Linux System Administration

Ari Mujunen, Metsähovi Radio Observatory

Linux Startup and Shutdown
Managing Hard Disks, Partitions, Backups
Rescuing a Failing PC / System
Modifying Configuration / fsadapt

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/sda1
- 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 alphabeti−cal 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' 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’ directo– ries 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/hosts
  - Has the name and IP address of the computer itself
- /etc/resolv.conf
  - Has the IP addresses of DNS name server(s)
- /etc/nsswitch.conf (old /etc/host.conf)
  - The order of DNS/file name––IP lookups

Network Protection with "tcpwrappers"

- During boot, the "Internet 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’ 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 4 primary partitions only
- 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 BIOS by keep–ing them below the 1024 cylinders boundary

The Root Partition

- The partition mounted as '/' by the kernel
  - LILO boot parameter can change this
- 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 "native" format for Linux is the "Second Extended Filesystem" ’ext2’
- MS–DOS/Windows floppies and FAT partitions can be used as ’vfat’
  - Supports long file names and FAT32
- Network File System ’nfs’, 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
mount -t ext2 -o defaults /dev/sda3 /mnt
```

fsck #

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, 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
- /etc –– configuration files
- /mnt, /floppy, /dosa –– temporary mount points
- /proc –– process information
- /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, email)
  - /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 pro–tection "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
- 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"
- 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 run ’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 (’|’)
- 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 cre–ated, it streams in the pipe (’|’)
- Combine with ’ssh’ to make remote copies:
  - ssh root@remote ’tar clf – /’ | (cd /bks; tar xvpf –)

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 in–stallation 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’)
  - 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/2.0.34’
  - 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’

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 an ’*’: ’amn:*:500:500:...’
- X: /etc/X11/XF86Config

Printers

- lpd printer daemon is controlled by '/etc/printcap'
- 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 package tools
  - Can install and remove '.deb' packages
  - Knows about package dependencies
- Keeps installed state in /var/lib/dpkg/info
  - <name>.list, <name>.postinst
  - fgrep blurb /var/lib/dpkg/info/*.list
- dselect, apt

’dselect’

- Knows what packages are available at desired servers / CD–ROMs; 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’, ’dselect’
  - Finding More Information