— LPIC General Linux Part 2 —

(Study Notes) [1] [2]

geoffrey hector robertson
geoffrey@zip.com.au

July 21, 2005

[2]RCS Id = Id: lpic.general-linux-1.notes.tex,v 1.2 2002/02/28 23:06:04 geoffrey Exp geoffrey

2

Contents

Topic 101

Hardware & Architecture

Objective 101.1

Configure Fundamental BIOS Settings

General Linux Part 2 []

The information in this section is taken from the LPI Project Objective Management System and is copyright to the Linux Professional Institute. It should be noted that the latest version of this information will be found at:
http://www.lpi.org/cgi-bin/poms.py.

A note about the numbering

- The numbers hard on the left margin with one dot, e.g. 1.103, are LPI Exam Topic numbers.

- The numbers in parentheses next to the LPI Exam Topic numbers are the old topic numbers and may be used to reference topics in older books and documentation.

- The numbers hard on the left margin with two dots, e.g. 1.103.1, are LPI Exam Objectives.

- The numbers in square brackets to the right of Topics and Objectives indicate weightings.

0.101 (1.1) Hardware & Architecture [11]

0.101.1 Configure Fundamental BIOS Settings [3]

Statement of Objective:

Candidates should be able to configure fundamental system hardware by making the correct settings in the system BIOS. This objective includes a proper understanding of BIOS configuration issues such as the use of LBA on IDE hard disks larger than 1024 cylinders, enabling or disabling integrated peripherals, as well as configuring systems with (or without) external peripherals such as keyboards. It also includes the correct setting for IRQ, DMA and I/O addresses for all BIOS administrated ports and settings for error handling.

Key files, terms, and utilities include:

```
/proc/ioports
/proc/interrupts
/proc/dma
```

```
/proc/pci
```

Resources of interest

Large Disk HOWTO by Andries Brouwer

```
http://www.linuxdoc.org/HOWTO/Large-Disk-HOWTO.html
```

0.101.2   Setup SCSI devices [2]

Statement of Objective:

Candidates should be able to configure SCSI devices using the SCSI BIOS as well as the necessary Linux tools. They also should be able to differentiate between the various types of SCSI. This objective includes manipulating the SCSI BIOS to detect used and available SCSI IDs and setting the correct ID number for different devices especially the boot device. It also includes managing the settings in the computer's BIOS to determine the desired boot sequence if both SCSI and IDE drives are used.

Key files, terms, and utilities include:

```
SCSI ID
/proc/scsi/
scsi_info
```

Resources of interest

0.101.3   Setup different PC expansion cards [2]

Statement of Objective:

Candidates should be able to configure various cards for the various expansion slots. They should know the differences between ISA and PCI cards with respect to configuration issues. This objective includes the correct settings of IRQs, DMAs and I/O Ports of the cards, especially to avoid conflicts between devices. It also includes using isapnp if the card is an ISA PnP device.

Key files, terms, and utilities include:

```
/proc/dma
/proc/interrupts
/proc/ioports
/proc/pci
pnpdump(8)
isapnp(8)
lspci(8)
```

Resources of Interest

Linux Hardware Compatibility HOWTO - Steven Pritchard :

```
http://www.linuxdoc.org/HOWTO/Hardware-HOWTO/index.html
```

Linux PCI-HOWTO by Michael Will :

```
http://www.linuxdoc.org/HOWTO/PCI-HOWTO.html
```

Plug-and-Play-HOWTO David S.Lawyer :

```
http://www.linuxdoc.org/HOWTO/Plug-and-Play-HOWTO.html
```

0.101.4 Configure communication devices [2]

Statement of Objective:

Candidates should be able to install and configure different internal and external communication devices like modems, ISDN adapters, and DSL switches. This objective includes verification of compatibility requirements (especially important if that modem is a winmodem), necessary hardware settings for internal devices (IRQs, DMAs, I/O ports), and loading and configuring suitable device drivers. It also includes communication device and interface configuration requirements, such as the right serial port for 115.2 Kbps, and the correct modem settings for outbound PPP connection(s).

Key files, terms, and utilities include:

```
/proc/dma
/proc/interrupts
/proc/ioports
setserial(8)
```

Resources of Interest

Linmodem-HOWTO by Sean Walbran and Marvin Stodolsky :

```
http://www.linuxdoc.org/HOWTO/Linmodem-HOWTO.html
```

Modem-HOWTO - David S.Lawyer :

```
http://www.linuxdoc.org/HOWTO/Modem-HOWTO.html
```

The Winmodems-and-Linux HOWTO by Alexandre J. :

```
http://www.linuxdoc.org/HOWTO/Winmodems-and-Linux-HOWTO.html
```

Serial HOWTO - David S.Lawyer original by Greg Hankins :

```
http://www.linuxdoc.org/HOWTO/Serial-HOWTO.html
```

The Linux Winmodem Support Website :

```
http://www.linmodems.org
```

0.101.5   Set Up USB devices [2]

Statement of Objective:

Candidates should be able to activate USB support, use and configure different USB devices. This objective includes the correct selection of the USB chipset and the corresponding module. It also includes the knowledge of the basic architecture of the layer model of USB as well as the different modules used in the different layers.

Key files, terms, and utilities include:

```
lspci(8)
usb-uhci.o
usb-ohci.o
/etc/usbmgr/
usbmodules
/etc/hotplug
```

Resources of Interest

The Linux-USB Project `http://www.linux-usb.org:`

0.102   (2.2) Linux Installation & Package Management [24]

0.102.1   Design hard disk layout [2]

Statement of Objective:

Candidates should be able to design a disk partitioning scheme for a Linux system. This objective includes allocating filesystems or swap space to separate partitions or disks, and tailoring the design to the intended use of the system. It also includes placing /boot on a partition that conforms with the BIOS' requirements for booting.

Key files, terms, and utilities include:

```
/ (root) filesystem
/var filesystem
/home filesystem
swap space
mount points
partitions
cylinder 1024
```

Resources of Interest:

Mini-FAQ from Karsten M Self on Linux Partitioning ：

```
http://pw1.netcom.com/~kmself/Linux/FAQs/partition.html
```

0.102.2   Install a boot manager [3]

Statement of Objective:

Candidate should be able to select, install, and configure a boot manager. This objective includes providing alternative boot locations and backup boot options (for example, using a boot floppy).

Key files, terms, and utilities include:

```
/etc/lilo.conf
/boot/grub/grub.conf
lilo
grub-install
MBR
superblock
first stage boot loader
```

Resources of Interest:

LinuxGazette GRUB Article ：

```
http://www.linuxgazette.com/issue64/kohli.html
```

The Gnu Grub Site ：

```
http://www.gnu.org/software/grub/
```

0.102.3 Make and install programs from source [5]

Statement of Objective:

Candidates should be able to build and install an executable program from source. This objective includes being able to unpack a file of sources. Candidates should be able to make simple customisations to the Makefile, for example changing paths or adding extra include directories.

Key files, terms, and utilities include:

```
gunzip
gzip
bzip2
tar
configure
make
```

Resources of Interest:

TBA

0.102.4 Manage shared libraries [3]

Statement of Objective:

Candidates should be able to determine the shared libraries that executable programs depend on and install them when necessary. Candidates should be able to state where system libraries are kept.

Key files, terms, and utilities include:

```
/etc/ld.so.conf
LD_LIBRARY_PATH
```

Resources of Interest:

Shared-Library HOWTO Used to be at:

    http://www.linux.com/howto/Program-Library-HOWTO/shared-libraries.htm

0.102.5 Use Debian package management [5]

Statement of Objective:

Candidates should be able to perform Debian package management.This objective includes being able to use command-line and interactive tools to install, upgrade, or uninstall packages, as well as find packages containing specific

files or software (such packages might or might not be installed). This objective also includes being able to obtain package information like version, content, dependencies, package integrity and installation status (whether or not the package is installed).

Key files, terms, and utilities include:

```
/etc/dpkg/dpkg.cfg
/var/lib/dpkg/*
/etc/apt/apt.conf
/etc/apt/sources.list
dpkg
dselect
dpkg-reconfigure
apt-get
alien
```

Resources of Interest:

The Debian GNU/Linux FAQ :

```
http://www.debian.org/doc/FAQ/index.html
```

0.102.6   Use Red Hat Package Manager (RPM) [6]

Statement of Objective:

Candidates should be able to perform package management under Linux distributions that use RPMs for package distribution. This objective includes being able to install, re-install, upgrade, and remove packages, as well as obtain status and version information on packages. This objective also includes obtaining package information such as version, status, dependencies, integrity, and signatures. Candidates should be able to determine what files a package provides, as well as find which package a specific file comes from.

Key files, terms, and utilities include:

```
/etc/rpmrc
/usr/lib/rpm/*
```

Resources of Interest:

TBA

0.105 (1.5) Kernel [7]

0.105.1 Manage/Query kernel and kernel modules at runtime [3]

Statement of Objective:

Candidates should be able to manage and/or query a kernel and kernel loadable modules. This objective includes using command-line utilities to get information about the currently running kernel and kernel modules. It also includes manually loading and unloading modules as appropriate. It also includes being able to determine when modules can be unloaded and what parameters a module accepts. Candidates should be able to configure the system to load modules by names other than their file name.

Key files, terms, and utilities include:

```
/lib/modules/kernel-version/modules.dep
/etc/modules.conf & /etc/conf.modules
depmod
insmod
lsmod
rmmod
modinfo
modprobe
uname
```

Resources of Interest:

TBA

0.105.2 Reconfigure, build, and install a custom kernel and kernel modules [4]

Statement of Objective:

Candidates should be able to customise, build, and install a kernel and kernel loadable modules from source This objective includes customising the current kernel configuration, building a new kernel, and building kernel modules as appropriate. It also includes installing the new kernel as well as any modules, and ensuring that the boot manager can locate the new kernel and associated files (generally located under /boot, see objective 1.102.2 for more details about boot manager configuration).

Key files, terms, and utilities include:

```
/usr/src/linux/*
/usr/src/linux/.config
/lib/modules/kernel-version/*
/boot/*
```

```
make
make targets: config, menuconfig, xconfig, oldconfig, modules, install, modules_ins
```

Resources of Interest:

TBA

0.107 (1.7) Printing [6]

0.107.1 Manage printers and print queues [2]

Statement of Objective:

The candidate should be able to manage print queues and user print jobs. This objective includes monitoring print server and user print queues and troubleshooting general printing problems

Key files, terms, and utilities include:

```
lpc
lpq
lprm
lpr
/etc/printcap
```

Resources of Interest:

TBA

0.107.2 Print files [1]

Statement of Objective:

Candidates should be able to manage print queues and manipulate print jobs. This objective includes adding and removing jobs from configured printer queues and converting text files to postscript for printing.

Key files, terms, and utilities include:

```
lpr
lpq
mpage
```

Resources of Interest:

TBA

0.107.3   Install and configure local and remote printers [3]

Statement of Objective:

Candidate should be able to install a printer daemon, install and configure a print filter (e.g.: apsfilter, magicfilter). This objective includes making local and remote printers accessible for a Linux system, including postscript, non-postscript, and Samba printers.

Key files, terms, and utilities include:

```
lpd
/etc/printcap
/etc/apsfilter/*
/var/lib/apsfilter/*/
/etc/magicfilter/*/
/var/spool/lpd/*/
```

Resources of Interest:

TBA

0.109   (1.9) Shells, Scripting, Programming, Compiling [9]

0.109.1   Customise and use the shell environment [4]

Statement of Objective:

Candidate should be able to customise shell environments to meet users' needs. This objective includes setting environment variables (e.g. PATH) at login or when spawning a new shell. It also includes writing bash functions for frequently used sequences of commands.

Key files, terms, and utilities include:

```
~/.bash_profile
~/.bash_login
~/.profile
~/.bashrc
~/.bash_logout
~/.inputrc
function (Bash built-in command)
export
env
set (Bash built-in command)
unset (Bash built-in command)
```

Resources of Interest:

TBA

0.109.2   Customise or write simple scripts [5]

Statement of Objective:

Candidate should be able to customise existing scripts, or write simple new (ba)sh scripts. This objective includes using standard sh syntax (loops, tests), using command substitution, testing command return values, testing of file status, and conditional mailing to the superuser. This objective also includes making sure the correct interpreter is called on the first (#!) line of scripts. This objective also includes managing location, ownership, execution and suid-rights of scripts.

Key files, terms, and utilities include:

```
while
for
test
chmod
```

Resources of Interest:

Bash Programming Introduction (LDP HOWTO) :

```
http://www.linux.org/docs/ldp/howto/Bash-Prog-Intro-HOWTO.html
```

0.110   (2.10) X [10]

0.110.1   Install & Configure XFree86 [4]

Statement of Objective:

Candidate should be able to configure and install X and an X font server. This objective includes verifying that the video card and monitor are supported by an X server, as well as customising and tuning X for the videocard and monitor. It also includes installing an X font server, installing fonts, and configuring X to use the font server (may require a manual edit of /etc/X11/XF86Config in the "Files" section).

Key files, terms, and utilities include:

```
XF86Setup
xf86config
xvidtune
/etc/X11/XF86Config
.Xresources
```

Resources of Interest:

XWindow-User HOWTO (LDP) :

0.110.2   Setup a display manager [1]

Statement of Objective:

Candidate should be able setup and customise a Display manager. This objective includes turning the display manager on or off and changing the display manager greeting. This objective includes changing default bitplanes for the display manager. It also includes configuring display managers for use by X-stations. This objective covers the display managers XDM (X Display Manger), GDM (Gnome Display Manager) and KDM (KDE Display Manager).

Key files, terms, and utilities include:

Resources of Interest:

XDM and XDMCP HOWTO (LDP) :

```
http://www.linux.org/docs/ldp/howto/XDMCP-HOWTO/index.html
```

The Linux Gazette :

```
http://www.linuxgazette.com/search.html
xdm, gdm, kdm
```

0.110.3   Install & Customise a Window Manager Environment [4]

Statement of Objective:

Candidate should be able to customise a system-wide desktop environment and/or window manager, to demonstrate an understanding of customisation procedures for window manager menus and/or desktop panel menus. this objective includes selecting and configuring the desired x-terminal (xterm, rxvt, aterm etc.), verifying and resolving library dependency issues for X applications, exporting X-display to a client workstation.

Key files, terms, and utilities include:

```
.xinitrc
.Xdefaults
xhost
DISPLAY environment variable
```

Resources of Interest:

TBA

0.112 (1.12) Networking Fundamentals [18]

0.112.1 Fundamentals of TCP/IP [4]

Statement of Objective:

Candidates should demonstrate a proper understanding of network fundamentals. This objective includes the understanding of IP-addresses, network masks and what they mean (i.e. determine a network and broadcast address for a host based on its subnet mask in "dotted quad" or abbreviated notation or determine the network address, broadcast address and netmask when given an IP-address and number of bits). It also covers the understanding of the network classes and classless subnets (CIDR) and the reserved addresses for private network use. It includes the understanding of the function and application of a default route. It also includes the understanding of basic Internet protocols (IP, ICMP, TCP, UDP) and the more common TCP and UDP ports (20, 21, 23, 25, 53, 80, 110, 119, 139, 143, 161).

Key files, terms, and utilities include:

```
/etc/services
ftp
telnet
host
ping
dig
traceroute
whois
```

Resources of Interest:

Linux Networking HOWTO - Joshua Drake :

    http://www.linuxdoc.org/HOWTO/Net-HOWTO/index.html

The Linux Networking Overview HOWTO by Daniel Lopez Ridruejo :

    http://www.linuxdoc.org/HOWTO/Networking-Overview-HOWTO.html

Linux Network Administrators Guide :

    http://www.linuxdoc.org/LDP/nag2/index.html

0.112.2   TCP/IP configuration and troubleshooting [10]

Statement of Objective:

Candidates should be able to view, change and verify configuration settings and operational status for various network interfaces. This objective includes manual and automatic configuration of interfaces and routing tables. This especially means to add, start, stop, restart, delete or reconfigure network interfaces. It also means to change, view or configure the routing table and to correct an improperly set default route manually. Candidates should be able to configure Linux as a DHCP client and a TCP/IP host and to debug problems associated with the network configuration.

Key files, terms, and utilities include:

```
/etc/HOSTNAME or /etc/hostname
/etc/hosts
/etc/networks
/etc/host.conf
/etc/resolv.conf
/etc/nsswitch.conf
ifconfig
route
dhcpcd, dhcpclient, pump
host
hostname (domainname, dnsdomainname)
netstat
ping
traceroute
tcpdump
the network scripts run during system initialisation.
```

Resources of Interest:

Linux Networking HOWTO by Joshua Drake :

```
http://www.linuxdoc.org/HOWTO/Net-HOWTO/index.html
```

Linux Ethernet-Howto by Paul Gortmaker :

```
http://www.linuxdoc.org/HOWTO/Ethernet-HOWTO.html
```

0.112.3   Configure Linux as a PPP client [4]

Statement of Objective:

Candidates should understand the basics of the PPP protocol and be able to configure and use PPP for outbound connections. This objective includes the definition of the chat sequence to connect (given a login example) and the setup

commands to be run automatically when a PPP connection is made. It also includes initialisation and termination of a PPP connection, with a modem, ISDN or ADSL and setting PPP to automatically reconnect if disconnected.

Key files, terms, and utilities include:

```
/etc/ppp/options.*
/etc/ppp/peers/*
/etc/wvdial.conf
/etc/ppp/ip-up
/etc/ppp/ip-down
wvdial
pppd
```

Resources of Interest:

Linux PPP HOWTO Corwin Light-Williams and Joshua Drake :

> `http://www.linuxdoc.org/HOWTO/PPP-HOWTO/index.html`

0.113 (1.13) Networking Services [20]

0.113.1 Configure and manage inetd, xinetd, and related services [5]

Statement of Objective:

Candidates should be able to configure which services are available through inetd, use tcpwrappers to allow or deny services on a host-by-host basis, manually start, stop, and restart internet services, configure basic network services including telnet and ftp. Set a service to run as another user instead of the default in inetd.conf.

Key files, terms, and utilities include:

```
/etc/inetd.conf
/etc/hosts.allow
/etc/hosts.deny
/etc/services
/etc/xinetd.conf
/etc/xinetd.log
```

Resources of Interest:

TBA

0.113.2   Operate and perform basic configuration of sendmail [5]

Statement of Objective:

Candidate should be able to modify simple parameters in sendmail configuration files (including the "Smart Host" parameter, if necessary), create mail aliases, manage the mail queue, start and stop sendmail, configure mail forwarding and perform basic troubleshooting of sendmail. The objective includes checking for and closing open relay on the mailserver. It does not include advanced custom configuration of Sendmail.

Key files, terms, and utilities include:

```
/etc/sendmail.cf
/etc/aliases or /etc/mail/aliases
/etc/mail/*
~/.forward
mailq
sendmail
newaliases
```

Resources of Interest:

TBA

0.113.3   Operate and perform basic configuration of Apache [3]

Statement of Objective:

Candidates should be able to modify simple parameters in Apache configuration files, start, stop, and restart httpd, arrange for automatic restarting of httpd upon boot. Does not include advanced custom configuration of Apache.

Key files, terms, and utilities include:

```
apachectl
httpd
httpd.conf
```

Resources of Interest:

Apache home page :

```
http://www.apache.org
```

0.113.4   Properly manage the NFS, smb, and nmb daemons [4]

Statement of Objective:

Candidate should know how to mount remote filesystems using NFS, configure NFS for exporting local filesystems, start, stop, and restart the NFS server. Install and configure Samba using the included GUI tools or direct edit of the /etc/smb.conf file (Note: this deliberately excludes advanced NT domain issues but includes simple sharing of home directories and printers, as well as correctly setting the nmbd as a WINS client).

Key files, terms, and utilities include:

```
/etc/exports
/etc/fstab
/etc/smb.conf
mount
umount
```

Resources of Interest:

TBA

0.113.5   Setup and configure basic DNS services [3]

Statement of Objective:

Candidate should be able to configure hostname lookups and troubleshoot problems with local caching-only name server. Requires an understanding of the domain registration and DNS translation process. Requires understanding key differences in configuration files for bind 4 and bind 8.

Key files, terms, and utilities include:

```
/etc/hosts
/etc/resolv.conf
/etc/nsswitch.conf
/etc/named.boot (v.4) or /etc/named.conf (v.8)
named
```

Resources of Interest:

TBA

0.113.6   Configure ntp.conf and ntp.drift to be used by xntpd [2]

Statement of Objective:

Candidate should be able to set up the system to synchronise the clock over NTP. Configure it to correct clock drift to match NTP clock.

Key files, terms, and utilities include:

```
/etc/ntp.conf
/etc/ntp.drift
ntpd
ntpdate
```

Resources of Interest:

TBA

0.113.7   Set up secure shell (OpenSSH) [2]

Statement of Objective:

The candidate should be able to obtain and configure OpenSSH. This objective includes basic OpenSSH installation and troubleshooting, as well as configuring sshd to start at system boot..

Key files, terms, and utilities include:

```
/etc/hosts.allow
/etc/hosts.deny
/etc/nologin
/etc/ssh/sshd_config
/etc/ssh_known_hosts
/etc/sshrc
sshd
ssh-keygen
```

Resources of Interest:

TBA

0.114 (1.14) Security [10]

0.114.1 Perform security administration tasks [4]

Statement of Objective:

Candidates should know how to review system configuration to ensure host security in accordance with local security policies. This objective includes how to configure TCP wrappers, find files with SUID/SGID bit set, verify packages, set or change user passwords and password aging information, update binaries as recommended by CERT, BUGTRAQ, and/or distribution's security alerts. Includes basic knowledge of ipchains and iptables.

Key files, terms, and utilities include:

```
/proc/net/ip_fwchains
/proc/net/ip_fwnames
/proc/net/ip_masquerade
find
ipchains
passwd
socket
iptables
```

Resources of Interest:

TBA

0.114.2 Setup host security [4]

Statement of Objective:

Candidate should know how to set up a basic level of host security. Tasks include syslog configuration, shadowed passwords, set up of a mail alias for root's mail and turning of all network services not in use.

Key files, terms, and utilities include:

```
/etc/inetd.conf or /etc/inet.d/*
/etc/nologin
/etc/passwd
/etc/shadow
/ets/syslog.conf
```

Resources of Interest:

TBA

0.114.3   Setup user level security [2]

Statement of Objective:

Candidate should be able to configure user level security. Tasks include limits on user logins, processes, and memory usage.

Key files, terms, and utilities include:

```
quota
usermod
```

Resources of Interest:

TBA

Part I

Resources

Chapter 1

(1.1) Hardware & Architecture [11]

Old number: ()
Weight: []

Configure Fundamental BIOS Settings [3]

Setup SCSI devices [2]

Setup different PC expansion cards [2]

Configure communication devices [2]

Set Up USB devices [2]

1.1   Configure Fundamental BIOS Settings [3]

1.2 Set Up USB devices [2]

1.2.1 Statement of Objective:

Candidates should be able to activate USB support, use and configure different USB devices. This objective includes the correct selection of the USB chipset and the corresponding module. It also includes the knowledge of the basic architecture of the layer model of USB as well as the different modules used in the different layers.

Key files, terms, and utilities include:

```
lspci(8)
usb-uhci.o
usb-ohci.o
/etc/usbmgr/      (A directory)
usbmodules
/etc/hotplug
```

1.2.2 The Universal Serial Bus

- A serial transmission scheme

- Two versions of USB Version 1 & Version 2

- Version

  - 1 released January 1996

  - supports speeds up to 12MBit/s (8.5Mbit/s in practice)

  - supports up to 127 devices connected to the bus

  Version 2:

  - announced 1999

  - supports speeds up to 480Mbit/s

- Devices can be self or bus powered

1.2.3 USB Topology

The system unit contains the host controller and one virtual root hub with at least one (and normally two) USB interfaces. These interfaces can then be connected directly to a USB device or to another HUB.

1.2.4 USB Device Driver Layers

The Device drivers used for the USB sub-system are split into two distinct layers: Hardware Layer - usbcore & usb-uhci / usb-ohci API Layer - Application / Product specific

1.2.5   USB Controllers

There are two categories of USB controller: usb-uhci - For Intel, PIIX4, Via controllers usb-ohci - For Compaq, iMacs, OPTi, SiS, ALi controllers

To determine your controller type, examine /proc/pci for a clue:

```
  [root@Node4] root]# cat /proc/pci
PCI devices found:
   .........
Bus  0, device   7, function  2:
    USB Controller: VIA Technologies, Inc. UHCI USB (rev 17).
      IRQ 10.
      Master Capable.  Latency=32.
      I/O at 0xe400 [0xe41f].
   .........
```

The UHCI controllers use a 16 bit IO address:

```
        I/O at 0xHHHH     eg:  I/O at 0xe400
```

The OHCI controllers use a 32 bit memory address:

```
     memory at 0xHH000000    eg memory at 0xee000000
```

1.2.6   USB Modules

Assuming you have a modular kernel, the following modules will be required:

usbcore - The base usb kernel module

plus one of the controller specific modules: usb-uhci - For Intel, PIIX4, Via controllers usb-ohci - For Compaq, iMacs, OPTi, SiS, ALi controllers

Configuration:

An entry in /etc/modules.conf aliases the specific controller to usb-controller as follows:

alias usb-controller usb-uhci

Starting up the USB sub-system

To have the usb sub-sytem startup automatically at boot time, all you need to do is ensure that the above alias line is present in /etc/modules.conf.

To startup manually, do the following steps: insmod usbcore insmod usb-uhci (or usb-ohci) mount the usbdevfs filesystem (optional but highly recommended)

Example: [root@Node4] root]# insmod usbcore Using /lib/modules/2.4.18-4/kernel/drivers/usb/usbcore.o [root@Node4] root]# insmod usb-uhci Using /lib/modules/2.4.18-4/kernel/drivers/usb/usb-uhci.o [root@Node4] root]# mount -t usbdevfs usbdevfs /proc/bus/usb

Once this is done, you should see the following entries in /proc/bus/usb: [root@Node4] root]# ls /proc/bus/usb 001 devices drivers

1.2.7   USB Interrogation Utilities

LSUSB - A console view of USB devices

Lsusb is a text utility contained in the usbutils package. Use 'rpm -Uvh usbutils.xxx.rpm' to install.

```
[root@node4]# lsusb

Bus 001 Device 001: ID 0000:0000 Virtual Hub
Device Descriptor:
  bLength                18
  bDescriptorType         1
  bcdUSB               1.00
  bDeviceClass            9 Hub
  iProduct                2 USB UHCI Root Hub
 ..........

Bus 001 Device 002: ID 03f0:0601 Hewlett-Packard ScanJet 6300c
Device Descriptor:
  bLength                18
  bDescriptorType         1
  bcdUSB               1.00
  bDeviceClass            0 Interface
  bDeviceSubClass         0
  bDeviceProtocol         0
  bMaxPacketSize0         8
  idVendor           0x03f0 Hewlett-Packard
  idProduct          0x0601 ScanJet 6300c
  bcdDevice            1.00
  iManufacturer           1
  iProduct                2 HP ScanJet 6300C
  iSerial                 3 SG9941706SPE
........

Bus 001 Device 003: ID 1189:6000
Device Descriptor:
  bLength                18
  bDescriptorType         1
  bcdUSB               1.00
  bDeviceClass            0 Interface
  bDeviceSubClass         0
  bDeviceProtocol         0
  bMaxPacketSize0         8
  idVendor           0x1189
  idProduct          0x6000
  bcdDevice            a.03
  iManufacturer           0
  iProduct                1 USB Optical Storage Device
  iSerial                 0
```

USBVIEW - An X view of USB devices

Usbview is a GUI utility contained in the usbview package. Use 'rpm -Uvh usbview.rpm' to install.

Usbview parses /proc/bus/usb/devices for connected USB devices. Any device that has a problem will be printed in red.

1.2.8   Hotplugging Usb Devices

When a device is plugged into a USB port, it will automatically register itself with the USB subsystem. The upper API drivers will not however automatically 'insmod' themselves unless the hotplug package has been installed.

With the hotplug package installed, an entry in /proc/sys/kernel/hotplug will be created which will contain the name of an executable to be called whenever a new device is detected on the bus.

```
$ ls /proc/sys/kernel/hotplug
/sbin/hotplug
```

For example, when a USB scanner is plugged in, hotplug will automatically load the module 'scanner.o'. The xsane application can then be run directly without any user intervention.

/sbin/hotplug is an executable which is called by the kernel (kernel space to user space interface)

/etc/hotplug is a directory containing configuration information for hotplug (which drivers to load when a device is plugged in)

1.2.9   Resources

- The Linux USB Sub System by Brad Hards, Sigma Bravo Pty Ltd

- `www.linux-usb.org`

Chapter 2

Linux Installation & Package Management

Old number: (2.2)
Weight: [24]

Design hard disk layout [2]

Install a boot manager [3]

Make and install programs from source [5]

Manage shared libraries [3]

Use Debian package management [5]

Use Red Hat Package Manager (RPM) [6]

2.1   Design hard disk layout [2]

2.2 Install a boot manager [3]

2.3   Make and install programs from source [5]

2.3.1   Objective

Candidates should be able to build and install an executable program from
source. This objective includes being able to unpack a file of sources. Candi-
dates should be able to make simple customizations to the Makefile, for exam-
ple changing paths or adding extra include directories.

Key files, terms, and utilities include:

```
gunzip
gzip
bzip2
tar
configure
make
```

Resources of interest

- LPI Linux Certification in a Nutshell
  by Jeffrey Dean
  O'Reilly

- LPIC 1 Certification Bible
  Angie Nash and Jason Nash
  Hungry Minds

2.3.2   Source Code Destribution

To distribute software in the form of source code a source tree is archived into
one file using the tar command and then compressed. The resulting file is
called a tarball.
    Source code may also be distributed using the package management tools
of a particular distribution.

Debian apt-get install kernel-source-2.2.27

Redhat rpm -Uhv at-3.1.8-23.src.rpm

Tarball tdb-1.0.6.tar.gz

2.3.3   Steps to Install a package from tarball

- Unpack the taball:

  $ **tar zxvf my-prog.tar.gz** ↩

- Change dircetory into the source tree:

  $ **cd my-prog** ↩

- Configure the `Makefile`:

    ```
    $ ./configure ↩
    ```

- Make:

    ```
    $ make ↩
    ```

- Install:

    ```
    $ su -c 'make install'↩
    ```

2.3.4   Installing the trivial database `tdb`

Download

Locate and download the `tarball`

- googling for it: `http://google.com`

- search on freshmeat: `http://freshmeat.net`

- see if it lives on sourceforge: `http://www.sf.net`

Downlload the tarball to a suitable directory such as `/tmp`.

Unpack

The tarball file is a compressed archived source tree.
    Most commonly the file will be compressed using either `gzip` or `bzip2`
    GNU `tar` can uncompress and unpack the archive:

```
$ tar zxvf tdb-1.0.6.tar.gz ↩
```

or

```
$ tar jxvf tdb-1.0.6.tar.bz2 ↩
```

cd into the tree

The unpacked tarball creates a source tree. The base of which is the name of
the program

```
$ ls ↩
tdb-1.0.6  tdb-1.0.6.tar.gz

$ cd tdb-1.0.6 ↩

$ ls ↩
configure tdb.c tdb.h  README  INSTALL  COPYING
...
```

cd into the tree

```
$ ls -w 70 ←↩
acconfig.h    install-sh     stamp-h.in        tdb.h
aclocal.m4    ltconfig       tdb.3             tdbiterate.c
AUTHORS       ltmain.sh      tdb.c             tdb_open.3
ChangeLog     Makefile.am    tdb_chainlock.3   tdb.spec
config.guess  Makefile.in    tdb_close.3       tdbspeed.c
config.h.in   missing        tdb_delete.3      tdb_store.3
config.sub    mkinstalldirs  tdbdump.c         tdbtest.c
configure     NEWS           tdb_error.3       tdbtool.c
configure.in  README         tdb_exists.3      tdbtorture.c
COPYING       spinlock.c     tdb_fetch.3       tdb_traverse.3
INSTALL       spinlock.h     tdb_firstkey.3    TODO
```

./configure

```
$ file configure ←↩
configure: Bourne shell script text executable
```

```
$ head -5 configure ←↩
#! /bin/sh

# Guess values for system-dependent variables
# Create Makefiles.
# Generated automatically using autoconf version 2.13
```

./configure

```
$ ./configure ←↩
 creating cache ./config.cache
checking for a BSD compat install... /usr/bin/install -c
checking whether build environment is sane... yes
checking whether make sets $MAKE... yes
checking for working aclocal... found
...
creating ./config.status
creating Makefile
creating config.h
```

The `Makefile`

```
SHELL = /bin/sh
CC = gcc
CFLAGS = -g -O2
prefix = /usr/local
includedir = $prefix/include
...
tdbtool: $(tdbtool_OBJECTS) $(tdbtool_DEPENDENCIES)
       @rm -f tdbtool
       $(LINK) $(tdbtool_LDFLAGS) $(tdbtool_OBJECTS)
...
distclean: distclean-am
       -rm -f config.status
```

make

```
$ make ↵
/bin/sh ./libtool --mode=compile gcc -DHAVE_CONFIG_H -I.
-I. -I.     -g -O2  -c tdb.c
mkdir .libs
gcc -DHAVE_CONFIG_H -I. -I. -I. -g -O2 -c  -fPIC -DPIC
tdb.c -o .libs/tdb.lo
gcc -DHAVE_CONFIG_H -I. -I. -I. -g -O2 -c tdb.c -o tdb.o
>/dev/null 2>&1
mv -f .libs/tdb.lo tdb.lo
/bin/sh ./libtool --mode=compile gcc -DHAVE_CONFIG_H -I.
-I. -I.     -g -O2  -c spinlock.c
...
```

make install

```
su -c 'make install'
Password:
make[1]: Entering directory '/tmp/tdb-1.0.6'
/bin/sh ./mkinstalldirs /usr/local/lib
/bin/sh ./libtool  --mode=install /usr/bin/install -c
libtdb.la /usr/local/lib/libtdb.la
...
chmod 644 /usr/local/lib/libtdb.a
PATH="$PATH:/sbin" ldconfig -n /usr/local/lib
```

2.3.5 Play with the trivial database `tdb`

The utility `tdbtool` may be used to have a play with `tdb`.

- Start it and display the help by typing something random:

  ```
  $ tdbtool ↵
  tdb> ?

  tdbtool:
    create    dbname    : create a database
    open      dbname    : open an existing database
    erase               : erase the database
    dump      dumpname  : dump the database as strings
    insert    key  data : insert a record
    store     key  data : store a record (replace)
    show      key       : show a record by key
    delete    key       : delete a record by key
    list                : print the database hash table and freelist
    free                : print the database freelist
    1 | first           : print the first record
    n | next            : print the next record
    q | quit            : terminate
    \n                  : repeat 'next' command
  tdb>
  ```

- Create a database:

  ```
  tdb> create test.tdb
  ```

- Add some data:

```
tdb> insert 1 thing
tdb> insert 2 foo
tdb> insert 3 bar
tdb> insert 55 whizz
```

- Have a look at an entry:

```
tdb> show 3

key 2 bytes
3
data 4 bytes
[000] 62 61 72 00      bar
```

- Experiment.

2.4 Manage shared libraries [3]

2.5   Use Debian package management [5]

2.6 Use Red Hat Package Manager (RPM) [6]

Chapter 1

Old number: ()
Weight: []

Create partitions and filesystems []

Maintain the integrity of filesystems []

Control mounting and unmounting filesystems []

Managing disk quota []

Use file permissions to control access to files []

Manage file ownership []

Create and change hard and symbolic links []

Find system files and place files in the correct location []

1.1 Create partitions and filesystems [3]

Part II

Practical Exercises

Chapter 103

Old number: (1.3)
Weight: [30]

Work on the command line [4]

Process text streams using filters [7]

Perform basic file management [2]

Use streams, pipes, and redirects [3]

Create, monitor, and kill processes [7]

Modify process execution priorities [2]

Search text files using regular expressions [3]

Perform basic file editing using vi [2]

103.1 Work on the command line [4]

103.2   Find system files and place files in the correct location [2]

Part III

Questions

103.103   (1.3) GNU & Unix Commands [30]

103.103.1   Work on the command line [4]