LPI Level 1, Exam 101

Linux Professional Institute Exam 101 Objectives are at
http://www.lpi.org/en/obj_101.html

$Revision: 56 $
$Date: 2005-01-17 15:52:48 -0500 (Mon, 17 Jan 2005) $

Hardware & Architecture

Configure Fundamental BIOS Settings

- BIOS firmware sets clock, boot devices, interrupts (IRQs), I/O addresses, and Direct Memory Access channels (DMAs).
- I/O ports, IRQs and DMA assignements are finite resources – avoid conflicts.
- Some (older) BIOSs can't read past a 1024 cylinder limit: avoid placing a boot loader and/or a kernel past the first 1024 cylinders.
- \proc filesystem contains useful entries for checking configuration.

\proc\interrupts	Currently allocated interrupts.
\proc\ioport	Currently allocated I/O ports.
\proc\dma	Currently allocated dma channels.

Configure Modem and Sound cards

- External modems on serial port. \dev\ttyS0
- Internal modems usually also present themselves as a serial port.
- *Winmodems* use the CPU for dsp and rely on proprietary firmware. Linux drivers are *sometimes* available. Avoid.
- Avoid I/O conflicts when configuring modems. PC serial ports can share a single IRQ, but not an I/O port.
- ISA sound cards use kernel modules which have settings stored in \etc\modules.conf.
- sndconfig helper app available for isa cards.
- PnP ISA uses isapnp configuration.

Setup SCSI Devices

- Most SCSI devices must have unique ids.
- SCSI disks may share a single id but have a logical unit number (lun).
- SCSI devices in linux are typically at \dev\sda, \dev\sdb\, etc. Assigned sequentially by SCSI id.
- SCSI uses terminators for signal conditioning/noise reduction.
- SCSI controllers have their own BIOS to determine boot order.
- Linux usb subsytem uses SCSI emulation. USB disks appear in \dev\sda

\proc\scsi	SCSI kernel parameters.

Setup different PC expansion cards

- \proc filesystem tools and lspci, isapnp, and pnpdump are useful for ISA and PCI expansion card installation.
- modprobe *module.o module-options* will attempt to insert kernel module and dependent modules.
- Save module configuration in \etc\modules.conf
- List modules currently in the kernel with lsmod

\proc\pci	PCI bus configuration.
lspci	list all PCI devices

Configure Communication Devices

- setserial can modify baud rates, IRQs and I/O ports for serial ports.
- setserial -a /dev/ttyS0 – Displays all serial port information for first port.
- setserial /dev/ttyS4 port 0x800 irq 9 – Sets serial port paramters

Configure USB devices

lspci	list all PCI devices
lsusb	lists usb bus devices
usb-uhci.o	USB module for Universal Host Controller (Intel, Via)
usb-ohci.o	USB module for Open Host Controller
/etc/usbmgr/ usbmodules /etc/hotplug	

Linux Installation & Package Management

Design hard disk layout

- fdisk /dev/hda – run formating utility on first ide drive.
- fdisk -l /dev/hda – list partitions on first ide drive.
- You may have a maximum of 4 primary partitions
- workaround is the extended partition, which one of the primaries may be.
- 1 extended partition contains a maximum 16 logical partitions.
- disk-space limited system:

\boot	50M
\swap	∼ 2× memory
\	Remaining space

- Larger system partition scheme

\boot	50M
\swap	2× memory
\tmp	200M
\usr	∼ 1G
\var	∼ 1G
\home	large!

Install a boot manager

lilo

- lilo configuration file in /etc/lilo.conf
- lilo must be rerun after changes to /etc/lilo.conf

grub

- grub configuration file in /boot/grub/grub.conf
- lilo must be rerun after changes to /etc/lilo.conf

Make and install programs from source

- tar xjvf tarball.tar.bz2 or tar xvzf tarball.tar.gz
- ./configure
- make
- make install

Manage shared libraries

- ldd – show shared libraries for a program
- /etc/ld.so.conf – paths for locating system libraries
- ldconfig – Run to generate ld.so.cache library db from ld.so.conf
- ldconfig -v – list current libraries pathed
- LD_LIBRARY_PATH – enviromental variable for user library search path

Use Debian package management

- apt-get update – update available packages from \etc\apt\sources.list
- apt-sources – interactive editor of \etc\apt\sources.list
- apt-get upgrade – upgrade all packages
- apt-get install *packagename* – install a package
- apt-get -s install *packagename* – simulated install
- dselect – interactive package selector
- apt-get dselect upgrade – interactive package selector
- dpkg -i *package.deb* – install a debian package
- dpkg -s *packagename* – status of package
- dpkg -L *packagename* – list files belonging to package
- dpkg -S *filename* – lists package file belongs to
- dpkg-reconfigure *packagename* – run interactive package configuration.

GNU & Unix Commands

Work on the command line

- bash – Bourne-Again shell
- PATH variable – lists the PATH to files you can run from command line w/o using the full path to the executable.
- shell variable to be used as environmental variable – export MYVAR
- Command seperator ; as in make;make install
- history – lists the history
- !! – most recent command
- !n – command n in history
- !-n – nth before history
- !?string – most recent command with string
- ^string1^string2 – the first occurance
  - C-p Previous
  - C-n Next
  - C-b Back a character
  - C-f Forward a character
  - C-a Begining line
  - C-e Begining line
  - C-l Clear the screen
  - M-< Top history
  - M-> Bottom of history
  - C-d Delete character from right
  - C-k Delete characters to end of line
  - C-y Yank text previously cut.
- $(COMMAND) or `command` replaces result of command.
- pwd – Current working directory
- set – Sets BASH options
- unset – Unsets a BASH option
- exec – Runs a program
- ~/.bash_history – History storage
- ~/.bash_profile – Bash runs with login instance.
- ~/.bashrc – Bash runs with each bash subshell
- ~/.profile – Bash runs if no /.bash_profile
- ~/.bash_logout – Bash runs upon logout.

Process text streams using filters

- `cat` – echo file contents, can be redirected
- `tac` – echo file contents in reverse order
- `uniq` – removes duplicate lines if already sorted
- `wc` – word count
- `head` – Output first part of file
- `head -10` – Output first 10 lines of file
- `tail` – Output last part of file
- `tail -f` – follow changes to tail of file
- `expand` – Expands tabs into spaces
- `cut -l` – Output last part of file
- `nl` – line numbering
- `pr` – breaks files into pages
- `od` – octal dump of file
- `tr` – translate characters from input to output
- `paste` – vertical columns from seperate files
- `split - n` – split file into groups of *n* lines.
- `tee` – stdin piped to stdout and to file.
- `sed` – stream editor
- `join` – joins files with field one repeating
- `fmt` – checks space formating
- `xargs` *command* – passes arguments from *stdin* as arguments of command.

Perform basic file management

- `cd` – change directory
- `stat` *file* – shows file information
- `ls` – list directory
  - `ls -l` – long directory displays mtimes.
  - `ls -a` – show hidden files
  - `ls -d` – show only directories
  - `ls -i` – show inode numbers
  - `ls -u` – show atimes (last use)
  - `ls -c` – show ctimes (change)
- `mkdir dirname` – make a directory
  - `mkdir -p` – parent directory
- `rmdir dirname` – remove a directory
- `touch` – modifies atime and mtime
  - `touch -a` – change atime
  - `touch -m` – change mtime
- `echo` – echos (can be redirected)
- `cp` – copy file
- `mv` – move or rename a file
- `rm` – remove a file (`-r` recursive, `-i` interactive)

Use streams, pipes, and redirects

- types of pipes
- redirects:
  - `>` – into a file
  - `>>` – append to file
  - `|` – pipe to next command
  - `2>` – stderr to file
  - `1>` – stdout to file
  - `2>&1` – stderr and stdout to same file

Create, monitor, and kill processes

- `C-c` – Issue kill for running app.
- `C-z` – Suspend app.
- `fg` – Run as a foreground application
- `fg %2` – Run job 2 in foreground
- `jobs` – list jobs
- `bg` – run job in background
- `xclock &` – run job in background
- `nohup` *job or pid* keep running after SIGHUP.
- `ps ax` list all processes w/o controlling tty
- `ps -x --forest` list all with dependencies
- `pstree` pretty printed ps tree-diagram
- `ps -al` process in a long format
- `top` interactive process monitor: curses
- `kill -9` *pid* – Send signal 9 to process
- `kill -s SIGKILL` *pid* Same as -9.
- `kill -SIGKILL` *pid* Same as -9.
- `kill -l` list kill signals by name and number.
- The important signals:

Signal	Number	Meaning
SIGHUP	1	Hang up – Logout signal
SIGINT	2	Interrupt – CTRL-c
SIGKILL	9	Drastic process kill
SIGTERM	15	Teminate nicely if possible
SIGTTSTP	18	Stop executing, ready to continue. Ctrl-z

Modify process execution priorities

- `nice` run a program with a low priority (10)
- `renice 2 -p 753` change priority to 2 of pid 753; negative is less nice

Search text files using regular expressions

- Wildcards expand to become arguments to command by shell:
  - `*` – matches none or multiple characters
  - `[14]` – matches one character between 1 and 4
  - `[!4]` – matches one character not 4
  - `?` – matches a single character
  - `{one,two,three}` – expands to match substrings
  - Single quotes and globs don't work together.
- Regular Expression evaluated by commands like grep and sed
  - `grep bash /etc/passwd` – matches lines with 'bash' in them.
  - escape `\` regex characters.
  - put in quotes to avoid glob substitution by shell
  - Escape literal searches for `+ . * [ ] \`.
  - `grep '\[' filename` – searches for bracket in filename.
  - `'.'` – matches any single character
  - `'[12]'` – matches a single character `1` or `2`
  - `'[^12]'` – matches any single character not `1` or `2`
  - `'.*'` – matches one or more character
  - `'r*'` – matches none or more r's
  - `'^s'` – matches lines that start with s
  - `'s$'` – matches that end in s
  - `'^#.*\.$'` – matches that starts with a `#` and ends in a `.`

Perform basic file editing operations using vi

- `vi` – world's best text editor, modal editing
- `vi +4` *filename* – line 4 ready
- `vi +/search \emph{filename}` – finds search
- i – insert
- esc – return to edit mode
- a – appends after cursor
- i – inserts before cursor
- C – change line
- R – replace line
- y – yank
- p – paste
- / – search
- ESC-:s/find/replace
- ZZ – write and quit
- ESC-:wq – write and quit
- ESC-:q – quit
- ESC-:q! – no, really quit

Devices, Linux Filesystems, Filesystem Hierarchy Standard

Create partitions and filesystems

- `fdisk` – format your drive
- `mkfs` – make a filesystem

Maintain the integrity of filesystems

- `du` *dirname* – Disk usage of *dirname*, human readable with `-u`
- `df` – Disk free, human reabable with `-h'`
- `fsck` – Fix the filesystem
- `e2fsck` – Fix the e2fs filesystem
- `mke2fs` *partition* – Make an e2fs filesystem.
- `mkswap` *partition* – Make an swap space.
- `swapon` *partition* – Turn on the swap.
- `debugfs` *debugfs* – Dirty nitty gritty fix the filesystem.
- `dumpe2fs` – dump filesystem to file.
- `tune2fs` – Set options for filesystem.

Control mounting and unmounting filesystems

- `\etc\fstab` –format: block device, mountpoint, type, options, numeric fields.
- numeric fields: dump field and fsck order
- common options:
  - defaults – standard mount options
  - noauto – Don't automatically mount on boot
  - noatime – Turns off recording of last access time.
  - ro – Mount Read-Only
  - user – Users can mount
  - usrquota – Quotas for users
  - grpquota – Quotas for groups
- `\mount -t` *fstype blockdevice mountpoint mountoptions* – Mounts a filesystem (need to be root)

Managing disk quota

- `quota` – set quota (`-u` for user, `-g` for group)
- `quota -v` – list quota (`-u` for user, `-g` for group)
- `edquota` – set limits on quotas.
- `edquota -p` – set limits on quotas using another user as a prototype.
- `repquota` – reports on quotas.
- `warnquota` – sends emails to users warning them quota is close.
- `quotaon` – turns quotas on.

Use file permissions to control access to files

- File types:
  - `-` – regular file
  - `d` – directory
  - `l` – symbolic link
  - `c` – character special device
  - `b` – block special device
  - `p` – fifo
  - `s` – socket
- Linux filepermission types:
  - `r` – read access, for directories
  - `w` – write access (and deletion), for dir deletion or rename of contents
  - `x` – execute, and directory access for traversal.
  - `X` – execute only if directory or already has permission of some user
  - `s` – set user id or group id on execution for this file.
  - `S` – set user id or group id but execution disabled for this file.
  - `t` – sticky bit when set for a directory means that only the owner of the file and the owner of that directory may remove the file from that directory.
  - Setting a direcory with a setgid bit causes files created by members of the group to have a group ownership for that group.
- The three triplets of octal permissions are for user(u), group(g), others(o):
  - `rwx` – octal 7, binary 111
  - `rw-` – octal 6, binary 110
  - `r-x` – octal 5, binary 101
  - `r--` – octal 4, binary 100
  - `-wx` – octal 3, binary 011
  - `-w-` – octal 2, binary 010
  - `--x` – octal 1, binary 001
- the fourth triplet:
  - setuid – octal 4
  - setgid – octal 2
  - sticky – octal 1
- `whoami` – check your current uid
- `groups` – check what groups you are in
- `chown` *username filename* – change file ownership
- `chgrp` *groupname filename*– Alters file permissions
- `chmod u+rx` *filename*– Alters file permissions adding read and execute
- `chmod g=r` *filename*– Defines group permissions as being read only.
- `umask 0022` determines the file creation default permissions. As opposed to chmod, the umask defines what permissions are turned off.

Manage file ownership

- `chown` – Changes file ownership
- `chgrp` – Changes file group ownership

Create and change hard and symbolic links

- `ln file_to_link link_name`
- `ln -s file_to_link link_name`
- Hard links only work on the same filesystem (need same inode).
- Symbolic links work across filesystems.

Find system files and place files in the correct location

- FHS:
  - `/` – root filesystem
  - `/boot` – static files of the boot loader
  - `/dev` – device files
  - `/lib` – essential libraries and kernel modules
  - `/tmp` – temporary files
  - `/usr` – non-essential files (secondary hierarchy
  - `/usr/local` – admin added executable
  - `/bin` – essential user command files
  - `/sbin` – essential boot administrative files (root part.)
  - `/usr/sbin` – admin files not needed for boot
  - `/etc` – host specific config files (root part.)
  - `/home` – home directories for users
  - `/opt` – optional add-ins
  - `/var` – databases, mail, logs, and such,
  - `/mnt` – mountable but not needed at boot filesystems
- `which` – search in the path
- `whereis` – supporting programs (like manpages)
- `find` *path* `-name` *globname* – find a filename
- `find` *path* `-type l` – find a symbolic links
- `find` *path* `-type d` – find a directory
- `find` *path* `-mtime` – find files matching an mtime
- `find` *path* `-size -50c` – find files smaller than 50 characters.
- `find` *path* `-iname` *globname* – find a filename, case insenstive.
- `find` *path* `-regex` *regex* – find a filename, case senstive regular expression search.
- `find` *path* `-iregex` *regex* – find a filename, case insenstive regular expression search.
- `find` *path matches* `-exec ls -al '{}' ';'` – find file and run ls -al on them.
- `locate` *name* – matches filenames with substring name.
- `updatedb` – Updates the locate database.

The X Window System

Install & Configure XFree86

- Check version with `X -version`
- Configuration stored in `/etc/X11/XF86Config`
- `xf86config` text mode program.
- `XF86Setup` graphical setup
- `xf86cfg` block diagram setup
- `XFree86 -configure` Probe and create a XF86Config to test.
- Sections of XF86Config:
  - "Files" – FontPath
  - "Module" – Runtime loadable modules
  - "ServerFlags" – VT Switching and core dump flags
  - "InputDevice" – Keyboards and mice
  - "Monitor" – Monitor and sync settings
  - "Device" – Graphics device settings
  - "Screen" – Screen settings
  - "ServerLayout" – Links together InputDevices and Screens

Setup a display manager

- `\etc\X11\xdm` configuration directory
- `\etc\X11\xdm\Xaccess` controls inbound requests from remote hosts.
- `\etc\X11\xdm\Xservers` Resource file
- `\etc\X11\xdm\Xsession` Script launches after a successful login
- `\etc\X11\xdm\Xsetup_0` Script started before login screen appears
- `\etc\X11\xdm\xdm-config` Expert admin settings

Install & Customize a Window Manager Environment

- `startx` Starts an X session by calling xinit with .xinitrc
- `xinit` Can call an alternate xinit.
- `xterm` Calls .Xdefaults
- `DISPLAY` variable to point to your display

init Runlevels

- init run by kernel – pid 1
- Chnage runlevel with telinit.
- runlevels:
  - 0 – shutdown now
  - 1 – single user maintenance
  - 2 – multiuser no nfs
  - 3 – multiuser text mode (command line)
  - 5 – GUI starts X
  - 5 – same as 4
  - 6 – Reboot immediately
- nicer way sends messages to users: `shutdown`
- edit files in `\etc\inittab`

`$Revision: 56 $ $Date: 2005-01-17 15:52:48 -0500 (Mon, 17 Jan 2005) $`
http://www.drfox.com