follows instructions in '/etc/inittab' (please see 'man 5 inittab')
- The main start-up script '/etc/init.d/rcS' is run; it merely:
  - Runs the start-up scripts in '/etc/rcS.d' in alphabetical order

Startup Scripts

- '/etc/rcS.d' actually contains only symbolic links to "real" scripts in '/etc/init.d'
- These "System V"-style symbolic links are automatically updated with 'update-rc.d' (see man)
- Symbolic links are used to enforce the desired execution order with 'Snn' prefixes
  - For example: /etc/rcS.d/S05keymaps.sh -> ../init.d/keymaps.sh

What Happens Early in Startup?

- The following script files will be executed:

```
eclipse:~> ls -1 /etc/rcS.d/
README
S05keymaps.sh
S10checkroot.sh
S15isapnp
S20modutils
S25mdutils
S30checkfs.sh
S35mountall.sh
S40hostname.sh
S40network
S45mountnfs.sh
S50hwclock.sh
S55bootmisc.sh
S55urandom
```

Note -1, one column, not -l

Instructions!

S10checkroot.sh 'fsck' Checks of File Systems

- If the system was not properly shut down, these scripts will attempt an automatic 'fsck'
- If repairs would require deleting something:
  - "fsck failed. Please repair manually and reboot."
  - Enter the 'root' password when asked to
  - fsck /dev/sda1
    - Answer 'y' to repair questions; Ctrl-C if hundreds
  - Exit with 'exit' or Ctrl-D and let the system reboot

Runlevels

- After executing the 'rcS.d' scripts, 'init' "changes runlevels" to the default level '2'
- Runlevel conventions:
  - Runlevel 0 is halt.
  - Runlevel 1 is single-user.
  - Runlevels 2-5 are multi-user.
  - Runlevel 6 is reboot.
    - By default 2=3=4=5 starts up the same processes.
    - By default 0=6 stops the same processes.

Runlevel Directories

- /etc/rcX.d
  - Where 'X' is replaced by 0123456S
- These directories only have symbolic links to '/etc/init.d' where the real scripts are located
  - Links managed by 'update-rc.d'
- 'init' uses a single script '/etc/init.d/rc X' to change to runlevel X
  - '/etc/init.d/rc X' first runs 'K*' "kill" scripts (/w 'stop') and then 'S*' "startup" scripts (/w 'start') of the new runlevel directory '/etc/rcX.d'

The Boot Continues...

- Going to standard multi-user runlevel '2'

```
eclipse:~> ls /etc/rc2.d
S10sysklogd     S20iplogger S20xfs          S89atd
S10watchdog     S20logoutd  S20xntp3        S89cron
S12kerneld      S20lpd      S25netstd_nfs   S99rmnologin
S15netstd_init  S20mon      S30netstd_misc  S99xdm
S18netbase      S20plan     S50junkbuster
S20anacron      S20ppp      S50tleds
S20gpm          S20ssh      S50wu-ftpd
```

Shutting Down Linux

- The startup process is reversed
- The reversed order is shown in "kill" scripts of '/etc/rc0.d' (for halt) or '/etc/rc6.d' (for reboot)
- The final steps are performed by 'S*' "start" scripts in those directories

```
eclipse:~> ls /etc/rc0.d
K01xdm          K20logoutd      K25netstd_nfs   S20sendsigs
K11cron         K20lpd          K30netstd_misc  S30urandom
K12kerneld      K20mon          K50junkbuster   S40umountfs
K15netstd_init  K20plan         K50tleds        S50mdutils
K18netbase      K20ppp          K50wu-ftpd      S90halt
K20anacron      K20ssh          K80watchdog
K20gpm          K20xfs          K89atd
K20iplogger     K20xntp3        K90sysklogd
```

Adding Something to Startup

- Put a new script into '/etc/init.d'
  - Use '/etc/init.d/skeleton' as a template
  - The script must support 'start' and 'stop'
- Add the symbolic links into '/etc/rc?.d' directories with 'update-rc.d'
  - update-rc.d newscript defaults
  - update-rc.d newscript defaults 95 05
    - late start, early stop

Configuration Files Affecting Startup - A Summary

- /etc/inittab
  - which runs first /etc/init.d/rcS which runs (using /etc/init.d/rc script):
  - /etc/rcS.d scripts and then
  - /etc/rc2.d scripts
- The real scripts referenced from the previous directories are really in '/etc/init.d'
  - Manually starting/stopping something:
    /etc/init.d/something start
    /etc/init.d/something stop

Periodical Jobs with Cron

- The 'cron' daemon runs in the background with 1 minute resolution, starting timed script jobs
- Debian's configuration files
    - /etc/cron.d
        - Precisely timed jobs
        - Special file format
    - /etc/cron.daily, /etc/cron.weekly, /etc/cron.monthly
        - Plain shell scripts
        - For periodical chores (like deleting old log files)

```
# Run queue every 30 minutes
08,38 *      * * *      mail   if [ -x /
usr/sbin/exim -a -f /etc/exim.conf ];
then /usr/sbin/exim -q >/dev/null 2>&1;
fi
```

Network Configuration

- /etc/init.d/network
  - Starts up the network interfaces with the correct IP addresses (ifconfig) and route commands
- /etc/hostname, /etc/defaultdomain, /etc/hosts
  - Has the name and IP address of the computer itself
- /etc/resolv.conf
  - Has the IP addresses of DNS name server(s)
- /etc/network/interfaces
  - The details of all available network interfaces

Network Protection with "tcp-wrappers"

- During boot, the "Internet super daemon" 'inetd' is started
- /etc/inetd.conf lists the services (TCP/UDP port numbers) 'inetd' will listen to (see 'fsadapt')
  - When a connection from the outside is made, 'inetd' runs the command listed in 'inetd.conf' to respond
  - For almost all services, this is the 'tcpd' wrapper which:
    - First checks restrictions
    - If allowed, starts the real service executable

/etc/hosts.allow and /etc/hosts.deny

- Have quite complex syntax (see 'man 5 hosts_access' for details)
- Effective only for entries with 'tcpd' in /etc/inetd.conf
  - Plus a couple of stand-alone server programs into which there is special support coded in
  - For example the X server doesn't obey these!

Executable names!

```
/etc/hosts.deny:
  ALL: ALL

/etc/hosts.allow:
  ALL: .foobar.edu EXCEPT terminal.foobar.edu
```

Hard Disks

- The whole disk
  - /dev/hda (IDE), /dev/sda (SCSI)
  - /dev/hdb, c, d etc.
- The primary partitions
  - Each disk can have up to 4 primary partitions
    - One of which can be an extended partition
  - /dev/hda1, 2, 3, 4
- A partition is a contiguous part of the whole disk ("a smaller disk")

Hard Disk Partitions

- An extended partition holds up to 16 "logical drives"
  - /dev/hda5, 6, 7...
  - Was invented to overcome the limitation of only 4 primary partitions
- Use 'fdisk' or 'cfdisk' to manipulate partitions
- Changing partitions usually DESTROYS all the data on the disk!

Why Partitions?

- To separate user files and system files
  - As done in FS PCs
- To have different systems (like Windows and Linux) on the same disk
- To have boot files accessible to older BIOS's by keeping them below the 1024 cylinders boundary

The Root Partition

- The partition mounted as '/' by the kernel
  - LILO boot parameter can change this
  - Hard encoded into the kernel ('man 8 rdev')
- Other partitions are mounted as listed in the '/etc/fstab' file (found in that '/' partition)

Root? Partition? File system?

- There are two different things in Unix/Linux customarily referred to as "root":
  - The superuser 'root' with all privileges
  - The "root" partition in which the "root" file system resides; this is used as '/'
- Hard disk drives can be split into chunks called partitions
- Partitions can be formatted i.e. a file system is created in a partition

Different File System Types

- Linux has extensive support for "foreign" file system types
- The current "native" format for Linux is the "Second Extended Filesystem" 'ext2' or 'ext3'
- MS-DOS/Windows floppies and FAT partitions can be used as 'vfat'
  - Supports long file names and FAT32
- Network File System 'nfs', Windows 'ntfs', others...

Formatting, Mounting

- Formatting (erases all data!)
  - Hard disk partitions: mke2fs /dev/hda1
- Mounting to a mount point (=directory)
  - mount /dev/hda3 /mnt
- Use normally, unmount with 'umount /mnt'
- Permanent mounts in /etc/fstab , 'mount -a'

```
/dev/sda3  /usr2  ext2  defaults  0  2 <------- fsck #

mount -t ext2 -o defaults /dev/sda3 /mnt
```

Managing Mounted File Systems

- To see what partitions are mounted:
  - mount
    - Displays information from /etc/mtab
    - More convenient to use 'df' to also display unused space
- Mount points are normal directories
  - Mounting "hides" the old directory contents
- Unmounting is necessary before:
  - fsck, mke2fs, fdisk, tune2fs
    - These directly alter file system structures!

The Root Directory Level

- /boot -- boot files
- /dev -- device special files
  - Map to major/minor numbers --> kernel drivers
- /etc -- configuration files (usually read-only)
- /mnt, /floppy, /dosa -- temporary mount points
- /proc -- process information (virtual file system)
- /root -- home directory of 'root' user

/usr, /var

- /usr/doc -- documentation
  - /usr/doc/HOWTO -- "cookbook" instructions
- /var -- "variable", run-time files
  - /var/log -- run-time log files
  - /var/spool -- queued files (e.g. Printer)
  - /var/mail -- mailboxes
  - /var/lock, run, tmp
  - /var/lib/dpkg/info -- status of installed software

File and Directory Protection

- All files and directories are owned by one user and one group (-> UID, GID from /etc/passwd)
- All files and directories have three sets of protection "bits" "ugo=rwx"
- Files marked with 'x' are checked for a special starting sequence:
  - #! /bin/interpreter
  - If found, the '/bin/interpreter' is run with the file as its standard input; this is how scripts work

Special Protection Bits

- u=s -- set UID when run, "setuid bit"
  - No effect on directories or scripts
- g=s -- set GID when run, "setgid bit"
  - For directories, put files created in the directory into the same group as the directory, no matter what group the user who creates them is in
- =t -- "save program text on swap device"
  - For directories, prevent users from removing files that they do not own in the directory

Finding Files with Suspicious Protections

- find / \(-type f -o -type d \) -perm +o=w
  - See 'man find'!
- Some files and directories (such as /tmp) need to be writable by all users
- See also /var/log/setuid.changes

Inodes

- The file descriptors (information about allocated disk blocks) are not stored in directories
- Instead, all files have "file numbers"
  - Directories just refer to the "real" file with this "inode number"
- If not properly closed at shutdown, 'fsck' will check the directory references
- Every mounted 'ext2' file system has a 'lost+found' directory for fsck-recovered files

Backup Operations

- All files of a working Linux system are "regular" and can be copied onto another disk as backup
- The only "magic" lies with LILO and the kernel file location (remember to rerun 'lilo')
- Current recommended FS backup scripts are based on 'tar'

Universal 'tar' archiver

- 'tar' -- "tape" archiver
  - Not strictly tape anymore!
- Can take complete directory trees and combine them (with all files) into a single archive file ('x.tar')
  - The archive file can be on tape, on floppies, on another disk, or it can be a pipe ('|') to another program
- Creating archives 'tar cvlzf x.tar.gz .'
- Listing the contents 'tar tvzf x.tar.gz'
- Extracting files 'tar xvzpf x.tar.gz'

The 'tar' Magic Letters

- 'tar c/t/x' -- the first command letter means:
  - c -- create a new archive
  - t -- table of contents of existing archive
  - x -- extract files from an existing archive
- The additional optional letters 'vlpzf':
  - v -- verbose, list file names as we go
  - l -- same file system, p -- retain protections
  - z -- compress/uncompress the archive with 'gzip'
  - f -- give archive file name (default tape...)

'tar'-based Copying/Moving

- For making faithful replications (or complete moves) of directory structures
  - (cd /loc1; tar cf - dir1 dir2) | (cd /loc2; tar xvpf -)
- A temporary archive file is never actually creat-ed, it streams in the pipe ('|')
- Combine with 'ssh' to make remote copies:
  - ssh root@remote 'tar clf - /' | (cd /bks; tar xvpf -)
- Should use 'backup' script from 'fsadapt' to make FS backups to avoid mistakes

Rescuing a Failed System

- We have covered rescue booting with a boot floppy in case of LILO problems
- Use boot floppy to run the RAM disk based installation system
    - Explore hard disk contents with "Mini-Linux"
- We have covered manually running 'fsck' if automatic 'fsck' fails
- What about unexplained crashes? Or complete system freezes? Disk-related problems?

Dealing with Potential Hardware Failures

- SCSI bus, cabling, connectors, and terminators
  - Always suspect!
  - Show up like undeterministic disk failures
- Real hard disk failures
  - Unreadable blocks (see '/var/log/kern.log') or noises
  - Increase over time --> backup quickly
- Memory / motherboard problems
  - Unexpected "Signal 15" and others
  - Dies in signals during long 'make' runs

Dealing with Memory Problems

- Memory problems are especially dangerous in Linux because it keeps frequently-used files in memory cache
  - Updating cached copy in memory may eventually lead in corrupt data being written back onto disk
- Make a 'memtest86.bin' bootable floppy disk
  - cat /usr/lib/hwtools/memtest86.bin >/dev/fd0
  - Boot it and let run for several hours

System Fans and Power Supplies

- The leading cause for hardware failures is clearly a failed, stopped fan
- CPU heatsink fans are especially nasty
  - Overheated CPUs cause similar problems as bad memory
- Do not expect a fan necessarily last for more than 2--3 years
- Power supply voltages are easy to check with a DMM at hard drive connectors (+5V, +12V)

Modifying Configuration Files

- Some of 'fsadapt' actions are illustrated
- Don't be afraid of reading the 'fsadapt' script!
- Loadable device drivers (like 'gpib0.o')
  - Modules themselves are within '/lib/modules/2.2.20'
  - Which modules to load is listed in '/etc/modules', can be edited for next boot
  - The command 'modconf' presents lists and "auto-edits" the file '/etc/modules', saving parameters in setup files in directory '/etc/modutils/'

Further Editing in '/etc'

- /etc/lilo.conf can be edited; remember to run 'lilo' after edits!
  - Adjust 'LILO' prompt wait delay
  - Create a dual-boot system
- Disabling user accounts for logins
  - Just replace the password in '/etc/passwd' with a '*': 'amn:*:500:500:...'
- X: /etc/X11/XF86Config is now autogenerated
  - Use 'dpkg-reconfigure xserver-xfree86' etc.

Printers

- lpd printer daemon is controlled by '/etc/printcap' (see 'fsadapt')
- lpc restart all

```
loukku|lp|lj5|hplj5|HP LaserJet 5M in Library:\
        :lp=:rm=print.kurp.hut.fi:rp=loukku:\
        :sd=/var/spool/lpd/loukku:\
        :sh:pw#80:pl#72:px#1440:mx#0:\
        :af=/var/log/lp-acct:lf=/var/log/lp-errs:

lp1|Raw byte stream for /dev/lp1 parallel port:\
        :lp=/dev/lp1:sd=/var/spool/lpd/lp1:\
        :sh:pw#80:pl#72:px#1440:mx#0:\
        :af=/var/log/lp-acct:lf=/var/log/lp-errs:
```

Updating, Adding, and Removing Software

- dpkg -- Debian's basic package tool
  - Can install and remove '.deb' packages directly
  - Knows about package dependencies but not about package archives and availability of updates
- Keeps installed state in /var/lib/dpkg/info
  - <name>.list, <name>.postinst
- All package installation, basic setup and removal is actually handled by dpkg

APT - packages made easy

- apt -- Debian's package archive tracking tool
  - Tracks package availability across multiple archives and releases
  - Allows installation by package name directly using 'apt-get install <name>' or upgrade of an installed package to the latest available version with a simple 'apt-get upgrade <name>'. Similarly removal using 'apt-get remove <name>'
  - Package archives are specified directly using the conffile '/etc/apt/sources.list' (see /man 5 sources,list') and CDROMs using 'apt-cdrom'

APT and Security Updates

- apt can also track security update availability at security.debian.org
  - First ensure following line is in /etc/sources.list (note the explicit 'woody' to stay within a particular re-lease)
    - deb http://security.debian.org woody/updates main contrib non-free
  - Use 'apt-get update' to reload package availability then 'apt-get -u upgrade' to see what upgrades are currently available
  - 'fsadapt' in FS Linux 5 installs automatic cron script based on this to warn about upgrades

'dselect'

- Tracks what packages are available on servers / CD-ROMs using APT; selects dependants

```
Debian GNU/Linux `dselect' package handling frontend.

   0. [A]ccess      Choose the access method to use.
   1. [U]pdate      Update list of available packages, if possible.
   2. [S]elect      Request which packages you want on your system.
   3. [I]nstall     Install and upgrade wanted packages.
   4. [C]onfig      Configure any packages that are unconfigured.
   5. [R]emove      Remove unwanted software.
 * 6. [Q]uit        Quit dselect.


Move around with ^P and ^N, cursor keys, initial letters, or digits;
Press <enter> to confirm selection.   ^L redraws screen.
```

Finding More Information

- The HOWTO collection of documents
- man 5 conffile
- cd /usr/doc/package; zless *.gz
- Linux Documentation Project
  - http://www.linuxdoc.org/
- Debian Bug Tracking System
  - http://www.debian.org/Bugs/
- www.google.com

Summary

- What we have covered today:
  - Getting System Information
  - Linux Startup & Shutdown
  - LILO Failures, 'fsck' Failures
  - Periodical Jobs with Cron
  - Network Configuration & Protection
  - Hard Disk, Partitions, File Systems, Mounting
  - The Root Directory Level -- /usr, /var

Summary

- What we have covered today:
  - File and Directory Protection, Inodes
  - Backup Operations, 'tar', tar copy/move
  - Rescuing a Failed System
  - Dealing with Potential Hardware Failures
  - Modifying Configuration Files
  - Updating, Adding, and Removing Software, 'dpkg', 'apt-get', 'dselect'
  - Finding More Information

DRAFT FS Linux 5 Disk Back-ups

Ed Himwich, Jonathan Quick, Dan Smythe

5 May 2005

Introduction

This document describes a scheme for maintaining back-ups of FS computers with removable disk packs. It is important to maintain back-ups of the system in case the disk being used for operations becomes corrupted. It is much easier to recover if you have a fairly recent back-up.

The scheme described here requires FS Linux 5. However it should be possible to adapt it to other Linux distributions.

This scheme is intended for use with three IDE disks, but only two are mounted at any one time. It can also be used with SCSI disks. A separate section describes how to modify the scheme for SCSI disks.

This back-up scheme is intended to meet four goals: (1) keep a fairly recent back-up available at all times, (2) keep a somewhat less recent back-up available in case the most recent back-up is compromised, (3) provide the station with flexibility for timing when the back-up is performed, and (4) keep all the disks exercised. The last point is important because disk drives seem to fail more often if they aren't exercised. One possible reason for this might be that the lubricant for the bearings does not stay well distributed when a drive is idle for a long time.

The back-ups are made using a script that re-initializes the file systems on the Back-up disk, transfers the contents of the Primary disk to the Back-up disk using the *tar* utility, and re-initializing the boot record on the Back-up disk. This should completely configure the back-up disk except for the sizes of the partitions. If necessary, the size of the back-up disk files systems can be adjusted using the *fdisk* (be very careful if you use this program) before a back-up is made. The *tar* utility is used for the back-up instead of making a disk image copy of the primary disk for two reasons: (1) with the very large disks sizes available now, an image copy of a disk can take a very long time, and (2) a *tar* back-up of the disk contents is more robust if the spare sector pool on the back-up disk is exhausted.

A. Rotating Back-up Scheme

The basic plan is to use three disks. They should be labeled 1, 2, and 3 (or A, B, C etc). Each disk is mounted in a "carrier". The computer has two disk "frames" (or "slots") that accept the disk "carriers". One slot is labeled "primary" and the other labeled "back-up" (or maybe "secondary"). One disk will always be in the Primary slot and is the

Operational disk. One disk will be in the Back-up slot is the **Back-up** disk. The third disk is kept on the shelf and is the **Shelf** disk. The section What if you don't have three disks describes what to do if you only have two disks.

The computer should normally be run with two disks inserted. The Operational disk is the disk in the Primary slot because by configuration this is the disk that is booted. The Back-up disk is in the Back-up slot because this is slot that back-ups are written to. The Back-up disk is not "mounted" by operating system when the operational disk boots; in other words it is not normally accessible to users. Having a Back-up disk installed at all times makes it relatively easy to make a back-up.

The standard back-up scheme is that the Operational disk is backed-up periodically, perhaps weekly, to the Back-up disk. At some longer interval, three months is suggested, the user will rotate the disks after the weekly back-up. The disk in the Primary slot will be moved to the shelf, the disk in the Back-up slot will be moved the Primary slot, and the one that had been on the shelf will be moved to the Back-up slot. In tabular form, a cycle of three rotations would look like:

Months	1-3	4-6	7-9	10-12
Primary	**Disk 1**	**Disk 2**	**Disk 3**	**Disk 1**
Back-up	**Disk 2**	**Disk 3**	**Disk 1**	**Disk 2**
Shelf	**Disk 3**	**Disk 1**	**Disk 2**	**Disk 3**

B. Periodic Back-ups

The timing of periodic back-ups is to some extent the decision of the station operators and based on how heavily the system is used and how often it is modified. The guiding principle in deciding how often to perform back-ups is a trade-off between the normally minor disruption it causes against the difficulty of reconstructing lost work since the last back-up in case a restore is necessary. Normally neither of these considerations is overwhelming. Consequently, it is often sensible to just establish some routine, perhaps weekly, for when back-ups should occur.

If back-ups are made weekly, a good time to start a back-up might be just before the last operator leaves for the end of the work week. The back-up can run unattended and the results checked when the first operator arrives at the start of the next work week. In contrast, starting a back-up first thing at the beginning of work week or day would not be as robust since there may be some uncertainty whether anything may have affected the system while people were away. The operators have more recent experience that the system was working well at the end of the week than they do when they come in for a new week. That experience should be utilized.

C. Other Back-ups

There are other times when it is useful to make a back-up, in particular right before making a major change to the system. The most obvious examples of this are FS and kernel upgrades. Having a pre-upgrade back-up available will make recovery very easy if something goes wrong in the upgrade. Please note that after installing such an upgrade, you should delay making the periodic next back-up of the operational system until after you have had a chance to verify the upgrade worked well.

Whenever the back-up is to be performed, please follow the directions in the section below on Making Back-ups.

D. Making Back-ups

Back-ups can be made using a script provided with the FS Linux 5 utility *fsadapt* (current version 5.3). Using the script requires that two disks be installed in the computer. The disk to be backed-up is in the Primary slot and the disk to receive the back-up copy in the Back-up slot.

Back-ups can be performed at anytime. However, it is desirable that they performed when as few users as possible are logged in and as few programs as possible are running. Ideally, they would be performed in single user mode, but this is not necessary. Please use the following procedure:

```
(login as root)
cd /root/fsadapt
./fullbkup >/dev/null
umount /db/usr2
umount /db
(log-out as root)
```

There are some useful variations of these commands. You can see a list of files being transferred by leaving off the "`> /dev/null`" off the command, i.e., using just "`./fullbkup`". Another variation that may be useful when you are leaving it to run unattended is "`./fullbkup >/dev/null &`", which will start the process in background, and then you should log-out as the root user (this is more secure than leaving the root user logged in while the back-up executes). However in this case, after the back-up completes, you will need to log in as root again execute the two "`umount …`" commands shown.

You can check the results of the back-up by listing the contents of the file `/root/fsadapt/tar-errs` for any error information:

```
less /root/fsadapt/tar-errs
```

The file will show the start and stop times and any error that may have occurred.

It is advisable to verify the back-up after you have made it. However, for ease of use this can normally be skipped for Periodic Back-ups. In come cases, such as when doing making a Disk Rotation or performing a Restore, it is implicit in the operation that you will begin using the resulting copy (and thereby practically verifying it) immediately. For many Other Back-ups you may wish to verify the back-up depending on the reason for the back-up and how costly having a problem would be. In order to verify a back-up: shut down the OS on the FS computer, turn off the power, remove the Primary disk and put the Back-up disk in the Primary slot, and turn on the power. After the computer boots, you should be able to identify some recently added or modified file on the disk in the Primary slot and that the FS runs as expected. Finally shut-down the OS, turn off the power, return the original Primary disk to the primary slot, and turn the power back on.

E. Disk Rotation

Periodically, perhaps every three months, the disks should be rotated. The disk rotation scheme requires three disks. Only two are in use at any time, but all three disks are cycled through to keep them exercised and to provide extra back-up security. The disk rotation can naturally be performed right after a Periodic Back-up or Other Back-up. The rotation pattern is described in Rotating Back-up Scheme. For future reference, please be sure to record the date of the last rotation somewhere prominent, like a label on the front of the PC. A suggested format is a table filed in by hand for each rotation, for example:

Date	new Primary	new Secondary	new Shelf
10/01/04	2	3	1
11/05/04	3	1	2

F. Restoring from a Back-up

Like the back-up scheme, restoring is very simple, but should be required rarely if at all. A restoration is typically needed only, e.g., if the operational primary disk has suffered a loss of data, if the operational primary disk becomes corrupted, or if the station wants to return to known previous working version of the FS or Linux that is installed on a back-up disk.

Although it is possible to just switch to using a disk that holds a back-up copy (and this can be done in an urgent situation), it is preferable to restore the back-up copy to the original Primary disk and use the Primary. This has the advantage that you will not be endangering the integrity of the back-up by using it for operations.

The first attempt at making a restoration should be to use the Back-up disk (in the Back-up slot), unless it is known to have older contents than the Shelf disk or suspected of having damaged contents. If the Shelf disk is more up to date or if the disk from the Back-up slot is not usable for some reason, you should restore from the Shelf disk.

The restoration process is the same as for the manual back-up process. However, you must reverse the disks, so that the disk that contains the back-up to be restored is in the

Primary slot and the disk you wish to restore the back-up to is in the Back-up slot. Once you have established this arrangement, you can follow the instructions in Making Back-ups. Once the back-up is complete, you will need to shutdown the OS, turn the computer off, place the disk from Back-up slot (the original operational disk) back into the Primary slot, and turn on the power.

Once the computer has booted from the restored Primary disk, you will need to perform any upgrades and add/change any files that were added/changed since the back-up was made. In the best case, this may require nothing or only adding some schedule (.skd or .drg) files and running DRUDG to make some SNAP (.snp) and SNAP procedure files (.prc). Things will be a little more involved if you need to upgrade the FS because of new features that are required. It is even more involved if you would need to install upgrades to the kernel. However, kernel upgrades are largely automated and in addition often can be bypassed for some period of time if there is an urgent need to get back to an operational state. However, installing kernel upgrades should not delayed any longer than necessary since not doing so may compromise the security of your computer.

G. What if you don't have three disks

This situation may arise either if had three IDE disks and one fails or if you have an older FS configuration that has only two removable, SCSI, disks. In this case you can continue the Periodic Back-ups and Other Back-ups, but will need to discontinue the Disk Rotations. However, it is recommended that at the interval for the Rotation, you should verify your back-up as described in the Making Back-ups section.

If you find yourself with only two disks because one of your IDE disks has failed, it is recommended that you buy a new disk as soon as possible so that you will have a third disk on hand. The replacement disk does not have to have the same capacity, but it should probably be at least as large. Given the low prices and rapidly increasing the size of IDE disks, there should be no obstacle to getting a new larger disk. The new disk can be placed in the carrier for the broken disk. Before it is used it will need to be initialized, please see the FS Linux 5 Installation instructions for information on how to do this.

If you have only two disks because you have an older FS PC configuration that uses SCSI disks, it is probably not economical, if it is even possible, to buy another disk that is compatible with your old disk controller. In addition, the rotation scheme is more tedious for SCSI disks because of the need to set the drive ID (but see SCSI issues for more details); typically this is not an attractive approach. In this case, you have the option of continuing to use just two disks without the added security of the rotation scheme or, preferably, you could upgrade to a new PC that uses removable IDE disks. You should consider an upgrade in any event if your PC is more than three years old.

H. SCSI issues

For older systems that use SCSI disks, a few considerations need to be kept in mind when doing back-ups. The most important is that whether a disk is the Primary or Back-up is determined by its ID number, not by its slot. The number is usually set with a jumper on the disk. (If your computer is set-up so that the slot does determine the ID of the disk, then you can ignore this section and can read the rest of this document as though it applies to you even though it is written for IDE disks).

The SCSI disk with the lowest ID number (usually 0) is the Primary (or Operational) disk. The disk with the next highest ID number (usually 1) is the Back-up disk. Despite this difference, most aspects of this document do not need to be modified for SCSI disks as long as you keep mind that the Primary disk is not determined by the slot, but by the disk itself. The cases where it makes a difference is for the sections for restoring and disk rotation and for the verification process described in Making Back-ups. For restoration, the thing is that is different is that instead of placing the disk to be copied from in the Primary slot and disk to be copied to in the Back-up slot, you will need to make sure that the disk being copied from has a lower ID number than the disk to be copied to. You may be required to change the jumpers on one of the disks. This might be accomplished most easily by changing only the ID (perhaps 0 to 4) of the Primary disk (the disk to copied to), so that it is larger than the ID (usually 1 or 2) for the Back-up disk (the disk being copied from). For disk rotation you will also need to make sure that the new Primary disk has a lower ID number than the new Back-up disk. However, another option with disk rotation for SCSI disks would be to just rotate the Back-up and Shelf disks, e.g., if the Primary disk has ID 0 and the Back-up and Shelf disks have IDs 1 and 2 (in no particular order), there will be no need to change IDs if only the latter two disk are rotated. This eliminates one of the advantages of the three disk rotation scheme, but is perhaps a reasonable trade-off.

Happily for the case of verifying a back-up, it is actually slightly easier for SCSI disks. In this case instead of moving the back-up to the Primary slot, you will merely need to remove the Primary disk, the Back-up disk automatically becomes the one that will be booted from. After verification, the Primary disk merely needs to be reinserted.