# Cailin's LPIC-101 summary

This document ain't much special :) Just a summary of all the stuff I'm learning about Linux that I didn't already know from Solaris. Do not use this summary exclusively to study for your LPIC certification! It doesn't cover everything you need to know and most definitely is not a good replacement for a complete book.

This summary was based on the following two books and a lot of mucking about using a basic Linux install.

- Ross Brunson – "Exam cram 2: LPIC 1", 0-7897-3127-4
- Roderick W. Smith – "LPIC 1 study guide", 978-0-7821-4425-3

**Please note!** I've glossed over the whole PPP and modem stuff, because right now (two days before the exam), I can't be pantsed to learn that crap. I'm just too tired and too busy to bother :)

## Contents

## Objectives and their weight in scoring your exam

Section 101

| | |
|---|---|
| 1 | BIOS configuration |
| 1 | Modem and sound |
| 1 | Non-IDE storage |
| 3 | PCI cards |
| 1 | Communications devices |
| 1 | USB devices |

Section 102

| | |
|---|---|
| 5 | Hard disk layout |
| 1 | Boot managers |
| 5 | Make and install from source |
| 3 | Shared libraries |
| 8 | Debian package manager OR RPM |

Section 103

| | |
|---|---|
| 5 | Working from the CLI |
| 6 | Processing text streams |
| 3 | File management |
| 5 | Streams, pipes and redirects |
| 5 | Process management |
| 3 | Process priorities |
| 3 | Searching with regular expressions |
| 1 | Vi |

Section 104

| | |
|---|---|
| 3 | Partitions and file systems |
| 3 | File system integrity |
| 3 | Mounting file systems |
| 3 | Quota |
| 5 | File permissions |
| 1 | File ownership |
| 1 | Hard and symbolic links |

Section 105

| | |
|---|---|
| 5 | Install and configure X11 |
| 3 | Setup a display manager |
| 5 | Setup a window manager |

# Stuff for working from the command line

Emacs history keys, versus Vi

| | |
|---|---|
| ^p = k = up | ^n = j = down |
| ^r = ? = search backward | ^s = / = search forward |
| ^a = 0 = beginning of line | ^e = G = end of line |
| ^d = x = delete | ^k = D = delete to end of line |
| ^x = X = backspace | ^t = switch case of char |

^x^e = start line editor from $EDITOR or $FCEDIT.

Options for **`bash`**

History is controlled by the environment variables $HISTFILE, $HISTCMD, $HISTFILESIZE and $HISTCONTROL. $HISTCONTROL can be set to “ignorespace” (to ignore command lines ending in a space) and “ignoredups” (to ignore duplicate command lines).

A few interesting options for the internal **`set -o`** command.

| | |
|---|---|
| emacs / vi | noclobber (disables > redirection) |
| history | noexec (for dry running scripts) |

The prompt

$PS1 is the basic prompt, while $PS2 is the “continuation” prompt (used when spanning a command line across multiple lines of input.

| | |
|---|---|
| \h = hostname | \u = username |
| \H = FQHN | \w = current directory (pwd) |
| \$ = # or $ | \W = base of current directory |

Manual sections

| | | | |
|---|---|---|---|
| 1 | (Shell) commands | 6 | Games |
| 2 | System calls | 7 | Miscelaneous |
| 3 | Library calls | 8 | Sysadmin commands |
| 4 | Device files | 9 | Kernel routines |
| 5 | File formats | | |

Options for **`cat`**

| | |
|---|---|
| -E = show line endings | -S = squeeze blank space |
| -n = show line numbers | -T = show special characters |

## Various text processing commands

join = merge line by line, based on an index (first column)
paste = merge line by line
od = dump octal (-o), hex (x), ascii (a), dec (d), float (f) in a safe way
split = split file into multiple files
tac = reverse cat
tr = translate character. Eg: **tr ‘A-Z’ ‘a-z’ $FILE**
fmt = paragraph formatting (mostly line width)
nl = adds line numbers. –ba includes blank lines
pr = adds headers to documents (for printing). also split into columns.
tee = send output to console and file
script = send input and output to console and file
xargs = send output as argument to $COMMAND
sort = sorting. +n sorts by column n. –n sorts numerically
cut = cut. –c = column/char. –d = delimiter. –f = field
uniq = unique. –u = show single unique. –d = show single dupe.
    -D = show all dupes.
split = -a = number of chars in suffix. –b = amount of bytes.
    -l = amount of lines.
expand = convert tabs to spaces
unexpand = convert spaces to tabs

## Usage of the **sed** command

- **sed s/bob/BOB/ $FILE** = single replace, on each line
- **sed s/bob/BOB/g $FILE** = replace all (global)
- **sed ‘x,y s/bla/BLA/g’** = replace all in lines x through y.
- **sed ‘s/bob/BOB/g ; s/BOB/snuffy/g’ $FILE** = two global replaces, in order inside **sed**.
- **sed –e** does the same: **sed –eCOMM1 –eCOMM2 $FILE.**
- sed –f $SCRIPT $FILE = reads $SCRIPT for commands.
- -n suppresses output of ‘normal’ (unchanged) lines.

## Usage of the **grep** command

-c = count instances
-Cn = show n lines as context
-H = show filename
-h = suppress filename
-i = case insensitive
-v = show unmatched lines
-l = show filename only
-L = show unmatched files
-w = search string as whole word
-r = process all files & dirs in $DIR
-x = only completely matched lines

## Options for **ps**

This command is actually a lot more complex than you’d think initially. It has three modes of operation, based on **$PS_PERSONALITY.**

- In UNIX98 mode one-char flags are used, that are separated using one dash.
- In BSD mode one-char flags are used, without dashes.
- In GNU mode, multiple-char flags are used, that are separated using two dashes.

Options for **ps** (continued)

| | | | |
|---|---|---|---|
| -u, -U | = U | = --User | = for user X |
| -g, -G | = | = --Group | = for group X |
| -H | = f | = --forest | = add hierarchy (kind of like Solaris' pstree). |
| -t | = t | = --tty | = all procs on terminal X |
| p | | = --pid | = only process with PID X |
| -f, -l, j, l, u | | | = additional information |
| -a | = a | | = all tty bound processes |
| -A | = -e | | = all procs on the system |
| -w | = w | | = wide view |
| -x | | | = all my procs |
| -N | | | = negate selection |
| T | | | = all procs on this terminal |
| r | | | = only running processes |

Options for **top**

-b = batch mode -> export all output to a file

During runtime, the following keys can be used.

| | |
|---|---|
| k = kill | s = update rate, per sec. |
| q = quit | P = sort by CPU usage |
| r = renice | M = sort by memory usage |

Nice levels

| | |
|---|---|
| -20 | = highest priority |
| 0 | = default |
| 10 | = default renice |
| 20 | = lowest priority |

**nice –n -10 = nice –10 = renice -10**

## RPM package management

<u>Options for **`rpm`**</u>

| | | |
|---|---|---|
| -i | = --install | = install new package |
| -U | = --upgrade | = upgrade existing, or install new package |
| -F | = --freshen | = upgrade existing package |
| -e | = --erase | = remove installed package |
| -f | = --file | = query package for $FILE on file system |
| -R | = --requires | = list of dependencies |
| -l | = --list | = list of files installed through package |
| -Va | = --verify | = check all packages for changes |
| -Vac | | = check all packages and output to file |
| -h | = --hash | = show progress with hashes |
| -i | | = show package information |
| -b | | = build package (move to rpmbuild) |
| -v | | = verbose |
| --root | | = chroot for install, including the database |
| --test | | = dry run |
| --prefix | | = installation directory |
| --rebuilddb | | = rebuild database from installed packages |
| --import | | = import GPG key into database |
| --checksig | | = verify GPG signature |
| --replacefiles= --force | | = overwrite existing files |
| --nodeps | | = ignore dependencies |
| --allmatches | | = (with erase) erase all with base $NAME |
| --repackage | | = (with erase) erase and recreate RPM |

<u>Query options</u>

| | | |
|---|---|---|
| -Rp | = --requires | = list of dependencies |
| -p | | = query .RPM package file, instead of database |
| -i | | = package info |
| -l | | = file list |
| -c | | = configuration file list |
| -f | | = show package to which $FILE belongs |
| --changelog | | = show changelog |

RPM verification checks the contents of each installed package for changes. All changes are show using nine character flags per file:

| | |
|---|---|
| S = size | L = readLink problem |
| M = mode | U = user |
| 5 = MD5 checksum | G = group |
| D = major/minor number | T = mtime |
| c = configuration files | |

The main RPM configuration file is **`/usr/lib/rpm/rpmrc`**. Do NOT change this file. Instead make modifications to **`/etc/rpmrc`** and/or **`~/.rpmrc`**.

The RPM database is stored in **`/var/lib/rpm`**. Each file in that directory is part of the database (which gets locked during operation).

Options for `rpmbuild`

| | |
|---|---|
| --rebuild | = update dependencies against current libraries |
| -ba | = build :) |

RPM packages are usually built in `/usr/src/redhat`. Subdirectories are:

- BUILD
- RPMS
- SOURCES
- SPECS
- SRPMS

Sections of a Spec file

- preamble: Summary, Name, Version, Release, Copyright, Group, Source, BuildRoot
- %description
- %prep
- %build
- %install
- %clean
- %files
- %doc
- %changelog

## Debian package management

<u>Options for **`dpkg`**</u>

| | | |
|---|---|---|
| -i | = --install | = install new package |
| -r | = --remove | = remove installed package |
| -P | = --purge | = remove everything |
| -p | = --print-avail | = show package information |
| -I | = --info | = show package information for package file |
| -l | = --list | = show all packages with name X |
| -L | = --listfiles | = list of files installed by package |
| -S | = --search | = find package for file on file system |
| -C | = --audit | = trace partials |
| -G | | = upgrade if older exists |
| -E | | = upgrade if newer than current |
| | --root | = chroot for install, including database |
| | --no-act | = dry run |
| | --configure | = rerun post install script |

The configuration for dpkg is stored in **`/etc/dpkg/dpkg.cfg`** and **`~/.dpkg.cfg`**. The file simply contains a list of modifiers.

List of download sites: **`/etc/apt/sources.list`**

The **`dselect`** command starts a CLI menu interface for apt-get. It allows you to configure all kinds of useful stuff.

Synaptic is a X11 GUI replacement for **`dselect`**.

## Compiling and libraries

Sections in a **`Makefile`**

- platform = platform and architecture
- debug = how to handle errors
- optimize = items changed by **`./configure`**
- source = source locations
- targets = all, install, clean, dist and so on

Typical variables for a **`Makefile`**

- install-prefix
- bin_dir
- uparam_dir
- include_dir

A system's main shared libraries are stored in **`/lib`**, **`/usr/lib`** and **`/usr/X11R6/lib`**. Required libraries must be stored in these directories, or in a directory in $LD_LIBRARY_PATH or **`/etc/ld.so.conf`**.

The **`ldd`** command lists an executables required libraries and their location.

Options for **`ldconfig`**

This command builds links in the library directories and **`/etc/ld.so.cache`**.

| | |
|---|---|
| -N | = update links, but do not rebuild cache |
| -X | = rebuild cache, but do not update links |
| -n | = rebuild all, but only with directory X |
| -r | = chroot |
| -p | = current cache |
| -f | = configuration file |
| -C | = cache file |

## Hardware

<u>IRQs</u>
These are defined in **`/proc/interrupts`**.

| | | | | | |
|---|---|---|---|---|---|
| 0 | System timer | 5 | Sound / LPT2 | 10 | Free |
| 1 | Keyboard | 6 | Floppy controler | 11 | Free |
| 2 | Cascade | 7 | LPT1 | 12 | PS2 mouse |
| 3 | COM2/4 | 8 | CMOS Clock | 13 | MPU / FPU |
| 4 | COM1/3 | 9 | free | 14 | ATA1 |
| | | | | 15 | ATA2 |

<u>IO addresses</u>
These are defined in **`/proc/ioports`**.

| | | | |
|---|---|---|---|
| 03F8 | COM1 | 037[8-F] | LPT1 |
| 02F8 | COM2 | 027[8-F] | LPT2 |
| 03E8 | COM3 | 03F[0-7] | FD1 |
| 02E8 | COM4 | 02F[0-7] | FD2 |

<u>DMA channels</u>
These are defined in **`/proc/dma`**.

<u>Hard disks</u>
SCSI and SATA disks do not generally appear in BIOS.

BIOS CHS allowed disks up to 580mb. BIOS with CHS geometry translation allows disks up to 8gb and BIOS with LBA allows disk over 8gb in size.

<u>ISA devices</u>
**`pnpdump`** shows the current configuration
IRQ and IO addresses are set with jumpers on the card
IRQ and IO addr can also be set using **`isapnp`** on a <2.4 kernel.

<u>PCI devices</u>
Plug&Play by nature.

Detection modes are (in order of preference): Any, Direct, MMConfig and BIOS.

- **`setpci`** lets you manually tweak settings.
- **`lspci`** lists all detected and working PCI devices.

Plug and play
Kernel 2.2 allowed for automatic locking of IRQs and DMA channels. Starting from kernel 2.4 automatic locking of IO addresses has also been implemented.

Actually, the kernel isn't "true" P&P. It does read various registers on the devices, but it usually goes through BIOS to make things happen properly.

At boot time, **isapnp** checks the ISA bus (**/etc/isapnp.conf** provides custom mapping to resources) and the PCI bus automatically informs the system.

You can show all information regarding devices on the ISA bus by running the **pnpdump** command. You can then add new stuff to **isapnp.conf**.

Serial ports
**setserial** shows you the current configuration, with the optional –a flag for verbose output.

Two types of UARTS:
- 8250, 16450: max 9600 bps
- 16550, 16550a: max 115.200 bps

Initialization is usually handled by **/etc/rc.local/rc.serial**.

Options for **setserial**

| | |
|---|---|
| $DEV –a | = show all information for $DEV |
| -g /dev/ttyS[0-3] | = show basic information for all devices |
| $DEV $VAR $VAL | = set variable $VAR to value $VAL |

USB bus and devices
- Two USB standards:
- UHCI: Universal Host Controller (Intel)
- OHCI: Open Host Controller (Compaq)

The USB tree in **/proc**:
- **/proc/bus/usb** is the top level.
- **/proc/bus/usb/drivers** is a list of loaded drivers.
- **/proc/bus/usb/devices** is a list of devices.
- **/proc/bus/usb/001** is the first USB controller in the system.
- **/proc/bus/usb/001/001** is the first USB device on the first controller.

Currently most devices require their own driver. Linux is moving towards a virtual file system solution in **/proc** that does away with this requirement. Also, in order to make USB itself work you'll need a few drivers:
- UHCI needs **usbcore.o**, **usb-uhci.o** and **uhci.o**.
- OHCI needs **usbcore.o** and **usb-ohci.o**.

To show the drivers for a certain device:
```
usbmodules –device /proc/bus/usb/xxx/yyy
```

The kernel does not like hot plugging. **usbmgr** detects changes on the USB bus and auto-loads the required modules. **hotplug** is a newer implementation for kernel >2.4 that uses scripts and a configuration stored in **/etc/hotplug**. **hotplug** however has been replaced by **udev**, also known as **usbfs** or **usbdevfs**.

All communications are handled by the controller. Devices cannot communicate amongst themselves.

USB is treated as SCSI by the kernel. The **scsi_info** command shows all USB and SCSI devices. The **usbview** command is an X11 GUI-application that shows comparable information.

<u>The **usbmgr** command</u>
The command (or daemon) is only needed if USB support is implemented in modules, instead of in the kernel itself.

- usbmgr.conf contains data for modules
- preload.conf defines the modules to load at boot time
- requires "START_HOTPLUG=true" at boot time

<u>Sound cards</u>

- **sndconfig**
- **system-config-soundcard** or **redhat-config-soundcard**
- Yast or Yast2
- **alsactl** (use **alsamixer** to unmute)

Your soundcard isn't working properly? It's probably a resource conflict in IRQ or IO address. If the card's working but there's no sound, use **sndconfig**.

<u>SCSI</u>
8 bit SCSI allows up to 8 devices, 16 bit SCSI allows for 16.

IDs are defined in **/proc/scsi/scsi**. ID 7 is always reserved for the controller as it has the highest priority.

Device IDs are based on the SCSI ID, starting from the lowest. It is best not to skip SCSI IDs when assigning because using the skipped on at a later time will bump the rest.

USB is mapped as SCSI. This also influences the bumping of SCSI IDs.

## Resolving hardware conflicts

Check the following files and directories in **`/proc`**: **`ioports`**, **`dma`**, **`interrupt`**, **`usb`**, **`pci`**.

Use the following commands: **`lsmod`**, **`lspci`**, **`lsscsi`**, **`lsdev`**, **`lsraid`**, **`lsusb`**.

## Device types in **`/dev`**

| | |
|---|---|
| hd = IDE disk | lp = parallel |
| sd = SCSI disk | tty = terminal or console |
| scd = SCSI CD | pty = remote terminal |
| st = SCSI tape | ttyS = serial port |
| ht = IDE tape | modem = first modem |
| fd = floppy | cua = communications port |

## The kernel and modules

Environment variables
$MODPATH overrides the contents of **`/etc/modules.conf`**.
$MODULECONF overrides the usage of **`/etc/modules.conf`** entirely.

Information on the **`depmod`** command
**`depmod`** determines all cross-dependencies between loadable modules, to ensure that loading these modules can be an automatic process.

From the **`man`**-page:

> **`depmod`** will not flag an error if a module without a GPL compatible license refers to a GPL only symbol (EXPORT_SYMBOL_GPL in the kernel). However **`insmod`** will refuse to resolve GPL only symbols for non-GPL modules so the actual load will fail.

Files used:

- **`/etc/modules.conf`**
- **`/lib/modules/*/modules.dep,`**
- **`/lib/modules/*`**

-a = update dependency file (run at boot time through an rc-script)
-A = update dependency file, only if there are changed modules
-C = alternate configuration file
-b = base directory for /lib/modules
-r = allow non-root owned modules

Options for **`insmod`**

| | |
|---|---|
| -f = --force | = force if mismatch in kernel and module version |
| -k = --autoclean | = allow kernel to unload unused modules |
| -m= --map | = show modules map, for debugging |
| -n = --noload | = dry run |
| -p = --probe | = verify that module could be loaded |
| -r = --root | = allow non-root owned modules |

Options for **`modprobe`**

| | |
|---|---|
| -a = --all | = load all matching modules, not just the first |
| -c = --showconfig | = show current configuration |
| -d = --debug | = show debugging information for the stack |
| -k = --autoclean | = allow kernel to unload unused modules |
| -n = --show | = dry run |
| -r = --remove | = unload specific module, or autoclean all |
| -t = --type | = load all of this type (part of directory path) e.g. **`–t drivers/net`** |
| -C = --config | = alternate configuration file |

Options for **`rmmod`**

|  |  |
|---|---|
| -a = --autoclean | = mark unused modules for autoclean<br>clean currently marked modules |
| -r = --stacts | = remove a complete modules stack |

Options for **`lsmod`**

From the manpage:

> Shows information about all loaded modules.
> The format is name, size, use-count, list of referring modules. The information displayed is identical to that available from **`/proc/modules`**.
> If the module controls its own unloading via a "can_unload" routine then the user count displayed by **`lsmod`** is always -1, irrespective of the real use count.

# Working with disks, file systems and such

Options for **`fdisk`**

| | |
|---|---|
| -b = sectorsize | -u = show sizes in cylinders (with l) |
| -l = show partition table | -s = print size in blocks of partition |
| -v = print version number | |

Menu options for **`fdisk`**

| | | |
|---|---|---|
| p = print | d = delete | a = make bootable |
| n = new | t = type | m = help |
| q = quit | l = list of types | w = write partition table |

Partition type 0x82 is used for both an active Solaris x86 partition and a Linux swap partition. This may lead to problems when dual booting the two OSes.

Other partition codes are:
0x0F = extended partition
0x05 = extended partition
0x83 = Linux

**`fdisk –l`** shows you a list of **`/dev`** entries for your storage devices. These can then be used in **`/etc/fstab`**.

hda[1-4] are primary partitions, while hda[5-n] are extended partitions.

Menu options for **`parted`**

| | |
|---|---|
| ? = help | rm |
| print | move |
| mkpart | resize |

Other partitioning tools
Debian and its derivatives use **`cfdisk`** and Red Hat uses Disk Druid.

Partitioning tips
Always keep PIBS in mind: Performance, Integrity, Backup, Security.

- Move heavy stuff to a non-OS disk.
- Put some distance between risky stuff and the OS.
- It's easier to make full file system backups.
- Isolation and jailing.

Bootable partitions must always be made before the 1024th cylinder of the boot disk. They must be of the "primary" type.

## Layout of **`/etc/fstab`**

1. block device or remote file system
2. mount point
3. type of file system
4. mount options
5. dump (file ssytem backup): yes or no?
6. fsck pass

## Making CD-ROMs

Use **`mkisofs`** and **`cdrecord`**.

## Making a boot floppy

- Red Hat: **`dd if=/mnt/cdrom/images/boot.img of=/dev/fd0`**
- Win98: **`d:\dosutils\rawrite.exe d:\images'bootdisk.img`**
- WinNT: **`d:\dosutils\rawritewin.exe d:\images'bootdisk.img`**

## Swap space

- **`mkswap`**
- **`swapon`**

Kernel <2.4.10 allows for eight swap files and devices, while all kernels above that version allow for thirty-two. If you can, spread these files and devices across multiple disks.

## LILO versus Grub

| | |
|---|---|
| **`/etc/lilo.conf`** | **`/boot/grub/grub.conf`** **`/boot/grub/menu.lst`** |
| reconfig = reinstall | reads config file every time. |

## LILO configuration

| | |
|---|---|
| timeout | = N times 1/10 of a second |
| default | = default booted OS |
| linear | = use when booting from SCSI |
| lba32 | = use when the boot partition is above the 1024cyl limit |

In the MBR, LILO only has space for 512 characters to store its configuration.

## LILO boot commands

If LILO runs in GUI mode, you may need to press ^x before entering the commands.

| | |
|---|---|
| linux $n | = boot into run level $n |
| root | = set root device |
| mem $n | = limit RAM to $n bytes |
| maxcpus $n | = limit the amount of active cores to $n |
| image | = path to kernel |

## Options for `lilo`

| | |
|---|---|
| -b = boot device | -q = query/boot/map for all boot-able kernels |
| -C = configuration file | |
| -D = boot $label | -R = pass commands to next boot |
| -l = switch to linear mode | -s = turn on backup copying |
| -L = switch to LBA32 mode | -t = dry run |
| -u = restore backed up MBR file | -T = show info (**`geom`**, **`video`**) |
| -A = either query active partition, or set it to N. | |

## Setting up `grub`

Grub auto-detects whether you're using SCSI or IDE.

Set things up using either:

```
$ grub-install –root-directory=/boot /dev/hda
$ grub
grub> root (hd0,1)
grub> setup (hd0)       # in case of MBR
```

or

```
grub> setup (hd0.1)     # in case of hda1
```

If you do not know which partition contains the boot the appropriate boot information, run:

```
grub> find /boot/grub/stage1
```

## Options for `dumpe2fs`

Gives you verbose information on a selected file system. The –h flag makes the command omit group descriptors; you're better off using this all the time.

## Options for `tune2fs`

| | |
|---|---|
| -c = maximum mount count | -i = check interval (d/w/m) |
| -C = force count to $num | -j = convert to ext3 (journaling) |
| -m = set reserved percentage | -r = set reserved blocks |
| -J = journal options | -l = list contents of super block |
| -L = set volume label | |

## Journal options for `tune2f`

| | |
|---|---|
| size | Create "inline" journal X blocks in size, where 1024 < X < 102400. |
| device | Device or label of external journal device. |

## Options for `debugfs`

This command lets you interactively hack the file system.

-w = open FS in rw-mode.
-c = open FS in "catastrophic" mode: read-only and disregards inodes.
-i = target is an ext2 image file
-s = read from superblock N. Requires –b $blocksize.

## Commands for **debugfs**

| | |
|---|---|
| cat $file | = dump contents of file |
| [show_super_]stats | = super block information |
| stat $file | = give inode information |
| undelete $inode $name | = undelete inode X to file Y |
| list_deleted_inodes = lsdel | = list of deleted files |
| write $int_file $ext_file | = extract $int to main file system $ext. Very handy in case of file system corruption |
| dump $int_file $ext_file | = dump contents of file to file |
| list_requests = lr = ? = help | = the help function |
| kill $file | = clear inode contents (does not affect the directory it's part of) |
| quit | = exit from **debugfs** |

## Extra information on EXT2

A super block is 36 bytes in size and occurs every 8192 blocks of a file system. It contains the FS size, its location, the inode count and the cylinder and block usage.

The **stat** command shows the inode contents for $FILE. The inode contains all information about a file, except its name. An inode points to a disk block, which is part of an eight block "block group".

## Options for **mkfs**

| | |
|---|---|
| -t = type | -m = set reserved space percentage |
| -c = sector and block check | |

Using the **mke2fs** command you can decide the block size for a file system.

| | | |
|---|---|---|
| -T news | 1024B block | 1 inode per 4 kB block |
| -T largefile | 2048B block | 1 inode per 1 MB |
| -T largefile4 | 4096B block | 1 inode per 4 MB |

## Options for **mke2fs**

| | |
|---|---|
| -j = convert to ext3 | -J = journal options (see tune2fs) |
| -l = read bad blocks list | -L = set volume label |
| -n = dry run | |
| -M = set the "last mounted" dir | |

-S = write super blocks, without touching anything else. Requires the correct blocksize and should be followed by e2fsck.

Options for **`fsck`**

| | |
|---|---|
| -A = all checkable file systems | -N = dry run |
| -C = progress indicator (e2fs only) | -t = type |
| -V = verbose | -t no $type = skip all of $type |
| -a = non-interactive | |

Fsck goes over your file systems in five passes.

1. inodes, blocks and file sizes
2. directory structure
3. directory connectivity
4. reference counts
5. group summary info

Options for **`mount`**

| | |
|---|---|
| loop | = for disk images |
| noauto | = disable auto-mounting in **`fstab`** |
| user | = any user can mount |
| users | = any user can mount and unmount |
| owner | = owner of device file can mount |
| remount | = set options without manually re-mounting |
| ro | = read only |
| rw | = read and write |
| usrquota | = enable quotas for users |
| grpquota | = enable quotas for groups |
| username | = username for a share |
| password | = password for a share |
| credentials | = a file containing username and password for the share in question. Safer alternative to using "username=" and "password=". |

Instead of device IDs, you can also use partition labels (as assigned using **`fdisk`**) to identify which partition to mount.

Info for **`umask`**

- For files, substract from 666.
- For directories, substract from 777.
- This is different from Solaris...

The **`chattr`** command

Use it to add (+) or remove (-) attributes from a file.

| | |
|---|---|
| a = append only | s = delete securely |
| c = compressed | t = no tail-merging |
| i = immutable | A = no atime updates |
| j = data journaling | d = exclude from dump execution |
| u = undeletable | S = sync immediately |

### Information on quota

**`quotaon, quotaoff`** = turn on/off quota
**`quotacheck`** = update config files
**`edquota, quota`** = edit / show quota
**`edquota -t`** = set soft limit grace period
**`edquota -a`** = report on all user’s quota
**`edquota $user`** = edit $user’s quota
**`aquota.user`** = config file for $DIRECTORY
**`aquota.group`** = config file for $DIRECTORY
**`usrquota, grpquota`**= used in **`/etc/fstab`** to check config files

### Disk usage

du –x = do not cross file system boundaries
df –l = only local file systems
-h = human readable output (on both commands)

### File access times

atime = last read/written
ctime = altered permissions, name, owner
mtime= altered file contents

**`touch –t yyyymmddhhmm $FILE`** = set last mtime.
**`touch –r $FILE2 $FILE1`** = sync $FILE2’s mtime to the other file.

### Options for **`cp`**

-d = copy link, not target
-l = create hard link
-s = create symlink
-u = update if source is newer

The longest command option in Linux, ever
**`rmdir ignore-fail-on-non-empty $DIR`**

### Options for **`getfacl`**

-a = show file name, owner, group and current ACL
-d = show file name, owner, group and default ACL

### Options for **`setfacl`**

-r = recalculate the ACL mask from the current configuration
-f = read configuration from $FILE (a dash indicates stdin)
-d = delete one or more entries
-m = add or modify one or more entries
-s = set ACL (completely overwrites current configuration)

Rules for ACLs

Required entries

- Exactly one user entry specified for the file owner.
- Exactly one group entry for the file group owner.
- Exactly one other entry specified.

If there are additional user and group entries:

- Exactly one mask entry specified for the ACL mask that indicates the maximum permissions allowed for users (other than the owner) and groups.
- Must not be duplicate user entries with the same uid.
- Must not be duplicate group entries with the same gid.

If file is a directory, the following default ACL entries may be specified:

- Exactly one default user entry for the file owner.
- Exactly one default group entry for the file group owner.
- Exactly one default mask entry for the ACL mask.
- Exactly one default other entry.

There may be additional default user entries and additional default group entries specified, but there may not be duplicate additional default user entries with the same uid, or duplicate default group entries with the same gid.

Example:

```
setfacl -m user:hope:r-- myfile.txt
```

## X11, XFree86 et al

| Configuration tools | XF 3.x | XF 4.x | X.org | GUI |
|---|---|---|---|---|
| **Xf86config** | X | | | |
| **Xconfigurator** | X | X | | |
| **XF86Setup** | X | | | X |
| `X server itself` | | X | X | |
| **System-config-xfree86** | | X | | |
| **Yast** and **Yast2** | | X | | |
| **Xf86cfg** | | X | X | X |
| **Xorgcfg** | | X | X | X |

| Configuration file | XF 3.x | XF 4.x | X.org |
|---|---|---|---|
| **/etc/X11/XF86Config** | X | X | |
| **/etc/X11/XF86Config-4** | | X | |
| **/etc/X11/xorg.conf** | | | X |

Configuration file sections, v3 versus v4

| V3 | V4 |
|---|---|
| Files | Files |
| Serverflags | ServerLayout |
| Keyboard | InputDevice |
| Pointer | |
| Monitor | Monitor |
| Device | Device |
| Screen | Screen |

Interesting parameters from the config files

- Files: RgbPath, FontPath
- InputDevice: ID, Driver, Protocol, Device, ZAxisMapping, Emul3Buttons

Other configuration files

- xinitrc: setup for X11 environment
- XClients: start window manager (if it fails, default to twm)
- Xsession: start programs, desktop and WM preferences
- Xresources: program defaults and customization
- Xdefaults: same as Xresources

Drivers for video cards are stored in **/usr/X11R6/lib/modules/drivers**. Filenames are made thusly: **$name_drv.o**.

The **xvidtune** command can be used to tweak sync rates and alignment. This can only be done if "DisableVidModeExtensions" is set to "no" in the XFree86 configuration. It can also literally blow up your display.

Fonts are stored in **/usr/X11R6/lib/X11/fonts** or **/usr/share/fonts**.
Postscript fonts are *.pfa and *.pfb.
TrueType fonts are *.ttf.
FontForge can be used to convert Mac OS suitcase fonts for Linux.

After adding fonts to a directory, run **mkfontscale** and **mkfontdir**, which creates **fonts.scale** (outline fonts) and **fonts.dir** (outline + bitmap). You can also create **fonts.alias**, which maps missing fonts to alternatives.

The X configuration file has directives called FontPath. Add one in the section "Files" for each font directory, by order of preference. You can add ":unscaled" to the path to only match bitmap fonts of the appropriate dimensions. Feel free to repeat the same path without the flag at the bottom of the list.

You can dynamically set your X font path: **xset fp+ $FONTPATH**. "fp+" adds the path at the end of the list, while "+fp" adds it to the front. After expanding your path, run: **xset fp rehash**. You can reset your font path to its default setting using: **xset fp default**.

The XFS font server usually runs on TCP port 7100. Its configuration directory is **/etc/X11/fs/config**. The **catalogue** file contains a comma separated list of font paths that it should serve. After making changes to the file, run: **xfs restart**. Then restart X or rehash its font path.

You can access your local XFS by setting your FontPath to either of these:
- **FontPath "unix/:7100"**
- **FontPath "unix/-1"**

Or a remote server:
- **FontPath "tcp/192.168.0.20:7100"**

FreeType is better than the X core fonts in many ways. It only works with local fonts though. **/etc/fonts/local.conf** contains a list of font directories. Each line is of the form: **<dir>$PATH</dir>**. FreeType makes use of a cache that can be rehashed using: **sudo fc-cache**.
Window managers are usually called from **/usr/lib/X11/xinit/xinitrc**. Before that come **~/.Xclients** and **/etc/X11/xinit/XClients**.

Popular window managers:

| | | |
|---|---|---|
| Afterstep | E(nlightenment) | ICEWM |
| Black box | FVWM / FVWM2 | Sawfish |

There are multiple ways to determine which XDMCP server to run:
- **inittab**
- SysV init script
- **prefdm** (/etc/sysconfig/desktop)

XDM configuration: **/etc/X11/xdm/xdm-config**. In order to enable remote connections you will need to set "DisplayManager.requestPort" to "177" instead of "0".
- Security is handled through **/etc/X11/xdm/Xaccess** and **./Xservers**.
- Environment settings are handled in **/etc/X11/xdm/Xresources**.

KDM uses the same config files as XDM, with the addition of **`~/.kdmrc`**.

GDM uses **`/etc/X11/gdm/gdm.conf`**, which has a format similar to **`kdmrc`**. You can also use the GUI configuration tool **`gdmconfig`**.

Each user also has his own config files: **`~/.Xresources`** (for startup settings) and **`~/.Xdefaults`** (are used at any time). The format of these files is as follows: $PROG*$RESOURCE: $VALUE.

For example:

```
XTerm*Background: linen
XTerm*saveLines: 1000
XTerm*geometry: +50+100
```

Unfortunately, resources are not 100% standardized across applications. You will need to consult the application's manual to make sure you're changing the right parameters. Many modern applications completely forego the use of Xresources, in favor of their own config file.

Added security, across networks.

- GDM: **`/etc/X11/gdm/gdm.conf`** -> DisallowTCP=true.
- KDM: **`/etc/X11/xdm/Xservers`** -> -nolisten tcp
- XDM: see KDM.
- Startx: **`/usr/bin/startx`** -> -nolisten tcp
- xhost +$HOST1 $HOST2 ... $HOSTn = allow
  xhost -$HOST1 $HOST2 ... $HOSTn = revoke
  xhost = show current list

<u>Other options for **`xset`**</u>
All values will be reset to default when you log out.

| | |
|---|---|
| -display | = set X server to use |
| [+\|-]dpms | = enable/disable Energy Star features |
| dpms X Y Z | = set timeout in seconds for standby, suspend and off<br>a value of "0" disables the specific feature |
| led N | = turn on LED number N on the keyboard |
| -led N | = turn off LED number N on the keyboard |
| m | = set mouse acceleration and threshold |
| r | = autorepeat value for the keyboard |
| s | = screen saver parameters |
| q | = show list of current settings |

<u>Options for **`startx`**</u>
A double dash is the delimiter between client and server settings. The system-wide **`xinitrc`** and **`xserverrc`** files are found in the /usr/lib/X11/xinit directory.

Example server settings:

- -depth 16
- -dpi 100
- -layout Multihead

<u>Example Xinitrc file</u>
```
xrdb -load $HOME/.Xresources
xsetroot -solid gray &
xbiff -geometry -430+5 &
oclock -geometry 75x75-0-0 &
xload -geometry -80-0 &
xterm -geometry +0+60 -ls &
xterm -geometry +0-100 &
xconsole -geometry -0+0 -fn 5x7 &
exec twm
```

## Networking

NIC configuration at boot time

This is done using **`/etc/sysconfig/network-scripts/ifcfg-$NIC`**.

Important parameters are:

| | |
|---|---|
| DEVICE | IPADDR |
| ONBOOT | NETMASK |
| BOOTPROTO | GATEWAY |

Options for **`arp`**

If an IP address is remote, the MAC address in your ARP cache will be the MAC address of the router that's the first hop along the way.

| | |
|---|---|
| -s $IP $MAC | = set static binding |
| -a | = show full cache |
| -d $IP | = remove entry from cache |

The PPP protocol

The serial connection is actually made using the **`chat`** command. The parameters come in input-reply pairs.

```
chat " " ATZ OK ATDT5558080 CONNECT " " login: $username word:
$password
```

**`Pppd`** uses the serial connection to pass through IP traffic.

```
pppd /dev/cua1 57600 crtscts defaultroute
```

Combining the two

```
pppd connect "chat –f $SCRIPT" /dev/cua1 57600 crtscts modem
defaultroute
```

Options for **`pppd`**

All these settings are made in **`/etc/ppp/options`**.

| | |
|---|---|
| demand | = connect if traffic needs to be sent to the other side |
| holdoff | = wait after a disconnect |
| _detach | = no fork |
| modem | = use modem control lines |
| lock | = lock serial device |
| crtscts | = use hardware flow control |
| defaultroute | = set as default gateway |
| asyncmap 0 | = don't use escaped control sequences |
| mtu / mru | = transmit and receive windows |
| name username | = set hostname to username |
| +papcrypt | = use PAP encryption |

- PAP = clear text, with a key-pair in **`/etc/ppp/pap-secrets`**.
- CHAP = encrypted, uses **`/etc/ppp/chap-secrets`** and uses hostname+random string for encryption.
- MSCHAP = identical to CHAP, but a special Microsoft implementation

Option for **`ifconfig`**

| | |
|---|---|
| [-]arp | = enable or disable the ARP protocol for this interface |
| [-]promisc | = when enabled, all packets on the network will be received |
| [-]allmulti | = when enabled, all multicast traffic will be received |
| mtu N | = set MTU |
| dstaddr $A | = set destination address for PTP connection (obsolete) |
| add \| del | = add or remove an IPv6 address to the interface |
| tunnel $A | = create IPv6-in-v4 SIT tunnel to destination |
| irq $A | = set IRQ |
| ioaddr $A | = set base IO address |
| media | = set media type |
| [-]pointtopoint $A | = create PTP connection to destination address |
| hw $class $A | = set hardware address for type $CLASS |
| mem_start $A | = set base shared memory address |

## Using the vi editor

Movement

|  |  |
|---|---|
| ^F = page forward | ^D = ½ page forward |
| ^B = page backward | ^U = ½ page backward |

Deletion

|  |  |
|---|---|
| D | = delete from cursor to end of line |
| dL | = delete from cursor to end of screen |
| dG | = delete from cursor to end of file |
| d^ | = delete from beginning of line to cursor |

Vi has 26 named buffers: ”a” through ”z” and ”A” through ”Z”. You can use these for yanking and pasting, like so: **`a3yy`**. Using the lower case letters replaces the contents of the buffer, while using the upper case letter appends to the existing contents. Thus: **`ayy`**, followed by **`A3yy`** results in buffer “a” containing four lines of text.

Other keys and commands

|  |  |
|---|---|
| ^G | = show status line |
| :+E+! | = reset to initial state after last write |
| :wq | = :x = ZZ = write and close immediately |
| ? and / | = search backward and forward |
| N and n | = repeat search backward and forward |
| %s | = search whole file |
| J | = join line with next line |
| :split and :vsplit | = split screen for same file |
| :split $FILE | = split screen with second file, $FILE |
| ^WW | = switch pane in split screen mode |
| :close | = close active pane |
| :only | = close all but active pane |
| :vimdiff $F1 $F2 | = auto-split for two files, with colors for diff |

Two files can be used to set options for vi: **`/etc/exrc`** and **`~/.exrc`**.

You can actually use vi to browse your file system, just edit a directory. Selecting a contained directory and pressing [Enter] takes you into that directory.

Other **`vi`** options

| | |
|---|---|
| - | = read $file from stdin and commands from stderr |
| +$N | = open $file at line $N |
| +/$S | = open $file at first instance of string $S |
| +"$C" | = -c "$C" = --cmd "$C" |
| | = run command $C after opening (up to ten commands) |
| -b | = binary mode, to edit binary files safely |
| -n | = do not use swap file |
| -r | = -L = list swap files and their state |
| -r $file | = recover from swap file |
| -s $script | = use $script as input for commands |
| -w $script | = send all commands to $script, to create a script |
| -x | = use encryption when saving |
| -- | = delimiter between options and file names |

## Various other things

Typical consoles during installation

| | | | |
|---|---|---|---|
| 1 | Installation CLI | 5 | Miscellaneous messages |
| 2 | Shell prompt | 6 | - |
| 3 | Installation log | 7 | Installation GUI |
| 4 | System messages | | |

Regular expressions

- \<... = word beginning with
- ...\> = word ending with
- [^...] = not at this position
- ^... = line beginning with
- ...$ = line ending with

The **`newgrp`** command
Use this command to switch your personal active group. Use the –l flag to go through the login process as being part of this group.

The **`date`** command
Setting the date: **`date 0701040007`** (MMddhhmmyy)
Setting the date: **`date –s "Sat Jul 1 04:00:00 CET 2007"`**
Showing the date: **`date +%Y-%d-%m`**

The **`hwclock`** command

| | |
|---|---|
| -r | = read |
| --systohc | = set hardware clock to system date |
| --hctosys | = set system date to hardware clock |

The **`initrd`** RAM disk
Shamelessly copied from the man-page:

> The special file /dev/initrd is a read-only block device. Device /dev/initrd is a RAM disk that is initialized (e.g. loaded) by the boot loader before the kernel is started. The kernel then can use the the block device /dev/initrd's contents for a two phased system boot-up.
>
> In the first boot-up phase, the kernel starts up and mounts an initial root file-system from the contents of /dev/initrd (e.g. RAM disk initialized by the boot loader). In the second phase, additional drivers or other modules are loaded from the initial root device's contents. After loading the additional modules, a new root file system (i.e. the normal root file system) is mounted from a different device.

<u>Options for **find**</u>

| | |
|---|---|
| -maxdepth | Descend at most N levels |
| -mount | Don't go into other file systems |
| -mindepth | Only start searching below the Nth level |
| | |
| -[a\|c\|m]min | File was accessed/changed/modified N minutes ago |
| -[a\|c\| ]newer | File access/change/modified is newer than $file |
| -[a\|c\|m]time | File was accessed/changed/modified N days ago |
| -empty | File is an empty directory or file |
| -gid, -group | File's group is... |
| -[i]name | File's name is... [case insensitive] |
| -[i]regex | File's name is... [case insensitive] |
| -perm | File's permissions are exactly... |
| -size | File size |
| -type | Type (block special, char special, directory, named pipe, regular file, symlink, socket) |
| -uid, -user | File's owner is... |
| | |
| -exec | Execute command for each result |
| -ok | Like –exec, but asks before each command |
| -print | Show full file name on stdout |

<u>File System Hierarchy (FHS) 2.2</u>

| | |
|---|---|
| /bin | = Essential user command binaries (for use by all users) |
| /boot | = Static files of the boot loader |
| /dev | = Device files |
| /etc | = Host-specific system configuration |
| /home | = User home directories (optional) |
| /lib | = Essential shared libraries and kernel modules |
| /lib<qual> | = Alternate format essential shared libraries (optional) |
| /mnt | = Mount point for a temporarily mounted filesystem |
| /opt | = Add-on application software packages |
| /root | = Home directory for the root user (optional) |
| /sbin | = System binaries |
| /tmp | = Temporary files |

<u>Random</u>

- **updatedb = slocate –u**
- To ignore aliases and force the real binary: **\$COMMAND**
- **xwininfo** shows a window's geometry, color depth and other stuff
- [ctrl][alt][+] or [-] = go up/down one resolution within current color depth