Linux System Administration

Jonathan Quick
Hartebeesthoek Radio Astronomy Observatory

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
- For the field system and Mark5's

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")

System Modifications

- In order to do any system-wide changes you usually have to be logged in as 'root‘
  - Or have root privileges
- There are a number of approaches for this
  - Log in as user “root”
  - Execute “su –” from the present user account
  - Execute the command directly with “sudo”
    - E.g. “sudo tail /var/log/kern.log”

Logging 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').

Sudo

- Sudo is a program designed to allow a sysadmin to give limited root privileges to users and log root activity.
- The basic philosophy is to give as few privileges as possible but still allow people to get their work done.
- The sudo package, by default is not installed
  - Mark5A, or FS
- To install
  - apt-get install sudo
  - /etc/sudoers is the access list

Getting System Information

- ps axf, top
  - Snapshot of current running processes
- kill, kill -9
  - Send a signal to a process,
  - -9 = KILL
- free
  - Display amount of free and used memory in the system

Getting System Information

- df, mount, du
  - File system commands
- netstat -an, ifconfig, route –n
  - TCP / IP administration commands
- w, who
  - Users logged in and what they are doing
- cat /proc/cpuinfo (and others)
  - Many commands display information that is kept under /proc

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 Grub, the Linux Loader
  - Shows Grub Menu, 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 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:

Note -1, one column, not -l

```
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
```

Instructions!

S10checkroot.sh 'fsck' Check of File Systems

- If the system was not properly shut down, these scripts attempt an automatic 'fsck'
- If repairs would require deletions etc:
  - "fsck failed. Please repair manually and reboot."
  - Enter the 'root' password when asked to
  - fsck /dev/sda1 or hda1
    - 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
  - 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 the 'K*' "kill" scripts (/w 'stop') and then 'S*' "startup" scripts (/w 'start') of the new 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 the "kill" scripts of '/etc/rc0.d'
- 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

Summary of Configuration Files Affecting Startup

- /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 those directories are really in '/etc/init.d'
  - Manually starting/stopping something:
    invoke-rc.d something start
    invoke-rc.d something stop

Periodical Jobs with Cron

- The 'cron' daemon runs in the background with 1 min resolution, starting timed 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 routes
- /etc/hostname, /etc/hosts
  - Has name and IP address of this computer
- /etc/resolv.conf
  - Has the IP addresses of DNS name server(s)
- /etc/network/interfaces
  - The details of all available network interfaces

Network Protection “tcpwrappers”

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

/etc/hosts.allow & /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

MD?

- MD is the software RAID (Redundant Array of Inexpensive Disks) layer in the Linux kernel :- /dev/mdX
  - RAID0: make one filesystem striped across many disks or partitions (NB: not redundant at all !! - done to speed disk access)
  - RAID1: mirror one filesystem across multiple disks or partitions (fully redundant)
  - RAID5: uses a parity disk/partition to make a (fault tolerant) filesystem from many disks.

LVM?

- Logical Volume Management is a further virtual partitioning layer in the Linux kernel
  - Similar to RAID0 in that it collects multiple disks partitions together into volume groups in which resizeable logical volumes (used to contain filesystems) can be allocated.
  - No redundancy so failure of any one disk could affect all logical volumes in the group thus best used on top of RAID.

The Root Partition

- The partition mounted as '/' by the kernel
  - GRUB 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 on the '/' 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 (or volumes)
- Partitions can be formatted i.e. a file system is created in a partition (or volume)

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' (with journalling)
- 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: mkfs.ext3 /dev/hda1
  - Raid volumes: mkfs.ext3 /dev/md0
- Mounting to a mount point (=directory)
  - mount -t ext3 /dev/hda3 /mnt
- Unmount with 'umount /mnt'
- Boot time mounts in /etc/fstab , 'mount -a'

/dev/sda3 /usr2 ext2 defaults 0 2 ← fsck #

mount -t ext2 -o defaults /dev/sda3 /usr2

Managing Mounted File Systems

- To see what partitions are mounted:
  - mount
    - Displays information from /etc/mtab
    - More convenient to use 'df' to display unused space
- Mount points are normal directories
  - Mounting "hides" the old directory contents
- Unmounting is necessary before:
  - fsck, mkfs.ext3, fdisk, tune2fs
    - These directly alter file system / partition structures!

Linux Directory Structure

- /boot -- boot files
- /dev -- device special files
  - Map to major/minor numbers & kernel drivers
- /etc -- configuration files (usually read-only)
- /bin, /usr/bin – contain important binaries
- /sbin, /usr/sbin – system administration programs
- /usr – user applications, source code, config files, docs

Linux Directory Structure (cont)

- /usr/share/doc -- documentation
  - /usr/share/doc/HOWTO -- "cookbook" instructions
- /mnt, /floppy, /dosa -- temporary mount points
- /proc -- process information (virtual file system)
- /root -- home directory of 'root' user

Linux Directory Structure (cont)

- /var – contains "variable“ data that changes constantly while system is running
  - /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/directories are owned by one user & 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 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

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"
- With the newer 'ext3' file system, journalling is used to keep thing consistent even on abrupt reboot - 'fsck' now only used for maintenance.
- Every 'ext3' file system still has a 'lost+found' directory for recovered files

Backup Operations

- Field System
  - Software Raid1 scheme
    - 2 disk drives inserted at all times
      - third disk on the shelf
    - Mirrors of each other
    - Provides automatic backup
  - RAID.txt
    - Details of operation

Dealing with Potential Hardware Failures

- SCSI bus, cabling, connectors, terminators
  - Show up as undeterministic disk failures
- Real hard disk failure
  - Unreadable blocks (see '/var/log/kern.log') or listen for noises
  - Increase rapidly 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
- Add a 'memtest86+' to your GRUB menu
  - aptitude install memtest86+
  - Reboot to 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 fans to 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

- Some of 'fsadapt' actions are illustrated
- Don't be afraid of reading 'fsadapt' script!
- Loadable device drivers (like 'nigpib.o')
  - Modules themselves are within '/lib/modules/2.6.18-6-686'
  - 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 '/etc/modutils/'

Further Editing in '/etc'

- Disabling user accounts for logins
  - Just replace the password in '/etc/passwd' with a '*': 'user:*:500:500:...'
- X configuration is now autogenerated
  - Use 'dpkg-reconfigure xserver-xorg' etc.

Printers

- CUPS printer daemon is configured in '/etc/cups/cupsd.conf'
- Easiest configuration is using the CUPS web interface:
  - Navigate to the URL http://localhost:631/

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 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 explicit ‘lenny' to stay within a particular release)
    - deb http://security.debian.org lenny/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 6 installs automatic cron script based on this to warn about upgrades

'dselect' or 'aptitude'

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

```
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.
```

Synaptic

- GUI based management of software packages
- A frontend for the apt package management system
  - Therefore performs all actions of apt-get
    - Installing, upgrading, downgrading and removing
      - of single packages
    - Upgrading your whole system.
- Nice search option

Finding More Information

- The HOWTO collection of documents
- man 5 conffile
- cd /usr/share/doc/<package>; zless *.gz
- The Linux Documentation Project
  - http://tldp.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
  - '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
  - Dealing with Potential Hardware Failures
  - Modifying Configuration Files
  - Updating, Adding, and Removing Software, 'dpkg', 'apt-get', 'dselect', 'aptitude‘, ‘synaptic’
  - Finding more information