Linux

LPI 101

Exam
Preparation
Version - 2

91- LPI-101 -V2 -Exam Preparation

Table of Contents

Notes about LPI 101 Exam:....................9
Number of Questions per topics....................9
Weight per topics....................9
About this document:....................10
Acknowlegments:....................10
LPI 101 Introduction: 10
Special exam questions attentions....................10
LPI-101 Detailed LPI Topics....................12
1.101 - Hardware & Architecture....................12
General hardware....................12
Address and IRQ conflicts....................12
IRQ Sharing: 12
DMA: 12
Setting Hardware ADDR,IRQ,DMA....................12
Setting and reading the hardware and system time....................12
Getting information on hardware....................14
USB 14
PCI....................14
PCMCIA....................14
PNP....................14
SCSI....................14
SERIAL....................15
CDROMS....................15
I/O ADDRESSES....................15
I/O MEMORY....................15
INTERRUPTS.................... 15
DMA....................15
CPU....................15
DEVICES....................15
KERNEL OPTIONS....................15
FILESYSTEMS....................15
SYSTEM MEMORY....................15
The /proc file system.
....................16
Hardware Parameters....................16
Other hardware information....................16
Kernel and software information....................16
Extra important directories in /proc....................16
Plug and Play
....................17
Description....................17
Manual Process:....................17
IO ADDRESS:....................17
IRQ:....................17
1.101.1 Configure Fundamental BIOS Settings....................18
Description: 18
Key files, terms, and utilities:....................18
Purpose of BIOS:....................18

CMOS Set-up program:....................18
Hard disk Set-up:....................18
BIOS error handling:....................18
Peripherals settings:....................18
IRQ Reserving for older ISA cards....................18
1.101.3 Configure Modem and Sound cards....................19
Description: 19
Modems....................19
Sound Cards....................19
1.101.4 Setup SCSI Devices....................20
Description: 20
Key files, terms, and utilities:....................20
Notes:....................20
Architecture of SCSI:....................20
Number of devices with SCSI, including the SCSI controller itself:....................20
Properties and rules of SCSI....................20
Types of SCSI:....................20
SCSI speed table:....................20
Addressing SCSI devices:....................21
SCSI Onboard BIOS....................21
Booting from SCSI drive....................21
SCSI in Linux....................21
Naming of SCSI devices....................21
1.101.5 Setup different PC expansion cards....................22
Description:....................22
Key files, terms, and utilities:....................22
Tools used:....................22
Important for LPI 22
PCI devices are identified by an unique ID....................22
1.101.6 Configure Communication Devices....................23
Description: 23
Key files, terms, and utilities:....................23
Tools:....................23
Modem AT Commands....................24
1.101.7 Configure USB devices....................25
Description: 25
Key files, terms, and utilities:....................25
USBDevFS Filesystem:....................25
Architecture of USB Modules:....................26
Dynamic loading of USB Modules....................26
hotplug....................27
Files involved:....................27
USB Manager....................27
1.102 - Linux Installation & Package Management....................28
1.102.1 Design a hard disk layout....................28
Description: 28
Key files, terms, and utilities:....................28
File Hierarchy Standard 28
Why multiple partitions in Linux....................28
File tree structure....................28
Swap Partitions....................29
1.102.2 Install a boot manager 30

Description: 30
Key files, terms, and utilities: 30
LILO 30
GRUB 31
How does GRUB work: 32
1.102.3 Make and install programs from source 33
Description: 33
Key files, terms, and utilities: 33
Tools and files used: 33
Difference between script and binary(compiled) programs 33
Verifying the validity of the package's content(checksum) 33
Standard file extentions for packages 33
Unpacking packages 33
Packing files 33
Compiling process 34
Modifying Makefile manually 34
1.102.4 Manage shared libraries 35
Description: 35
Key files, terms, and utilities: 35
Sequence of events 35
Tools and files used 35
1.102.5 Use Debian package management 36
Description: 36
Key files, terms, and utilities: 36
1.102.6 Use Red Hat Package Manager (RPM) 37
Description: 37
Key files, terms, and utilities: 37
Filename format of RPM pacakges 37
Content of RPM packages 37
Where are the RPM databases 37
Syntax 37
Action 37
Installation/Upgrade/Unstallation 37
Installation Options 38
De-Installation Options 38
Package queries 38
Query Package selection 39
Examples: 39
Advanced Examples: 39
PGP and GPG Signature test 40
Extract from RPM Man page 41
1.103 - GNU & Unix Commands 42
1.103.1 - Work on the command line 42
Description: 42
Key files, terms, and utilities: 42
Command format 42
Entering commands 42
Prompt ($PS1) and Incomplete command syntax ($PS2) 42
Entering command sequences 42
Shell (local) and Environment (exported) variables 42
Exported variables 42
Setting and unsetting variables 43

Read-Only Variables......43
$TERM (present terminal type).43
Command substitution......43
Command history and editing......43
Command history navigation......43
Cmd Line Editing commands......45
Applying commands recursively......45
Help of commands using man and info......45
1.103.2 - Process text streams using filters......46
Description:......46
Key files, terms, and utilities:......46
Commands check list......46
Command Examples......47
1.103.3 Perform basic file management......52
Description:......52
Key files, terms, and utilities:......52
Directories, files and Inodes......52
File/directory commands......52
File naming wildcards......52
Finding files with find......52
1.103.4 Use streams, pipes, and redirects......53
Description:......53
Key files, terms, and utilities:......53
Standard I/O......53
Standard I/O......53
pipes......53
|xargs......53
Redirection......53
1.103.5 Create, monitor, and kill processes......54
Description:......54
Key files, terms, and utilities:......54
What is PID, PPID etc......54
Process Viewing Commands......54
Signalling active processes......54
Terminating processes......55
Shell job control and '&'......55
Job Name Format......56
1.103.6 Modify process execution priorities......57
Description:......57
Key files, terms, and utilities:......57
nice......57
renice......57
snice......57
1.103.7 Search text files using regular expressions......58
Description:......58
Key files, terms, and utilities:......58
1.103.8 Perform basic file editing operations using vi......59
Description:......59
Key files, terms, and utilities:......59
Search......59
Cursor move......59

Goto....59
Editing....59
Saving/switching file....60
Load/Reload/Quit....60
Special:....60
Topic 104: Devices, Linux Filesystems, Filesystem Hierarchy Standard.....61
I-nodes....61
Content of boot Block:....61
Content of Super Block....61
Content of Normal inodes....61
....62
EXT2 Filesystem....63
Journaling Filesystems....64
1.104.1 Create partitions and filesystems....65
Description: 65
Key files, terms, and utilities:....65
Partitioning:....65
Creating a filesystem.....67
1.104.2 Maintain the integrity of filesystems....68
Description: 68
Key files, terms, and utilities:....68
1.104.3 Control mounting and unmounting filesystems....70
Description: 70
Key files, terms, and utilities:....70
1.104.4 Managing disk quota....72
Description:....72
Key files, terms, and utilities:....72
Terms of quota editing and reports:....72
Procedure for installing quota for users and groups:(short form)....72
Detailed preparation of quotas....73
Repquota....74
Important Files involved in Quota....74
1.104.5 Use file permissions to control access to files....75
Description: 75
Key files, terms, and utilities:....75
File types....75
Files and directories access rights....75
Changing a file's access rights....75
Directory's access rights....75
SUID and SGID for programs 75
SGID for Directories....76
Sticky Bit for Directories 76
Sticky Bit for programs....76
Command 76
Attributes(chattr & lsattr)....76
Display Attributes of files and directories....76
Attributes list:....78
umask for new files an directories....78
1.104.6 Manage file ownership....79
Description:....79
Key files, terms, and utilities:....79

chown : Change user and group ownership of a file or directory....79
Important Options:....79
IMPORTANT:....79
chgrp : Change group ownership of a file or directrory....80
options:....80
chmod : Change the access rights of a files or directories....80
1.104.7 Create and change hard and symbolic links....81
Description: 81
Key files, terms, and utilities:....81
Creating a Symbolic link:....81
Creating a Hard Link:....81
Options:....81
Important Notes:....81
1.104.8 Find system files and place files in the correct location....83
Description:....83
Key files, terms, and utilities:....83
find : Finding file recursively....83
locate....84
slocate 84
Database Build Options:....84
Slocate Search Options:....85
whereis....85
which....85
Topic 110: The X Window System....86
1.110.1 Install & Configure XFree86....86
Description: 86
Key files, terms, and utilities:....86
X Server....86
Search Path of XF86Config file....86
XF86 Configuration programs....87
Configuration of the XF86Config file 87
Creation of a symbolic link to the configured X-Server....88
Starting the Xserver and Windowmanager:....88
Fine Tuning the monitor settings:....88
X11 Fonts and Fonts server:....88
xset: Temporarily Changing the X-Server's FontPath settings as it runs:....88
Setting-up a Font server:XFS....88
Fonts names Format....89
Installing new fonts:....89
Controlling X-Server settings....90
1.110.2 Setup a display manager....91
Description: 91
Key files, terms, and utilities:....91
Methods of starting an X session....91
Log in from a virtual terminal....91
Via an X-Display-Manager (XDM)....91
Popular display managers: 91
Properties of the Display Managers:....92
xdm configuration:....92
kdm configuration....92
gdm configuration....92
XTerminals using the xdm/kdm Display Managers:....93
1.110.4 Install & Customize a Window Manager Environment....95

Description: 95
Key files, terms, and utilities:...95
Window manager:..95
Window managers names:..95
Configuration files of window managers:..95
Configuration of X Clients (X programs):..95
Positioning and size of window when starting an X Client: 95
Selecting a font for X Clients: 97
~/.Xresources or ~/.Xdefaults file:.. 97
Starting an X session with startx:...97
Starting an X session with xdm/kdm/gdm:..98
X11 in Network..98
Xclient preparation:... ..98
Xserver preparation:..98
Permanent allowance of access:...98
Checking the libraries dependencies for X-Client programs:..................................98

Notes about LPI 101 Exam:

Number of Questions per topics:

Topic	Nr. of Questions
Hardware	7
RPM	14
GNU Cmds	20
Dev. FHS	16
X Server	8
Total ----->	65

Weight per topics:

Topic	Weight
Topic 101: Hardware & Architecture	
1.101.1 Configure Fundamental BIOS Settings	1
1.101.3 Configure Modem and Sound cards	1
1.101.4 Setup SCSI Devices	1
1.101.5 Setup different PC expansion cards	3
1.101.6 Configure Communication Devices	1
1.101.7 Configure USB devices	1
Topic 102: Linux Installation & Package Management	
1.102.1 Design hard disk layout	5
1.102.2 Install a boot manager	1
1.102.3 Make and install programs from source	5
1.102.4 Manage shared libraries	3
1.102.5 Use Debian package management	8
1.102.6 Use Red Hat Package Manager (RPM)	8
Topic: 103 GNU & Unix Commands	
1.103.1 Work on the command line	5
1.103.2 Process text streams using filters	6
1.103.3 Perform basic file management	3
1.103.4 Use streams, pipes, and redirects	5
1.103.5 Create, monitor, and kill processes	5
1.103.6 Modify process execution priorities	3
1.103.7 Search text files using regular expressions	3
1.103.8 Perform basic file editing operations using vi	1
Topic 104: Devices, Linux Filesystems, Filesystem Hierarchy Std.	
1.104.1 Create partitions and filesystems	3
1.104.2 Maintain the integrity of filesystems	3
1.104.3 Control mounting and unmounting filesystems	3
1.104.4 Managing disk quota	3
1.104.5 Use file permissions to control access to files	5
1.104.6 Manage file ownership	1
1.104.7 Create and change hard and symbolic links	1
1.104.8 Find system files and place files in the correct location	5
Topic 110: The X Window System	
1.110.1 Install & Configure XFree86	5
1.110.2 Setup a display manager	3
1.110.4 Install & Customize a Window Manager Environment	5

About this document:

This document been produced to help candidates pass the LPI 101 exam. I have created it essentially as a reference document and not as a tutorial. That's why in general, it doesn't have many explanations for the subjects treated. I usually use it in my courses as exam preparation. To my knowledge it covers the most important aspects of the topics asked in the exam, but it's layout and content organization is not perfect. Helped by this document and with enough practice, most of my students passed the exam. In some topics I have added more information than is needed for the LPI 101 exam. When in doubt, just read again the description of the requirements located at the beginning of each topic.

This is a free document. You may distribute, modify, or improve it for personal or commercial use as you wish.. I take no responsibility of any kind for the accuracy of the information in this document, nor for the success or failure of any participants in passing the exam.
I would appreciate it that if you make modifications to this document, you send me a copy of the new version.

Please let me know of any errors or inaccuracies in the information in this document, that would help me improve it. Feedback of any kind is welcome. If anybody wants to contribute to this document, you're very welcome, please contact me at.
`michel@linuxint.com`

I hope it will help you to prepare for the LPI 101 exam and remember, that practice, practice, and more practice is the key.

Acknowlegments:

Thanks to Alan McKinnon from South Africa for correcting the English of this document.

LPI 101 Introduction:

This is a required exam for LPI certification Level 1. It covers basic system administration skills that are common across all distributions of Linux.
Each objective is assigned a weighting value. The weights range roughly from 1 to 10, and indicate the relative importance of each objective. Objectives with higher weights will be covered in the exam with more questions.

Special exam questions attentions:

- Most questions that require that you fill-in the blanks, don't require any options. eg. `cat` or `ls` or `cp` (without options)

- Use your experience and common sense to deciding what is important and what is not when studying. When in doubt, just read the description of the requirements located at the beginning of each topic again.

- I recommend you create a checklist of topics for yourself and to review it once in a while. This can help avoid spending too much time in one subject at the expense of other important subjects.

- Note the weight of each topic and spend the equivalent amount of time on it.

- When doing the exam, I recommend you first answer the questions that you are sure of and then go back to the other ones afterwards.

- Read the questions thoroughly and make sure you understand them well. Then read ALL the answers carefully before answering. I almost got caught a few times, answering something I was sure couldn't be anything else, but when I read the other answers I saw which one was really the correct answer.

- The exam is difficult and needs concentration and a good memory. It is not recommended to eat a heavy meal before the exam.

- There is no need to rush through the exam and risk overlooking something. There is more than enough time to answer all the questions. When you're finished and there is still time left, review your answers once.

LPI-101 Detailed LPI Topics

1.101 - Hardware & Architecture

- General hardware
  - Processor, BIOS, RAM , Address Bus system, Data Bus system
- Address and IRQ conflicts
  - IRQ Table

- Used by system:	1,2,6,8,14,15
- Mostly free:	5,9,10,11,13
- May be freeable:	3,4,7,12

- IRQ Sharing: PCI Mostly OK, ISA – Not sharable
- DMA:
  - 8 DMA Channels.
  - DMA 4 is always busy.
  - Normal use: DMA 1,2 or 3.
  - Watch for conflicts in DMA!!
- Setting Hardware ADDR,IRQ,DMA

• Peripherals Integrated in Motherboard:	via BIOS
• Old ISA expansion boards:	Jumpers and DIP Switches on boards
• Newer ISA expansions boards:	Jumpers(ADDR)and software (IRQ)
• ISA Plug-And-Play expansion boards:	BIOS or OS PNP feature
• PCI expansion cards:	Via BIOS and OS using PCI bus

  - Memory Base of certain PCI cards for RAM or BIOS direct on the cards

- **Setting and reading the hardware and system time**

```
date                 Show current system date and time
date -s "15:34"      Set the System date and time

hwclock              Show the hardware clock time setting
hwclock --localtime  Hardware clock stores local time
hwclock --utc        Hardware clock stores utc time

hwclock --set --date="9/22/2002 16:45:05"
hwclock --hctosys    Sets the system time to current hardware clock
hwclock --systohc    Sets the hardware clock to current system time
```

- Setting the Hardware and System clock with one command:
  `setclock 09/18/2003 21:13:00`
  (Thu Sep 18 21:13:00 EDT 2003)
- Time Variables in `/etc/sysconfig`
  `HWCLOCK="--localtime"` for `localtime` mode `-u` for `utc` time mode
  When SuSE boots-up it sets the time from the script `/etc/init.d/boot`
- Files that have some relation to time are:
  `/usr/lib/zoneinfo/localtime` --> `/etc/localtime` (binary)
  `/etc/adjtime` Temporary file used to adjust the time regularly

- `ksysctrl`- Is good for displaying the found system devices a-la-Windows.
- `hwinfo` - Shows a lot of info about automatically found hardware. (SuSE)
- `lsdev` - Shows a list of recognized devices and their I/O Addr, IRQ and DMA
- `procinfo`- Shows a list of recognized devices and their I/O Addr and IRQ
- `MAKEDEV` - Command to create devices
- `losetup` - Set up and control loop devices

- **KERNEL MODULES** (general)

  - To list all the Kernel modules already loaded:
    ```
    lsmod
    cat /proc/modules
    ```

  - To get more info about a module
    ```
    modinfo modulename
    ```

  - To load a kernel module use `modprobe` or `insmod`.
    `modprobe` is recommended because it also checks the dependencies of the module.

  - To remove a kernel module:
    `modprobe -r` *`modulename`* (without the `.o`) or
    `rmmod` *`modulename`* (without the `.o`)

  - To list all loadable kernel modules that wouldn't load properly because of missing symbols: (missing symbols = dependency not respected):
    `depmod` (see `man depmod` for more info on module dependencies)

  - Configuration files for hardware modules:
    `/etc/modules.conf` Older configuration used by modprobe to change the way a module is loaded or unloaded. Although this file is an older format it provides a lot of functions.

    `/etc/modprobe.conf` Newer configuration file for modprobe command which is used for the same purpose as `/etc/modules.conf` (older).

    Note: It is unknown to me which one of the two above configuration files would be used if both were present in a system.

  - **Listing the modules options of `/etc/modprobe.conf`:**
    ```
    modprobe -c
    ```

- **Getting information on hardware**

- **USB**

`lsusb` - Lists all connected USB devices
`/sbin/hotplug` - Script; handles hot-pluggable PCI & USB devices.
`rchotplug {start|stop}` - Starts/Stops USB and PCI configurator.
`usbmodules --device /proc/bus/usb/`*`NNN/nnn`*
- Lists kernel modules corresponding to USB devices currently plugged into the computer. *eg.*
`usbmodules --device /proc/bus/usb/001/009`

- **PCI**

`lspci` - List all PCI devices
`cat /proc/pci` - "" "" ""
`setpci` - Configure PCI devices
`pcitweak` - Read/write/list PCI config space
`scanpci` - Scan/probe PCI buses
`/sbin/hotplug` - SuSE cript to handle hot-pluggable PCI and USB devices
`rchotplug {start|stop}` - Starts/Stops USB and PCI configurator

- **PCMCIA**

`cardinfo` - X-Program to list and control PCMCIA cards
`cardctl` - ASCII program to control the PCMCIA cards
`dump_cis` - ASCII program to list PCMCIA cards and their parameters
`cardmgr` - Daemon that loads and unloads PCMCIA kernel modules for inserted cards.
`/etc/init.d/pcmcia` - Script to load PCMCIA `cardmgr` as daemon

- **PNP**

`lspnp` - To list Plug and Play BIOS device nodes and resources.
`/etc/isapnp.conf` - File used by `isapnp`
see also `man setpnp` for info on controlling pnp devices resources.
`isapnp /etc/isapnp.conf` - Sets the PNP devices according to `/etc/isapnp.conf`

- **SCSI**

`sg_map` - Displays mapping between `sg` and other SCSI devices.
`cat /proc/scsi/scsi`- Displays information about all possible SCSI devices:
`hdx`, `srx`, `sgx`, `scdx`
`scsiinfo -l` - List of active SCSI device in system.
eg. `/dev/sda /dev/scd0` etc.
`sg_reset` - exercises SCSI *device/bus/host* reset capability
`scsi_info` - SCSI device description tool
`sg_test_rwbuf` - Tests the SCSI host adapter by issuing write and read operations on a device's buffer and calculating checksums.
`lsscsi` - list all SCSI devices (or hosts) currently on system
`mover` - utility to control SCSI media changers
`sg_scan` - does a SCSI bus scan and prints the results to STDOUT
`sg_senddiag` - performs a SCSI SEND DIAGNOSTIC command
`sg_logs` - reads SCSI LOG SENSE pages

`scsidev`	- populate `/dev/scsi` with device names that are persistent against SCSI configuration changes.
`sg_start`	- starts (spins-up) or stops (spins down) SCSI devices
`sg_map`	- displays mapping between sg and other SCSI devices
`scsiinfo`	- query information from a SCSI device
`sg_readcap`	- calls a READ CAPACITY command on a SCSI device
`sg_rbuf`	- reads data using SCSI READ BUFFER command
`sg_inq`	- outputs data retrieved from the SCSI INQUIRY command
`sginfo`	- outputs mode sense information for a SCSI generic device
`sg_modes`	- reads SCSI MODE SENSE pages
`xmover`	- X11 frontend for SCSI media changers
`scsi_devfs_scan`	- Scan SCSI devices within a devfs tree
`sane-find-scanner`	- find SCSI and USB scanners and their device files
`scsiformat`	- low level format a SCSI disk device

- **SERIAL**
  `cat /proc/tty/drivers` - Display detected serial ports.
- **CDROMS**
  `cat /proc/sys/dev/cdrom/info`
  - The CD-ROM device names and their capabilities. Note: SCSI CDROMs can be `scdx`
- **I/O ADDRESSES**
  `cat /proc/ioports` - I/O Addresses used by which device.
- **I/O MEMORY**
  `cat /proc/iomem` - Memory Addressusage.
- **INTERRUPTS**
  `cat /proc/interrupts` - Interrupt usage
- **DMA**
  `cat /proc/dma` - DMA channels in use.
- **CPU**
  `cat /proc/cpuinfo` - CPU hardware information
- **DEVICES**
  `cat /proc/devices` - Character & Block devices used and their IDs.
  `lsdev` - Displays recognized devices IRQ, DMA and IO.
- **KERNEL OPTIONS**
  `cat /proc/cmdline` - Kernel options given at boot time
- **FILESYSTEMS**
  `cat /proc/filesystems` - Filesystem types recognized by Linux.
  '`nodev`' = it doesn't have any physical device.
- **SYSTEM MEMORY**
  `cat /proc/meminfo` - System Memory management information.

The /proc file system.

- Displays the kernel's internal workings. Mostly ReadOnly.

- Each process get a directory in /proc (named after the PID). Content is:

cmdline	What started the process
cwd	Symlink to dir where user was when he started the command
environ	Environment of process.
exe	Symlink to the running program (full path)
root	root dir for the process. (may be changed using command chroot)
fd	file descriptors (eg. 0,1,2,255. used in prgm 1>&2 etc.)

- Hardware information/parameters: readable with the program cat or less:

Hardware Parameters

interrupts	IRQ used by peripherals
ioports	IO Address used by peripherals
dma	DMA used by peripherals
iomem	Video RAM/ROM, System RAM/ROM, PCI system memory, VESA Frame buffer, reserved areas.

Other hardware information

cpuinfo	Processor type/model, speed, internal cache size, etc.
partitions	List of known local PC partitions with major and minor numbers.
pci	Scan of peripherals on PCI bus and AGP slot.

Kernel and software information

cmdline	Kernel start command and parameters.
filesystems	List of file systems know by the kernel.
meminfo	Info about usage of available memory
modules	List of loaded modules
mounts	List of mounted filesystems. Here are also the mounted filesystems that were mounted with the option -n and hidden from /etc/mtab and df command.
version	Present kernel version.

Extra important directories in /proc

bus	Info about system buses found in systems
ide	Info about IDE controllers and devices
scsi	Info about SCSI controllers and devices
net	Network info like ARP Info, Routing table etc
sys	WRITEABLE system control table.

Plug and Play

- Description A PNP card has an internal list of Addresses, IRQs and DMAs that it can use if requested. Linux is NOT automatic PNP compatible.
  It must be done manually. Two programs are available for this:

  - `pnpdump` Scans the ISA bus for PNP cards and displays the possible settings of each PNP card found.
  - `isapnp` Reads a PNP configuration file and sets the PNP cards accordingly.

- Manual Process: (Using: SuSE package:`isapnp` Debian package: `isapnptools`
  1. Collect possible settings from PNP cards. Scans addresses `0x0273` to `0x03f3`
     `pnpdump > /etc/isapnp.conf`
  2. Edit the file and activate the desired settings of each PNP card (*)
     `vi /etc/isapnp.conf`
  3. Set the PNP cards as per `/etc/isapnp.conf`. Must be done at every boot.
     `isapnp /etc/isapnp.conf`

  (*)Editing the `/etc/isapnp.conf`

  Find:
  IO ADDRESS:
  - First IO base address possible: `Minimum IO base address 0x0`*240*
  - Last IO base address possible: `Maximum IO base address 0x0`*3e0*
  - Address block size: `Number of IO addresses required:` *32*
  - Take a look at the already used IO addresses in system : `cat /proc/ioports`
  - Make a list of possible IO base addresses for this card.
    (First IO base address possible + Address block size) etc
    eg. `240, 260, 280, 2A0, 2c0, ..., ..., 3e0`
  - Choose a free address, write it in the following line and uncomment the line:
    `(IO 0 (BASE 0x`**`340`**`))`

  IRQ:
  - Proceed the same way as above for IRQs and at the end uncomment the line:
    `(INT 0 (IRQ` *xx* `(MODE +E)))` (*xx*=chosen IRQ)
  - Finally activate the card by uncommenting the line: `(ACT Y)`

1.101.1 Configure Fundamental BIOS Settings Weight: 1

Description: 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:**
```
/proc/ioports
/proc/interrupts
/proc/dma
/proc/pci
```

- **Purpose of BIOS:** Middleman program (in ROM) between non-standard hardware architecture (main board) and the operating system.
  Linux deals directly with some hardware (eg. IDE Controller) for speed and better control.

- **CMOS Set-up program:**
  Triggered at boot-time by possible key combinations: DEL (Entf), F2, <Ctrl-Alt-ESC> etc.

- **Hard disk Set-up:**
  Although the hard disk controller is accessed directly by Linux, some HD settings in CMOS are still important.
  - HD cylinders have physically less sectors at the inside of the disk than at the outside.
  - LBA (Large Block Address) logically reduces the number of cylinders and increases the number of heads.
  - LBA is important if number of physical cylinders is more than 1024 even if Linux doesn't use the BIOS to access the HD.
    Reasons: - `fdisk` reads the BIOS for HD Parameters
    - `lilo` and GRUB are loaded from the BIOS.

- **BIOS error handling:**
  Normally: `Halt on all errors` (booting does not continue if any type of error occurs)
  Linux server without keyboard: `Halt on all errors but keyboard`.

- **Peripherals settings:**
  - Turn off any unused device. eg. COM ports, Mouse, IDE channels if SCSI used, etc

- **IRQ Reserving for older ISA cards**
  These parameters are read from the ISA PNP and PCI cards and deletes them from their possible parameters list. When asked for possible set-ups,(eg. by `pnpdump`) the PNP cards will not have these reserved addresses.

1.101.3 Configure Modem and Sound cards

Weight: 1

Description: Ensure devices meet compatibility requirements (particularly that the modem is NOT a win-modem), verify that both the modem and sound card are using unique and correct IRQ's, I/O, and DMA addresses, if the sound card is PnP install and run sndconfig and isapnp, configure modem for outbound dial-up, configure modem for outbound PPP | SLIP | CSLIP connection, set serial port for 115.2 Kbps

Modems

- Check the hardware compatibility list from the distribution used.
- Good source of hardware info is the Hardware-HOWTO
- Normal modems are controlled by AT commands (Hayes compatible)
- Watch for WinModems. They are not real hardware modems. This section is not for them. A Winmodems is hardware that doesn't have this AT command intelligence and relies on drivers to simulate it. More about it at `www.linmodems.org`

Sound Cards

- LPI concentrates on OSS sound technique. (Open Sound System)
- Each sound board type needs its own kernel module.
- Program for sound card installation: `sndconfig` (RedHat and others)
  - It scans possible sound cards IO ports and is menu driven.

    Standard I/O port for soundcard is: **********

    It handles the PNP and older ISA sound cards as well.

1.101.4 Setup SCSI Devices

Weight: 1

Description: 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:**

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

- **Notes:**
  - SCSI=**S**mall **C**omputer **S**ystem **I**nterface
  - Purpose: Learning to set-up the SCSI devices with respect to BIOS, SCSI-ID, booting
  - Use of SCSI: Still in server industry, offers reliability, endurance, Hot-Plug features.
  - Tools: SCSI-ID, `/proc/scsi, scsi_info`
    (`scsi_info` is from Packages: SuSE: `pcmcia`, Debian: `pcmcia-cs`)

- **Architecture of SCSI:**
  - Number of devices with SCSI, including the SCSI controller itself:

Standard:	8
Wide:	16

  - Properties and rules of SCSI
    - Cable joining the devices is 50 wires wide
    - No 'T' branching in the cable
    - Each end of the cable must be terminated by 330 Ohms to GND and 220 Ohms to +5V
    - Minimum 10cm of cable between SCSI devices
    - Maximum length of 50 strand cable: 3 Meters (>4 devices Max:1.5 Meters)
    - End of the cable must have a terminated device attached to it.

  - Types of SCSI:
    - Standard(SCSI-1): 8 Devices 10 MHz Maximum
    - SCSI-2, FAST-SCSI-2, Wide-SCSI-2(68 strand cable, 16 bit bus):
      Faster, command set is better
    - SCSI-3 even faster but still in development (no meaning for LPI)

  - SCSI speed table:

Bus width	Cable Width	Standard	Fast	Ultra	Devices
8-Bit	50 Strands	5 MB/sec	10 MB/sec	20 MB/sec	7+Ctrlr
Wide-16-Bit	68 Strands	10 MB/sec	20 MB/sec	40 MB/sec	15+Ctrlr

Possible names alike Ultra-Wide- or Fast-Wide, etc are possible

- Addressing SCSI devices:
  - SCSI-ID = 0 to 7 or 0 to 15
  - The SCSI Controller with the highest priority = highest ID: 7 or 15
  - If booting from SCSI then boot HD must be ID 0
  - Each SCSI-ID can contain LUNs (Logical Unit Number)
  - Each CSCI cable (Bus) also receives a number (0,1,2 etc)
  - Each SCSI device can then be identified as follows:
    BusNumber, SCSI-ID, LUN
    Normally `0,x,0` eg. `/dev/sda` is on `0,0,0`

- SCSI Onboard BIOS
  - Separate and unknown from system BIOS
  - Used to boot SCSI drives and change controller parameters
  - Cheap Controllers don't usually have On-Board BIOS. More expensive ones do.
  - Newer Controllers even allows software to assign SCSI-IDs to devices.
  - Role of the Controller:
    - Assignment of SCSI-IDs to devices
    - Selecting the data transfer rate of devices
    - Selection of boot drive

- Booting from SCSI drive.
  - Controller must have an onboard BIOS
  - In SCSI onboard BIOS: Set the boot drive
  - In System BIOS: Set boot drive sequence to 'SCSI'

- SCSI in Linux
  - `/proc/scsi` directory contains all SCSI devices as a sub-directory
  - Each sub-directory contains files named by SCSI-BUS number (0,1,2)
  - Theses files contain the list of devices attached to this bus.
  - The file `/proc/scsi/scsi` contains the list of all found SCSI devices.

- Naming of SCSI devices
  - Hard disks are named sda, sdb ... in the sequence they are found
  - Removable ZIP and USB Chip readers are also in the hard disk class
  - SCSI CD-ROMS have 2 names at the same time: `srx` & `scdx` (*x*=0,1,2,3,..)
  - Each device is also identified by ***SCSI-BUS***, ***SCSI-ID***, ***LUN***
  - Program `scsi_info` shows info on individual device:
    eg. `scsi_info /dev/scd0`

1.101.5 Setup different PC expansion cards Weight: 3

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

- Tools used:
  - Info files: `/proc/dma, /proc/interrupts, /proc/ioports, /proc/pci`
  - Programs: `pnpdump(8), isapnp(8), lspci(8)`

- Important for LPI is: - Hardware parameters (IO Port, IRQ,DMA)
  - `/proc` directory
  - ISA Plug and Play in Linux
  - Setting and reading the time

- PCI devices are identified by an unique ID just like MAC address in network cards. Linux saves these PCI IDs in the file:

- `/usr/share/pci.ids`	(SuSE)
- `/usr/share/hwdata/pci.ids`	(RedHat & Debian)
- `/usr/share/mics/pci.ids`	(Old Debiandistributions)

  `update-pciids` command updates the list from Internet into:
  `/usr/share/mics/pci.ids.new` or equivalent as per distribution.

- Linux support PCI(Bus ID=`00`) devices fully without needing manual settings.

- AGP Is a separate PCI bus(Bus ID=01) reserved for Graphic Cards, having only one slot. Made for undisturbed data transfer between the graphic chips and the CPU.

- PCI Bus system is addressed the same way as SCSI:
  `BusNr``:`*`SlotNr`*`:`*`FunctionNr`*`(Device Nr.)`

- `lspci` is used to list the PCI devices in system.
  `lspci` finds the manufacturers info from the file `/usr/share/pci.ids.`
  `lspci -n` display vendor codes as numbers instead of lookingthem up in `pci.ids`.

- Kernels after `2.1.82` have more info about devices on PCI-Bus in `/proc/pci`.

- Serial ports known as `COM1`, `COM2` etc in DOS, are known in Linux as: `ttyS0,ttyS1` etc.Parallel Printer ports known as `lpt1, lpt2 n DOS`, are known in Linux: `lp0, lp1 etc`

1.101.6 Configure Communication Devices Weight: 1

Description: 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 correct modem settings for outbound PPP connection(s).

- **Key files, terms, and utilities:**

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

- Tools: `/proc/dma`, `/proc/ioports`, `/proc/interrupts`, `setserial(8)` (`setserial` is from package `setserial` for SuSE,RedHat & Debian) `minicom` is one of the modem terminal programs for linux.

- `setserial /dev/ttySx` Shows the settings of the serial port.
  or `/dev/cuax` `x=0,1,2,3...`eg. `ttyS0=COM1, ttyS1=COM2`

- `setserial /dev/ttySx` *`parameter`*
  Sets the serial port to the parameters.

  Parameters are:

`port` *`PortNr`*	IO Port number
`irq` *`IRQ`*	IRQ number

`uart` *`UART_Type`*	UART(Universal Asynchronous Receiver Transmitter)
	Possible values are: none, 8250, 16450, 16550, 16550, 16550A, 16650V2, 16654, 16750, 16850, 16950, 16954.
	none=Turn device OFF

Most older application know only up to 38400 Baud. To allow for faster speeds even though the application asks for 38400 Baud, extra parameters to `setserial` set flags in hardware that translates requests from applications of 38.4Kb to higher speeds in UART.

Parameter	Speed requested by Aplication	Real UART speed
`spd_normal`	38.4Kb	38.4Kb
`spd_hi`	38.4Kb	57.6Kb
`spd_vhi` **(Important for LPI)**	38.4Kb	115Kb
`spd_shi`	38.4Kb	230Kb
`spd_warp`	38.4Kb	460Kb

- Modem AT Commands

Hayes compatible commands that controls most modems.

AT	Sets the baud rate between Modem and PC
ATD Nr.	Dial the Number (Nr.)
ATH0	HangUp
ATH1	Answer the phone (Opposite of HangUp)
ATX0	Dial blind, CONNECT when connection OK
ATX1	Dial blind, CONNECT *speed* when connection OK
ATX2	Wait for DIALTONE and CONNECT *speed* when connection OK
ATX3	Dial blind, CONNECT *speed* when connection OK or BUZY
ATX4	Wait for DIALTONE and CONNECT *speed* when connection OK
ATX5	Dial blind, CONNECT *speed* when connection OK, BUSY, VOICE
ATX6	Wait for DIALTONE and CONNECT *speed* when connection OK, BUSY, VOICE
ATZ	Reset the modem.
AT&F	Reset the internal modem configuration to factory settings.

1.101.7 Configure USB devices

Weight: 1

Description: 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:**
  ```
  lspci(8)
  usb-uhci.o
  usb-ohci.o
  /etc/usbmgr/
  usbmodules
  /etc/hotplug
  ```
- Main USB module is usbcore (although often already integrated in kernel)
- There are 2 types of USB controllers:
  OHCI Open Host Controller Interface (Compaq)
  UHCI Universal Host Controller Interface (Intel)
- All USB devices are compatible with both OHCI and UHCI.
- Main boards manufacturers using:

OHCI	**UHCI**	**EHCI**(USB 2.0)
Compaq	Intel	Intel
Ali	VIA	VIA
NEC		NEC
Opti Chipset		Philips

- `lspci` or `less /proc/pci` To recognize the USB controller type:
  IO address format: `0xHHHH=UHCI`, `0xHH000000=OHCI`
- The possible modules are: `ohci.o` , `uhci.o` or `ehci-hcd.o`
- Autoloading at boot-time: in `/etc/modules.conf`---->entry: `alias usb uhci`
  To also autoload (post-install) other submodules:
  eg.(in `/etc/modules.conf`):
  ```
  alias usb uhci
  post-install uhci modprobe printer
  post-install printer modprobe joydev
  post-install joydev modprobe hid
  ```

USBDevFS Filesystem:

This dynamic filesystem (like `/proc`) is normally mounted on `/proc/bus/usb`.
Its `/etc/fstab` entry looks like:

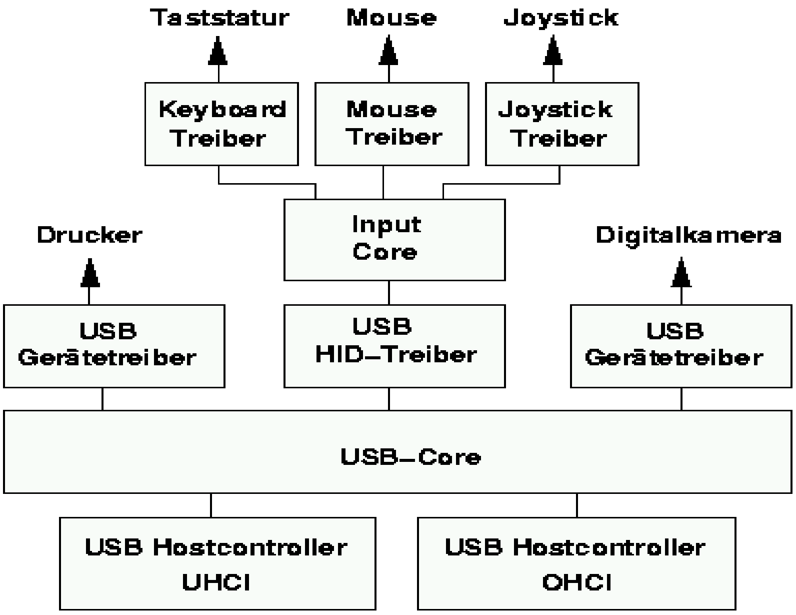
```
none  /proc/bus/usb  usbdevfs  defaults  0 0
```

After the mounting, the content(2 files) of `/proc/bus/usb` looks like this:

```
-r--r--r--  1 root root 0 2003-10-18 00:02 devices
-r--r--r--  1 root root 0 2003-10-18 00:02 drivers
```

After loading the driver(`usb-ohci` or `usb-uhci`) then content of this directory grows to include 1 numbered(001,002 etc) directory for each USB device.
The files in these numbered directories are in binary format.

- **Architecture of USB Modules:**



HID = Human Interface Device.(`hid.o`) and Input Core (`input.o`) are only for a USB keyboard(`usbkbd.o`), Mouse(`usbmouse.o`) or Joystick otherwise not needed. Other USB modules:

printers	`printer.o`
storage	`usb-storage.o`

USB devices can be listed with the command: `lsusb`

For printers,when `printer.o` is loaded it creates devices `/dev/usb/lp0 ..lp1.`

List of current USB modules:

Commmand: `find /lib/modules/ -name "usb*" -exec basename {} \;`

```
usb-ohci.o          usbserial.o
usb-uhci.o          usb-storage.o
usbcore.o           usb-midi.o
                    usbkbd.o
                    usblcd.o
                    usbmouse.o
                    usbnet.o
                    usbvideo.o
                    usbvnet5.o
                    usbvnet5_2958.o
                    usbvnetr.o
```

Dynamic loading of USB Modules
For dynamic loading of the proper USB module when a USB device is inserted, 2 dynamic systems are available:

`hotplug`	Oversees the Hotplugged devices: USB, PCMCIA, FireWire(ieee1394)
`usbmgr`	USB manager that oversees only the USB devices.

- **hotplug**
  - At boot time the hotplug daemon is started via the script `/etc/init.d/hotplug`.
  - When a new device is inserted, the kernel senses it , it then passes an agent name as parameter to the daemon listed in the file: `/proc/sys/kernel/hotplug` (normally `/sbin/hotplug`).
  - The kernel then fills in the Environment Variable DEVICES with the info about the device, and ACTION indicating if the device was plugged or unplugged.
  - The hotplug daemon starts the proper agent script.
  - The agent script reads the content of the DEVICES and ACTION variables as well as possibly other variables provided by the kernel. It uses also the program `usbmodules` to find-out all about the device inserted.

The specific 'agents' scripts are.

The USB system uses	`/etc/hotplug/usb.agent`
The PCMCIA uses	`/etc/hotplug/pci.agent` (via a bridge)
Firewire(ieee1394) uses	`/etc/hotplug/ieee1394.agent`
The Network system uses	`/etc/hotplug/net.agent`

- **Files involved:**

`/lib/modules/*/modules.*map`	depmod output
`/proc/sys/kernel/hotplug`	specifies hotplug program path
`/sbin/hotplug`	hotplug program (default path name)
`/etc/hotplug/*`	hotplug files
`/etc/hotplug/`*`NAME`*`.agent`	hotplug subsystem-specific agents
`/etc/hotplug/`*`NAME*`*	subsystem-specific files, for agents
`/etc/hotplug/`*`NAME/DRIVER`*	driver setup scripts, invoked by agents
`/etc/hotplug/usb/`*`DRIVER`*`.usermap`	depmod data for user-mode drivers
`/etc/init.d/hotplug`	hotplug system service script used also to load and configure already plugged hot-plug devices at boot time.

USB Manager (`usbmgr`)

Is a daemon that will load the proper module according to 2 parameters given by the kernel: USB-Vendor-ID and USB-Device-ID

It uses the following configuration files:

`/etc/usbmgr/usbmgr.conf`	List of Vendor-ID/Device-ID and module names
`/etc/usbmgr/preload.conf`	List of modules to load when usbmgr starts.
`/etc/usbmgr/host`	List of module names od the USB controller: either `usb-ohci` or `usb-uhci`.

- The `usbmgr` needs the following conditions:
  - The kernel must be USB capable (`usbcore`)
  - The USBDEVFS mus be supported
  - The needed modules must be available.

• 1.102 - Linux Installation & Package Management.

The content of this section includes:

	Weight
1.102.1 Design hard disk layout	5
1.102.2 Install a boot manager	1
1.102.3 Make and install programs from source	5
1.102.4 Manage shared libraries	3
1.102.5 Use Debian package management	8
1.102.6 Use Red Hat Package Manager (RPM)	8

• 1.102.1 Design a hard disk layout Weight: 5

Description: 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:**

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

- **File Hierarchy Standard** (FHS)
- **Why multiple partitions in Linux**:

  - Multiple hard disks can be used
  - Easier backups
  - Quotas are active per partition
  - Mount partitions Read-only for protection
  - Possible limit of Boot Manager (<1024 cylinders)

- **File tree structure**:

  Absolutely Needed: `/bin, /dev/, /etc, /lib, /sbin`
  Note: `/root` is recommended in case root must do a rescue

`/usr`	Can be ReadOnly in it's own partition.
`/home`	Recommended as a separate partition. Reason: quotas, non-interference and fast system recovery
`/tmp`	Recommended as a separate partition. Reason: quotas, non-interference
`/var`	Highly recommended: When full, doesn't interfere with system.

`/var/lib`	- Modifiable settings
`/var/lock`	- Lock files for programs and daemons
`/var/log`	- Log files of system, daemons and programs
`/var/run`	- PIDs of daemons and programs (if needed)
`/var/spool`	- Queues for printing, mail etc.
`/var/tmp`	- Space for temporary files. Writable by anybody

- **Swap Partitions**
  - Created and managed by: `fdisk, mkswap, swapon, /etc/fstab`

Creating a swap file (64MB):

```
    dd if=/dev/zero of=/path/of/swapfile bs=1024 count=64000
    mkswap /path/of/swapfile 64000
    swapon [-p 42] /path/of/swapfile
```

Entry in `/etc/fstab`

```
         /path/of/swapfile   none   swap
```

`[cat /proc/swaps]` to see all the swap files

• 1.102.2 Install a boot manager Weight: 1

Description: 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:**

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

- - MBR(<512 bytes), Partition table and Boot sector
  - Stage 1 Program in MBR or Bootsector loads the bootmanager from the hard disk. (performs direct disk access)
  - Stage 2 Presents menu and waits
  - Stage 3 Starts the selected operating system

  - **LILO:**

`/etc/lilo.conf`	Main and only config file.
`/boot/boot.b`	Boot Menu file

`/boot/boot-menu.b`
`/boot/message,`
`/boot/boot-bmp.b`
	Other possible menu elements

`/boot/map`	Physical Address and size of kernel files.
`/sbin/lilo`	Program that: - Reads config file and writes the first stage bootloader to MBR - Uses BIOS functions and creates `/boot/map`

Note: After any change to `/etc/lilo.conf` or any location or size of any file in `/boot` directory, `lilo` MUST be rerun.

`lilo -u` - Rewrites the previous boot manager in MBR (eg. windows MBR)

  - LILO display codes:

Nothing	The partition booted is not Boot-activated or no bootmanager.
L*ErrorNr*	Second part of LILO cannot be loaded and error number is the reason.
LI	Second part of lilo is loaded but is probably invalid. Reason: `lilo` was probably not run after changes in `/boot` or config file.
LIL	Second part of LILO is loaded but the content of `/boot/map` is wrong. Reason: Media error or wrong media geometry.
LIL?	Second part of LILO is loaded but it is garbage. Reason: file `/boot/boot.b` has moved or changed.
LIL-	Second part of LILO is loaded but the `/boot/map` is garbage. Reason: file `/boot/map` has moved or changed.
LILO	All OK. LILO has loaded properly.

- **`/etc/lilo.conf`**

`append="reboot=warm"`	NO RAM check when rebooting.
`boot=/dev/hda`	Where the LILO part1 should be written `hda`=MBR, `hda1`=Boot sector of `hda1` etc.
`lba32`	HD has physically more than 1024 cylinders. and LILO should be using the LBA mode. Only valid if BIOS supports LBA32
`message=/boot/message`	Message loaded in MBR with part1of `lilo.conf`
`prompt`	LiLO will wait for user selection.
`timeout=300`	Boot default system when timeout occurs. 300 = 30 seconds

<u>Individual Systems sections</u>

`image = /boot/bzlinuz`	Location of the kernel to load.
`root = /dev/hda2`	Partition to use as root directory `'/'`
`initrd = /boot/initrd`	Location of ramdrive file: temporary file system.
`label = linux`	Name of menu item.
`other = /dev/hda3`	Location of a non-Linux OS. Goes to that partition and loads the boot sector it finds there. Normally used with Windows.j

- **GRUB** (GRand Unified Bootloader)

- Hard disks are numbered as `(hd0)` - the first HD found in system, `(hd0,0)` is the the first partition of the first hard drive; normally `/dev/hda1`.

- NO static binary menu (`/boot/boot.b`). Instead it's `/boot/grub/menu.lst`

- `/boot/grub/grub.conf` can also be used as menu/config file.

- Entries in GRUB menu/config file:

`default=0`	First menu item starts if no selection done before timeout.
`timeout=10`	Timeout of 10 seconds will occur if no selection is made by user.
`splashimage=(hd0,0)/boot/grub/splash.xpm.gz`	The menu image will be taken from `/dev/hda1` in this path.

<u>Individual Systems sections</u>

`title linux`	Menu item text
`root (hd0,0)`	First partition of first found Hard Drive is used for the dir `'/'`
`kernel /boot/bzlinuz ro root=/dev/hda1`	The kernel is `/boot/bzlinuz` and some parameters like `ro` and `root=/dev/hda1` is given to the kernel when started.
`initrd`	Ramdrive for booting (if used by kernel)

`map (hd0,1) (hd0,0)`
`map (hd0,0) (hd0,1)`
	Used to swap the assignment of physical partitions.
	Useful to let Windows 98 boot from a partition which is not the first one (picky fellow!!) and make it think that it is.

`rootnoverify (hd0,1)`
Set GRUB's root device without mounting.

`chainloader +1`
Jump to the Boot Sector of the root partition and hope a boot loader is there waiting....

`makeactive` Make the partition active.

To boot from a CD/Floppy/HD and use the root directory of another Linux as it own root dir (`/`) then use the kernel parameter:
`root=/dev/hda4`

How does GRUB work:
`stage1` file is written in the MBR.
It contains the physical address of the *`fssys`*`_stage1_5`.(*`fssys`*=filesystem)
GRUB Booting sequence:

- MBR (`stage1`) is loaded
- `stage1` loads *`fssys`*`_stage1_5` (filesystem converter)
- *`fssys`*`_stage1_5` loads `stage2` file
- `stage2` loads the `menu.lst`
- After boot menu item selection is done,
  `stage2` loads the kernel(`vmlinuz`) and possibly the `initrd`

See document `20_LILO-GRUB.sxw` for more info on Grub.

1.102.3 Make and install programs from source

Weight: 5

Description: 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 customizations to the Makefile, for example changing paths or adding extra include directories.

- **Key files, terms, and utilities:**

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

- **Tools and files used:**
  - `gzip, gunzip, bzip2, tar, configure, Makefile, make`

- **Difference between script and binary(compiled) programs.**
  CPU only understands binary instructions. programs must be translated from programming language to binary. A compiled program is done once and program file is binary. With script each line is translated to binary then executed as it runs.

- **Verifying the validity of the package's content**(checksum)
  - Get the MD5 checksum file from the location where you downloaded the file.
  - Put the tar file and the checksum file in the current directory

```
md5sum --check checksumfilename
```

- **Standard file extentions for packages:**

Tarred files:	`*.tar`
Compressed tarred files:	`*.tar.gz` or `*.tar.bz2` or `*.tgz`
Zipped files:	`*.gz`
Bzipped files:	`*.bz2`

- **Unpacking packages:**
  Compressed tar files: (a new directory will be created in destination directory.)

```
cd DestinationDirectory
tar fvxz tarfile.tar.gz or    zcat tarfile.tar.gz|tar xvf -
tar fvxj tarfile.tar.bz2 or bzcat tarfile.tar.bz2|tar xvf -
```

Un-compressed tar files:

```
tar fcx tarfile.tar
```

Compressed files:

```
gunzip   File.gz ----> File   (original File.gz is deleted!!)
bunzip2  File.bz2     ----> File     ""   File.bz2 ""
bzip2 -d File.bz2     ----> File     ""      ""    ""
```

- **Packing files:**
  Uncompressed tar files

```
tar fvc newfile.tar /dir/to/pack
```

Compressed tar files

```
tar fvcz newfile.tar.gz  /dir/to/pack
tar fvcj newfile.tar.bz2 /dir/to/pack
```

Compressed files

```
gzip filename  ----> filename.gz      (original is deleted!!!)
bzip2 filename ----> filename.bz2     ( ""   ""   ""   )
```

- **Compiling process:** (Absolutely needed commands are in **`Bold`**)

  `cd` *`SourceBaseDirrectory`*
  `./configure` This script studies system environment and creates `Makefile`

  `make` or `make all` Reads `Makefile` and start the compiling of the source files.

  `make install` Installs the compiled files and possibly others in the system.
  Normally only possible to run as root.

  `make clean` Deletes all the already compiled binary files.

  Note: `make` works with timestamps which tells whether the binary file should be recompiled if the source has been changed since last compile.

- Modifying Makefile manually
  Changes to the Makefile would normally be done to adjust the installation paths.
  These changeable parameters are normally at the beginning of the Makefile.
  They are in the normal bash variable assignment format: `var=value`

• 1.102.4 Manage shared libraries

Weight: 3

Description: 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:**

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

Libraries are SHARED between running programs within RAM. So only one copy of a shared library is needed to be loaded in RAM for all programs using it.

- **Sequence of events**:

- `bash` tells the kernel to start a program
- The kernel starts the Dynamic Library Linker `ld.so`
- `ld.so` searches for all libraries needed for the program in the following order:
    - Looks in the (':' separated) paths listed in shell Env. variable `LD_LIBRARY_PATH`
    - Looks in the paths listed in the library cache `/etc/ld.so.cache`
    - Looks in `/lib` and `/usr/lib`
- `ld.so` loads itself
- `ld.so` loads the program in memory and pass-on control to the program.

- **Tools and files used:**

`LD_LIBRARY_PATH`	Bash Environment Variable containing List of paths of Libraries to search.
`ldd` *`/path/to/program`*	Lists all the libraries a program needs.

`/lib`
`/usr/lib`
`/usr/local/lib`
	Standard dir. where are most libraries installed.

`ldconfig`	Program that keeps track of all libraries in system. When a library is installed in a directory other than above standard library locations then we need to: - Enter new library path in `/etc/ld.so.conf` - Run `ldconfig`. This updates the libraries paths cache: `/etc/ld.so.cache` .
`/etc/ld.so.conf`	Configuration file of `ldconfig`
`/etc/ld.so.cache`	Libraries paths cache.

1.102.5 Use Debian package management Weight: 8

Description: Candidates should be able to perform package management skills using the Debian package manager. 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:**

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

```
TODO
```

• 1.102.6 Use Red Hat Package Manager (RPM) Weight: 8

Description: 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:**

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

- **Filename format of RPM pacakges:**
  `PackageName-VersionNumber.Architechture``.rpm`

- **Content of RPM packages:**
  - Information about the package
  - List of files to install
  - List of Dependencies
  - 4 Scripts: - Before Installation - Before De-Installation
    - After Installation - After De-Installation
  - Files to install

- **Where are the RPM databases:**
  - Directory of RPM Database of installed packages(in binary format):
    `/var/lib/rpm/*`
  - Directory of RPM tools needed to manage RPM packages:
    `/usr/lib/rpm/*`
  - To rebuild the RPM Database: `rpm --rebuilddb`

- **Syntax:**
  `rpm` *`Action`* `[`*`Options`*`]` *`Packagename`*`[.rpm]`

  - **Action:**

Short Format	Long Format	Description
Installation/Upgrade/Unstallation		
`-i`	`--install`	Install the package. Works only when no older package was already installed.
`-U`	`--upgrade`	Upgrade the package. Works like Install. but will also erase an older version of it.
`-F`	`--freshen`	Upgrade the package. Works only when an older version of it is already installed.
`-e`	`--erase`	Uninstall the package.

Installation Options

`--nodeps`	Installs and does not check dependencies.
`--noscripts`	No Pre/Post-Install scripts will be run.
`--test`	Do not install, just simulate installation.
`--excludedocs`	Install but without the documentation.
`--replacepkgs`	Install all even if some pacakges are already installed.
`--replacefiles`	Overwrite already installed files if they exist.
`--oldpackage`	Allow downgrading a package version.
`--force`	Install all no matter what. It can be seen as the same as: `--replacepkgs --replacefiles --oldpackage`

De-Installation Options

`--nodeps`	De-installs and does not check dependencies.
`--noscripts`	No Pre/Post-de-install scripts will be run.
`--test`	Do not de-install, just simulate de-installation.
`--allmatches`	De-install all packages names that matches pattern. In this case no errors would be produced if the pattern matched 2 or more packages.

Package queries

- If a query is made on installed packages the package name needs to be naked without the version or revision number or `.rpm`
- If a query is made for an RPM file, then the actual file name including the `.rpm` must be given as the package name.

`-q[options]`	`--query`	Queries info on the package. (no `.rpm`)
`i`	`--info`	Information header of package.
`l`	`--list`	List of all files
`c`	`--configfiles`	List of Configuration files.
`d`	`--docfiles`	List of Documentation files.
	`--provides`	Programs/Libs provided by the package.
`R`	`--requires`	List of files on which this package depends.
	`--changelog`	Display log of package changes.
	`--scripts`	Displays all 4 InstalluUninstall scripts**.**
	`--dump`	List of all files and their attributes.
	`--filesbypkg`	Same as `--list` + package name per line
	`--last`	Date of last installations of the package.
	`--state`	`--LIST` + Files Installation status: `normal`, `not installed` or `replaced`

`--qf %{`*`QUERYTAG`*`}` or

`--queryformat %{`*`QUERYTAG`*`}`

Extracts specific items from info header.

eg. `rpm -q --qf %{DESCRIPTION} apache`

Displays only description part of the info.

eg. `rpm -qa --qf "%{NAME}\n"|sort|less`

Lists only names of all installed packages.

`--querytags`

Lists of *`QUERYTAG's`* usable in `--queryformat`.

Query Package selection:(combined with `-q` option)

a	`--all`	Query all installed packages
f	`--file` *`file`*	Query installed package owning file (incl. path)
p	`--package`	Query Specific uninstalled packages (`.rpm`)

- `--whatrequires`
  Query all installed packages that depend on this one.
  Note: `rpm -q --whatrequires --qf "%{NAME}\n" apache|less`
  Displays all names of packages that depends on `apache` package.

- `--whatprovides` *`program_or_libname`*(incl. path)
  Query all installed packages that provides this program or library.
  Note:
  ```
  rpm -q --whatprovides --qf \
  "%{NAME}\-%{VERSION}\-%{RELEASE}\n" /bin/sed
  ```
  Displays names of packages that provides `/bin/sed`.
  Same output as `rpm -qf /bin/sed`

- `g` *`group`* or `--group` *`group`* All installed packages belonging to group
  Note: `rpm -qa --qf "%{GROUP}\n" | sort -u | less`
  Lists group names of which some packages are installed.

Examples:

`rpm -qil` *`PackageName`* Information and install file list of package.
`rpm -qa | sort | less` Display all installed packages(all `.rpm` files)
`rpm -qai | grep -2 "^Release" | less`
Same as above but more complete info.
`rpm -hiv` *`PackageName`*`.rpm` Install with progress bar (hash #)
`rpm -hiv --replacefiles` *`PackageName`*`.rpm`
Install on top of existing package with progress bar (hash #)
`rpm -hUv` *`PackageName`*`.rpm` Upgrade with progress bar (hash #)
`rpm -hUv --force` *`PackageName`*`.rpm`
Upgrades and overwrite existing package even if conflict or lack of dependencies exists.
`rpm -qf` *`filename`*(incl PATH) Tells which packet this file belongs to
`rpm -qdf` *`filename`*(incl PATH) Tells which help documents came with this file

Queries for not installed `rpm` packages

`rpm -qpi` *`PackageName`*`.rpm` header information of this package.
`rpm -qpl` *`PackageName`*`.rpm` List of files where this package installs

Advanced Examples:

- To display a list of all already installed packages and their summary description:

```
rpm -qa --qf "%{NAME} : %{SUMMARY}\n" | sort | less
```

- To search for an already installed PackageName by pattern:

```
rpm -qai | awk -F: ' /^Name|^Version/ {print $1,$2} /^Version/ \
   {print ""}' | cut -b-30 | grep -1 $1
```

- To search through non installed RPM files for a filename:

```
rpm -qp --filesbypkg *.rpm | grep filename
```

- Verifying integrity of packages

`rpm -V  PackageName`	Verify integrity of the installed packages
`rpm -Va`	Verify integrity of all installed packages

The result of both of these above commands will be shown as follows:

- One line per file is displayed.
- Each line contains a status field(8 chars.) File type (1 char) Filename.

eg. `S.5....T c /etc/samba/smbpasswd`

Meaning of Status field:

S	Size of file has changed	U	File Owner has changed
M	Access rights has changed	G	Group of file has changed
5	MD5 Checksum doesn't match	T	Timestamp has changed
L	ReadLink system call failed	.	OK
D	Major/Minor numbers of device has changed		

Meaning of filetype:	'c'	Configuration file.	' '	Normal File
	'?'	Couldn't check (maybe because read access failed)		

- **PGP and GPG Signature test**

Syntax:

```
rpm --checksig Packagefilename.rpm
```

result should be:

```
   Packagefilename.rpm md5 gpg OK
                       ----------
```

- **Extract from RPM Man page:**

QUERYING AND VERIFYING PACKAGES:

```
rpm  {-q|--query} [select-options] [query-options]
rpm  {-K|--checksig} [--nogpg] [--nopgp] [--nomd5] PACKAGE_FILE ...
rpm  {-V|--verify} [select-options] [--nodeps] [--nofiles] \
     [--nomd5] [--noscripts]
```

INSTALLING, UPGRADING, AND REMOVING PACKAGES:

```
rpm  {-i|--install} [install-options] PACKAGE_FILE ...
rpm  {-U|--upgrade} [install-options] PACKAGE_FILE ...
rpm  {-F|--freshen} [install-options] PACKAGE_FILE ...
rpm  {-e|--erase}   [--allmatches]    [--nodeps] [--noscripts] \
     [--notriggers] [--repackage]     [--test]   PACKAGE_NAME ...
```

MISCELLANEOUS:

```
rpm  {--initdb|--rebuilddb}
rpm  {--querytags|--showrc}
rpm  {--addsign|--resign}    PACKAGE_FILE ...
rpm  {--setperms|--setugids} PACKAGE_NAME ...
```

SELECT-OPTIONS

```
[PACKAGE_NAME] [-a,--all] [-f,--file FILE] [-g,--group GROUP]
[-p,--package PACK-AGE_FILE] [--querybynumber NUMBER]
[--triggeredby PACKAGE_NAME] [--whatprovides  CAPABILITY]
[--whatrequires CAPABILITY]
```

QUERY-OPTIONS

```
[--changelog]  [-c,--configfiles] [-d,--docfiles] [--dump]
[--filesbypkg] [-i,--info]        [--last]        [-l,--list]
[--provides]   [--qf,--queryformat QUERYFMT]      [-R,--requires]
[--scripts]    [-s,--state]       [--triggers,--triggerscripts]
```

INSTALL-OPTIONS

```
[--allfiles]  [--badreloc] [--excludepath OLDPATH] [--excludedocs]
[--force]     [-h,--hash]     [--ignoresize]       [--ignorearch]
[--ignoreos]  [--includedocs] [--justdb]           [--nodeps]
[--noorder]   [--noscripts]   [--notriggers]       [--oldpackage]
[--percent]   [--prefix  NEWPATH]  [--relocate OLDPATH=NEWPATH]
[--repackage] [--replacefiles][--replacepkgs]      [--test]
```

Note: Options for Building packages are left out here. See man page for further info.

- **Other source of info are at** `http://www.rpm.org`
  The program `kpackage`, `KpackViewer` and `kpm`

• 1.103 - GNU & Unix Commands

Content:	Weight
1.103.1 - Work on the command line	5
1.103.2 - Process text streams using filters	6
1.103.3 - Perform basic file management	3
1.103.4 - Use streams, pipes, and redirects	5
1.103.5 - Create, monitor, and kill processes	5
1.103.6 - Modify process execution priorities	3
1.103.7 - Search text files using regular expressions	3
1.103.8 - Perform basic file editing operations using vi	1

1.103.1 - Work on the command line Weight 5

Description: Candidates should be able to Interact with shells and commands using the command line. This includes typing valid commands and command sequences, defining, referencing and exporting environment variables, using command history and editing facilities, invoking commands in the path and outside the path, using command substitution, applying commands recursively through a directory tree and using man to find out about commands.

- **Key files, terms, and utilities:**

```
.
bash            man
echo            pwd
env             set
exec            unset
export
~/.bash_history
~/.profile
```

- Command format (command, options and parameters)
    - short (-) and long (--) form options.
    - short form options combinations
- Entering commands
    - Which are in the `PATH`
    - Which are not in the `PATH`
    - `'.'` as part of `PATH` and `./command`
    - Where am I?: pwd
- Prompt ($PS1) and Incomplete command syntax ($PS2)
- Entering command sequences
    - With ';' '&'
- Shell (local) and Environment (exported) variables:
    - Exported variables (Environment variables)
        - `env` Lists all environment variables
        - `printenv` Same as env
        - `export` Lists all exported (environment) variables
        - `declare -x variable[=value]` Sets the env. variable
        - `export variable[=value]` " "" ""

- Setting and unsetting variables:
    - `set` Lists all local and environment variables incl. functions
    - Setting: `set variable=value`
    - UnSetting: `unset variable`
- Read-Only Variables (variable cannot be changed or unset)
    - `readonly` Lists all read-only variables
    - `readonly variable` Sets the variable to read-only
- `$TERM` (present terminal type)

screen	Console in 'screen' mode
dumb	From cron
linux	From tty1-tty6
xterm	Xserver terminal

- Terminal info Database
    - -Old: `/etc/termcap` Still in SuSE
    - -New: `/etc/terminfo/*` Debian uses it

- Command substitution: old `` `command` `` and new `$(command)`

eg.
```
echo "My present directory is `pwd`"
ls -la /lib/modules/$(uname -r)/*
```

- Command history and editing

Command history navigation:

set +o history	Turns history recording OFF
set -o history	Turns history recording ON
$HISTFILE	Variable containing the history file name. Normally ~/.bash_history
$HISTFILESIZE	Variable containing the maximum number of commands the history file can contain. Default=500
$HISTSIZE	Variable containing the maximum number of commands in history. Default=500

history	Displays the whole history
history 10	Displays the last 10 lines of history
fc -l -10	Displays the last 10 lines of history
fc -l *Pattern*	Search the history for Pattern & display the result
<Ctrl>-r	Reverse search in history
history -c	Clears the whole history

!!	Most recent command
!*n*	Command *n* in the history
!-*n*	Backwards command *n* in history
! *string*	Last recent command starting with *string*
!? *string*	Last recent command containing with *string*
^*string1*^*string2*	Quick substitution *string1* to *string2*
<Ctrl>-p	Previous Line in history (also up-arrow)
<Ctrl>-n	Next Line in history (also down arrow)
<Alt>-<	Go to beginning of History
<Alt>->	Go to end of History

Cmd Line Editing commands

(E-macs editing cmds -`readline` library)

`<Ctrl>-l`	Clear screen
`<Ctrl>-b`	Back one character (also left arrow)
`<Ctrl>-f`	Forward one character (also right arrow)
`<Ctrl>-a`	Go to beginning of line (also `Pos1` key)
`<Ctrl>-e`	Go to end of line (also `Ende` key)
`<Ctrl>-k`	Delete text from cursor to end of line
`<Ctrl>-d`	Delete a character on the right (or under cursor)
`<Alt>-d`	Delete from cursor to end of current word
`<Ctrl>-y`	Paste text previously cut (deleted)

- Applying commands recursively (`-r,-R,--recursive`)

Command	Short format	Long format
ls	-R	--recursive
chown	-R	--recursive
chmod	-R	--recursive
chgrp	-R	--recursive
grep	-r	--recursive
cp	-r and -R	--recursive
rm	-r and -R	--recursive

- Help of commands using `man` and `info`.

`man [`*n*`]` *`command`* Syntax to call a man page for a command
n Represents the man page type (`1-9`)

1	Executable programs or shell commands
2	System calls (functions provided by the kernel)
3	Library calls (functions within program libraries)
4	Special files (usually found in `/dev`)
5	File formats and conventions eg. `/etc/passwd`
6	Games
7	Miscellaneous (including macro packages and conventions), e.g. `man`(7), `groff`(7)
8	System administration commands (usually only for root)
9	Kernel routines [Non standard]

1.103.2 - Process text streams using filters Weight 6

Description: Candidates should be able to apply filters to text streams. Tasks include sending text files and output streams through text utility filters to modify the output, and using standard UNIX commands found in the GNU textutils package.

- **Key files, terms, and utilities:**

```
cat
cut
expand
fmt
head
join
nl
od
paste
pr
sed
sort
split
tac
tail
tr
unexpand
uniq
wc
```

Commands check list

❒	- `cut`	Extracts columns from file:
❒	- `expand`	Expands TABs to SPACES in text files
❒	- `fmt`	Formatting of text files
❒	- `head`	Display first x lines of text file
❒	- `join`	Joins lines of a data file on common fields
❒	- `nl`	Number the lines of a text file
❒	- `od`	Display file content in Octal, Hex or Decimal.
❒	- `paste`	Pastes corresponding lines of 2 text files
❒	- `pr`	Convert text files ready for printing
❒	- `split`	Splits files into multiple files
❒	- `cat`	Concatenate files / Display files content
❒	- `tac`	Displays content of text file bottom to top
❒	- `tail`	Display last x lines of text file
❒	- `tr`	Translate or delete characters of file
❒	- `wc`	Counts number of chars, words, lines of files
❒	- `xargs`	Extends the argument list of a command
❒	- `sed`	Stream file editor
❒	- `sort`	Sorting content of files
❒	- `grep`	Filtering/extracting text from files
❒	- `more`	Display content of files - Page Forward
❒	- `less`	Display content of files - Lines Forward/Backward

Command Examples:

`cut -d`*`x`*` -f`*`y`* - Extracts columns from file: field(*y*) separator(*x*)
eg. `cut -d: -f1,6 /etc/passwd` (Extract field 1 and 6)

`expand` - Expands(converts) TABs to SPACEs in text files.
eg. `expand /etc/init.d/at > ~/atnew`

`unexpand` - Opposite of `expand`: Converts SPACEs to TABs in text files.
eg. `unexpand -a /etc/services > ~/serv ; vi ~/serv`

`fmt` - Formatt text files: Before printing. (for continuous text only)
Note. Each line must have at least one space within it.
eg. `fmt -w50 /usr/share/doc/packages/apache/ABOUT_APACHE`

`head [-|+][n]`*`x`* - Display first x lines of text file (default 10)
eg. `head -40 /etc/services`
Displays the first 40 lines of the file

`join` - Joins lines of a data file on common fields
eg. `join -t: -11 -21 /etc/passwd /etc/shadow`

`nl` - Number the lines of a text file.
`nl -ba` *`filename`* Numbers empty lines as well
Default options: `-v1 -i1 -l1 -sTAB -w6 -nrn -hn -bt -fn`
eg.1 `nl -s" - " /etc/services`
Number the lines, adding " - " after line number
eg.2 `nl -bp"^#" file1` Numbers only the lines starting with '#'

`od -bih -t`*`x`* - Display file content in Octal (`-b`), Decimal(`-i`), Hexadecimal.(`-h`)
eg. `od -h /bin/ping`
x = Format type

-a	same as -t a,	select named characters
-b	same as -t oC,	select octal bytes
-c	same as -t c,	select ASCII characters or '\' escapes
-d	same as -t u2,	select unsigned decimal shorts
-f	same as -t fF,	select floats
-h	same as -t x2,	select hexadecimal shorts
-i	same as -t d2,	select decimal shorts
-l	same as -t d4,	select decimal longs
-o	same as -t o2,	select octal shorts
-x	same as -t x2,	select hexadecimal shorts

Note: `-x` is not the same as `-tx`

`paste` - Pastes corresponding lines of 2 text files
eg.1 `paste /etc/passwd /etc/shadow`
eg.2
```
cut -d: -f1 /etc/passwd > file1
cut -d: -f3 /etc/passwd > file2
paste -d: file1 file2 > file3
```

`pr` - Convert text files for printing
eg. `pr /etc/services | less`

`split -lx -by[b|k|m]`

- Splits files into multiple files containing (*x*) lines, (y) bytes, kilobytes or megabytes. Syntax: `split [options] filename prefix`

eg1. `split -l100 /etc/services serv`

creates `servaa servbb` etc. To get the original back

`cat serv?? > servicesnew`

eg2. `split -b1440k /bin/rpm rpms` (for backups to diskettes)

creates `rpmsaa rpmsab` etc. To get the original back

`cat rpms?? > rpmnew`

`cat`

- Displays content of text file top to bottom and exits

eg.1 `cat -n /etc/hosts`

Show all lines of file with line numbers(`-n`)

eg.2 `cat -b /etc/hosts` Numbers only the non-empty lines

`tac`

- Displays content of text file bottom to top and exits (reverse of `cat`)

eg. `tac /etc/passwd`

List starts with the lasts users created in system

`tail [-|+][n]x`

- Display last *x* lines of text file (default 10)

eg1. `tail -30 /etc/services` (Display last 30 lines)

eg2. `tail +100 /etc/services`

Bypass first 100 lines and display the rest till end of file

eg3. `tail -fs5 --retry /var/log/httpd/error_log`

Read the last 10 lines of the file every 5 sec. and keep retrying even if the file is not available

`tr -d`

- Translate or delete characters of file

eg1. `tr "a-z" "A-Z" < /etc/motd` (translates a-z to A-Z)

eg2. `tr -d "#" < /etc/services | less` (deletes all #)

`wc -l|-c|-w`

- Counts number of `--lines(-l)`, `--words(-w)` or `--chars(-c)` of text file. Without options it counts all lines, words and chars.

eg. `wc /etc/motd`

`xargs`

- Reads text from pipe and provides it as parameter(s) to specified command - up to max 64kb per command launch.

eg. `find /etc -name *.conf | xargs cat > /root/confs`

Finds all `.conf` files in `/etc` and accumulates their contents all in one file called `/root/confs`.

`sed`

- Stream file editor (see `45_Editing_Text-sed.sxw`)

eg.1 `sed 's/#/;-/g' /etc/services`

eg.2 `sed '12,$s/Versions/Revisions/g'`

Start global(g) substitution at line 12 till end of file ($)

`sort -ky[n] -tx`

- Sort text file by field(*y*) with field separator(*x*)default sep.:<space>

eg.1 `ls -la | sort -k5n` (sorted by file size: field 5)

eg.2 `ls -la | sort -n +0.32`

Same result as above: excludes characters 0 to 32 in sorting

`grep [-virns]` - Extract all lines of text where pattern is [not] found

eg1. `grep -ins "^f.p.*SSL$" /etc/services`
Display all lines of file where pattern(ignoring case (`-i`) is found with its line numbers(`-n`) and no error messages (`-s`)

eg2. `ps -ax | grep httpd | grep -v grep`
Display all instances of processes where `httpd` is found excluding(`-v`) the `grep httpd` command itself

`more` - Forwards only display of text file content

eg. `more -30 /etc/services`
Scrolls display next 30 lines when pressing space bar, press enter to scroll to the next line

`less` - Scrollable display of text file/pipe content. Press v to edit the file

eg1. `less -X +G /etc/services`
Go to the end of the file (`+G`) and leave the display as is(`-X`) when leaving `less.`

eg2. `less -phttps /etc/services`
Load file and go to first occurence of search pattern `https`

`uniq` - Filters consecutive line repetitions of a file.

eg.

```
   rpm -qa --qf "%{LICENSE}\n" | sort | uniq | less
or rpm -qa --qf "%{LICENSE}\n" | sort -u | less
```

Display all the licences types used by installed packages.

`awk -F`*`x`* - Programmable text formatter fields delimited (*x*)

eg. `awk -F: '{ print $1,"\t- ", $3 }' /etc/passwd`

1.103.3 Perform basic file management Weight: 3

Description: Candidates should be able to use the basic UNIX commands to copy, move, and remove files and directories. Tasks include advanced file management operations such as copying multiple files recursively, removing directories recursively, and moving files that meet a wildcard pattern. This includes using simple and advanced wildcard specifications to refer to files, as well as using find to locate and act on files based on type, size, or time.

- **Key files, terms, and utilities:**

```
cp              ls
find            rm
mkdir           rmdir
mv              touch
file globbing
```

Directories, files and Inodes

File/directory commands

`cd` `/newdir` or `cd newdir` Absolute/relative change directory
eg. `cd ~foo` Changes to the home directory of user `foo`

`ls` `[dir|file]` List content of directory or file information.
eg. `ls -lai /etc` Long format lists of files inc. inode numbers

`cp` *`source destination`* Copy files or directories
eg. `cp source1 source2 ... .` Copy all files in the current directory
`cp /dev/null newemptyfile` Create a new empty file

`mv` *`source destination`* Move or rename file or directories
eg. `mv -f` (default) `file1 file2` Allows overwriting of `file2` if exists
`mv -i file1 file2` Request confirmation before
overwriting

`mkdir` Create directories.
Options: `-p|--parents` Creates full paths, existing or not
`-m 755` To set the access rights mode

`rmdir` Deletes Directories. `-p|--parents` for non-empty directories
Non-Empty directories must be empty.

`touch` Change file modification time of a file.
Can also be used to create an empty file:
`touch file1 or >file1 or echo >file1`

File naming wildcards `* ? [...] [...-...] [!...]`

Finding files with `find`:
See document `42_Finding_Files-find.pdf` for info on `find`.

1.103.4 Use streams, pipes, and redirects Weight: 5

Description: Candidates should be able to redirect streams and connect them in order to efficiently process textual data. Tasks include redirecting standard input, standard output, and standard error, piping the output of one command to the input of another command, using the output of one command as arguments to another command and sending output to both stdout and a file.

- **Key files, terms, and utilities:**

```
tee
xargs
<    <<
>    >>
|    ` `
```

Standard I/O (`STDIN(0)` and `STDOUT(1)` and `STDERR(2)`)

pipes '**|**'. Note: '|' redirects only the STDOUT and NOT the STDERR
`prg1 2>&1 | prg2` Redirects STDOUT and STDERR

|xargs Uses output of a program as list of arguments for another one.
eg. `prgm1 | xargs prgm2` is same as `prgm2 $(prgm1)`
The diference is that `xargs` will deliver the arguments in chunks of 64kBytes to `prgm2` and run `prgm2` multiple times until all arguments (output of `prgm1`) are used up.

`xargs` example:

```
find /etc -name "issue*" 2>/dev/null | xargs grep -c "SuSE"
```

- Redirection `>   >>   <<   <   1>   2>   &>   2>&1`

`>` first overwrite/creates new file, processes the command and then writes the STDOUT of command into the file.
Therefore the command: `sed 's/#/;/g' file1 > file1`
Simply overwrites `file1` with an empty one !!!

Note. `>newfile` Is the same as `touch newfile`

`<` Redirects STDIN from a file instead of the keyboard.
`prgm < file1` Reads its input from `file1`.

'**here-document**'

```
prgm << EOF Text goes here ....
EOF
```

`prgm` gets its input from text between first OEF and last OEF

Combining outputs:

```
prgm 2>&1 1>file or prgm &>file
```

Both commands combine STDOUT and STDERR to be sent to `file`

`program |` <u>`tee`</u> *`filename`* (Redirecting to STDOUT and file)

eg. `prg1 | tee file1 | prg2|tee file2 | prg3 >file3`
gives the same result as the following detached commands:

```
prg1 > file1
prg2 < file1 > file2
prg3 < file2 > file3
```

1.103.5 Create, monitor, and kill processes Weight: 5

Description: Candidates should be able to manage processes. This includes knowing how to run jobs in the foreground and background, bring a job from the background to the foreground and vice versa, start a process that will run without being connected to a terminal and signal a program to continue running after logout. Tasks also include monitoring active processes, selecting and sorting processes for display, sending signals to processes, killing processes and identifying and killing X applications that did not terminate after the X session closed.

- **Key files, terms, and utilities:**

```
&             kill
bg            nohup
fg            ps
jobs          top
```

-What is `PID`, `PPID` etc

Process Viewing Commands

```
ps          Show process table
Examples:
ps waux     All processes with user in a wide format
ps caux     All processes with user with true command name
            Practical for killall command.
ps -fe      All processes (-e) with full listing(-f)
ps -la      All processes (-a) excluding session leaders
ps -eo "%p &P %n %y %x %c"

            Formatted output as:
            PID  PPID  NICE  TTY  TIME     COMMAND
            %p   %P    %n    %y   %x         %c

pstree      Show process tree
top         Interactively show most processor 'time hungry'
            processes
```

- Signalling active processes

```
kill SIGxxx     Send signals to a process
          eg.   kill SIGHUP 1329 same as
                kill HUP 1329  or  kill -1 1329

nohup prgm      Runs prgm with HUP signal immunity.
                STDOUT and STDERR is sent to
                ./nohup.out or $HOME/nohup.out

kill -l         List of signals possible
```

- Terminating processes

`kill [-9]`	Brutal killing of a process
`killall`	Killing many processes in one command
`xkill`	X Program to kill a process owning a window
`pkill`	See `man pkill`
`skill [`*`signal`*`]`	`[`*`option`*`]` *`parameter`* Allows sending signals to multiple processes at the same time.

<u>`skill` Options(optional) & parameters</u>:

`-t` *`terminal`*

Affects all processes running off a specific terminal. (`tty`*`x`* or `pts/`*`x`*)

`-u` *`username1`* `[`*`username2`*`....]`

Affects all process belonging to one or more users.

`-p PID1 [PID2 ...]`

Affects all process owning the PID(s).

`-c` *`CommandName`*

Affects process having the *`CommandName`*

<u>examples:</u>

`skill -KILL -v pts/*`

Kill and list(`-v`) all processes on new-style PTY devices

`skill -STOP viro lm davem`

Stop 3 users: `viro lm` and `davem`

- Shell job control and '&'

Job Control Commands

`bg` `[%`*`n`*`]`	Resume current or stopped job *n* in the background
`fg` `[%`*`n`*`]`	Move current or background job *n* into foreground
`jobs` `[`*`option`*`]`	Display status of all jobs `-n` Status since last job change `-r` List of running jobs only `-s` List stopped jobs only `-l` display status of all jobs and their process ID's `-p` display process ID's of all jobs
`jobs` `-x` *`command`*	Replace job *n* in *command* with corresponding process group id, then execute *command*
`kill` `[-`*`signal`*`]` *`%n`*	Send specified signal to job *n* (default **15**)
`stop` *`%n`*	Stop job *n*
`stty [-]tostop`	Allow/prevent background jobs from generating output
`suspend`	Suspend execution of current shell
`wait`	Wait for all background jobs to complete
`wait` *`%n`*	Wait for background job *n* to complete
`Ctl-z`	Stop current job
`disown` `[`*`option`*`] [%`*`n`*`]`	Disown the last activated(+) background job or job *%n*. Disowned job will not die when shell dies. `-a` Disown all the background jobs `-r` Disown only the running jobs `-h` Disown active job (+)from shell <u>only</u> when shell is closed:

Job Name Format

`%%, %+`	current job
`%`*n*	job *n*
`%-`	previous job
`%`*`string`*	job whose name begins with *string*
`%?`*`string`*	job that matches part or all of *string*

1.103.6 Modify process execution priorities Weight: 3

Description: Candidates should should be able to manage process execution priorities. Tasks include running a program with higher or lower priority, determining the priority of a process and changing the priority of a running process.

- **Key files, terms, and utilities:**
  `nice`
  `ps`
  `renice`
  `top`

Notes:

-Possible priorities: `19`(min) to `-20`(max)
-Users can only change to a lower priority than the current one
-Priority when normally starting a program: `0`

`nice`	Start a job with pre-defined priority	
	`nice --8 prgm`	Start `prgm` with priority `-8`
	`nice -11 prgm`	Start `prgm` with priority `11`
	`nice -n-12 prgm`	Start `prgm` with priority `12`
`renice`	Change priority of a running process	
	`renice -6 1247`	Change priority of prgm w/ PID-1247 to `-6`
	`renice  8 1247`	Change priority of prgm w/ PID-1247 to `8`
`snice`	Change priority of a multiple running processes by category.	
	syntax: `snice` [*`newpriority`*] [*`options`*] *`category`*	
	eg.	
	`snice +7 seti crack`	Slow down `seti` and `crack`
	`snice -17 root bash`	Give priority to root's shell.

1.103.7 Search text files using regular expressions Weight: 3

Description: Candidates should be able to manipulate files and text data using regular expressions. This objective includes creating simple regular expressions containing several notational elements. It also includes using regular expression tools to perform searches through a filesystem or file content.

- **Key files, terms, and utilities:**

```
grep
regexp
sed
```

```
grep  Search for patterns in text.      See 43_Finding_Text-grep.pdf
      Syntax: grep "regexp" filename
      See also: grep -F, grep -E

sed   Edit text using patterns.           See 45_Editing_Text-sed.pdf
      Ranges are declared as start,end
      eg. sed '1,$s/^\#/##/'
            Substitute from line 1 till end($) of document
eg.   sed -f sedscr file1  Uses sed commands in sedsrc
      sed -e 'cmd1' -e 'cmd2' file1  Multiple commands
      sed 's/pattern/replacement/g'  Global substitution
      sed '/pattern/d'Delete matching lines or grep -v "pattern"
      sed 's/^\(.*\) \(.*\)/\1_\2/'  Using Variables(\1 \2)
            Last example inserts '_' between first 2 words in all lines.

Regular expressions(regex) .       See. 94_Regular_Expressions.pdf
List:  .  *  ^  $  \<  \>  \b  \B  [..]  \  (..)  {..}  +  ?  |
```

1.103.8 Perform basic file editing operations using vi Weight: 1

Description: Candidates should be able to edit text files using vi. This objective includes vi navigation, basic vi modes, inserting, editing, deleting, copying, and finding text.

- **Key files, terms, and utilities:**

```
vi
/, ?
h,j,k,l
G, H, L
i, c, d, dd, p, o, a
ZZ, :w!, :q!, :e!
:!
```

Search:	*`/pattern,?pattern`*			
		Search forward, backward *`pattern`*		
Repeat Search:	`/,?`	forward, backward		
Goto:	`n`	forward next found		
	`N`	Backward next found		
Cursor move:	`l`	forward	`h`	backward
	`k`	up	`j`	down
	`w,W`	Forward one word		
	`b,B`	Backward one word		
	`e`	End of current word		
	`0`	Beginning of line		
	`$`	End of line		
Goto:	`0`	Beginning of line	`$`	End of line
	`H`	Top of screen	`L`	Bottom of screen
	`:1` First line	`G` Last line	`23` Line 23	
Editing:	`<esc>`	Command mode		
	`i`	Insert mode		
	`:sp`	Split screen in 2		
	`<Ctrl>w w`	Change to other spilt window		

<u>Delete and put in clipboard:</u>

`x` Deletes the char on the right or under the cursor(DEL)

`d` Delete current char or line till (incl.) next cursor move.

eg. `dl` Delete next char. on the right same as `x`

`dk` Delete current line & one line above

`d0` Delete from cursor till begin of line.

`d$` Delete from cursor till end of line.

`c` Same as `d` but starts inserting after

eg. `ch` Delete 1 char backward then insert mode.

`cj` Delete current line then insert mode.

`C` Delete till end of line and then insert mode.

`dd` Delete lines

eg. `dd`-Delete current line

`3dd`-Delete 3 lines (incl. current line)

Clipboard Copy and Paste:

`yy,`*`n`*`yy`	Copy current line, *n* lines to clipboard
`p,P`	Paste Clipboard before, after cursor position

Start editing(insert mode):

`i,a,A`	Insert text before, after cursor, at end of line
`o,O`	Open new line for text below, above cursor

Undoing actions:

`u,<Ctrl>r`	Undo last action in command mode.
`<Alt>u`	Undo last action in insert command mode.

Saving/switching file:

`:wq   :x   ZZ`	Save file and exit
`:w    :w!`	Save file, Save file (overwriting files)
`:w` *`file`*	Save file under....(no overwrite)
`:w!` *`file`*	Save file under....(can overwrite)
`:x` *`file`*	Save file under....(no overwrite)
`:n    :N`	Show the next, previous buffer
`:f`	Show name of current file

Load/Reload/Quit

`:e` *`file`*	Loads a new *`file`* if current file is saved.
`:e!` *`file`*	Loads a new *`file`* even if current file is not saved.
`:r` *`file`*	Insert the content of *`file`* at cursor position
`:!` *`cmd`*	Run shell command(*`cmd`*) and come back to current file editing.
`:e!`	Load last saved version of this file
`:q!`	Quit without saving

Special:

Substitutions:

`:1,$s/`*`pattern`*`/`*`replacement`*`/g`   (same as `sed`)

Running a shell command (filter) on a range of text

`:`*`range`*`!` *`shellcommand`*

Runs the range of text through shell command (filter) and replace the original with the results.

eg. `:1,$! grep -v '^\#'`   Deletes all comments lines.

Topic 104: Devices, Linux Filesystems, Filesystem Hierarchy Standard

I-nodes

- A fixed number of inodes are created when a filesystem is created depending on the size of the hard disk.
- Directories are files (type 'd') containing filenames and their respective inodes.

Storage elements names in filesystems:

- msdos : *Clusters*
- Linux filesystems: *Blocks*

- Normal size of Blocks: 1024 Bytes
- Possible sizes: 512,1024 & 2048 Bytes
- Each Linux filesystem partition contains:
    - 1 Boot block
    - 1 SuperBlock
    - inodes area
    - Data area

Content of boot Block:

- Boot sector normally used to store a Boot Manager

Content of Super Block(partial) :

- Depending on filesystem, includes:
    - Number of blocks in filesystem
    - Size of Blocks
    - Address of first free Data Block
    - Address of first free iNode
    - Various status flags

`tune2fs -l /dev/hda2` Full content of superblock of partition.

Content of Normal inodes:

Type and access rights
Number of hard links
UID
GID
Filesize in bytes
mtime (last content modified)
ctime(last properties modified)
atime(last time accessed)
Address of Block 0
....
Address of Block 9
Address of single-indirection block
Address of double-indirection block
Address of triple-indirection block

Block 0 to 9:	Block containing data
Single-indirection block:	Block listing up to 128 Data Blocks
Double-indirection block:	Block listing up to 128 Single-indirection Blocks
Triple-indirection block:	Block listing up to 128 Double-indirection Blocks

EXT2 Filesystem

The main difference between normal filesystems and the EXT2 is the content of the inodes which are slightly different to accommodate future expansion and special features.

Content of EXT2 inodes:

<table>
<tr><td><b>permissions</b></td><td colspan="2"><b>Nr. of Hard links</b></td><td><b>owner(UID)</b></td><td><b>group(GID)</b></td></tr>
<tr><td colspan="3"><b>size</b></td><td colspan="2"><b>properties change time(ctime)</b></td></tr>
<tr><td colspan="3"><b>modification time(mtime)</b></td><td colspan="2"><b>access time(atime)</b></td></tr>
<tr><td colspan="3"><b>deletion time(dtime)</b></td><td colspan="2"><b>blockcount</b></td></tr>
<tr><td colspan="3"><b>flags(attributes)</b></td><td colspan="2"><b>file version (NFS)</b></td></tr>
<tr><td colspan="3"><b>file ACL</b></td><td colspan="2"><b>dir ACL</b></td></tr>
<tr><td><b>fragment addr.</b></td><td><b>fr. size</b></td><td><b>frag. nr</b></td><td colspan="2"><b>reserved</b></td></tr>
<tr><td colspan="3"><b>1. block data</b></td><td colspan="2"><b>2. block data</b></td></tr>
<tr><td colspan="3"><b>3. block data</b></td><td colspan="2"><b>4. block data</b></td></tr>
<tr><td colspan="3"><b>5. block data</b></td><td colspan="2"><b>6. block data</b></td></tr>
<tr><td colspan="3"><b>7. block data</b></td><td colspan="2"><b>8. block data</b></td></tr>
<tr><td colspan="3"><b>9. block data</b></td><td colspan="2"><b>10. block data</b></td></tr>
<tr><td colspan="3"><b>11. block data</b></td><td colspan="2"><b>12. block data</b></td></tr>
<tr><td colspan="3"><b>simple indirect</b></td><td colspan="2"><b>double indirect</b></td></tr>
<tr><td colspan="3"><b>triple indirect</b></td><td colspan="2"><b>reserved</b></td></tr>
<tr><td colspan="3"><b>reserved</b></td><td colspan="2"><b>reserved</b></td></tr>
</table>

Some differences between EXT2 and normal filesystem:

- Deletion Time entry: Helpful for un-deleting files
- Field for 12 attributes(flags): `A,a,c,d,i,S,s,u`.

Useful are:	Append(`+a`)	Allows only to append to it via redirection(>>)
	Immutable(`+i`)	Not changeable, no new hardlinks, not deletable
	Safe-delete(`+s`)	Fills file with '0s' before deleting it.

  `chattr` *`attribute filename`* Changes the file's attributes. Only root allowed!!
  `chattr +i file` Turns attribute `i` ON
  `chattr -i file` Turns attribute `i` OFF
  `lsattr` *`filename`* Lists a file's attributes

- File Version Entry: Can be used by NFS server.
- File ACL and Dir ACL: (Access Control List) for better access control.
- Support for fragmented files
- 12 Direct Block Addresses instead of 10(standard)
- The Superblock has multiple backups of itself at the start of some block groups. I found between 6 to 10 copies (backups) in 3 to 10 GB Partitions

- Ext2 superblock has extra entries:
    - Valid-Flag Entry: if ON means the filesystem was not unmounted properly.
`e2fsck` uses this flag to know if it should do a full check.
`e2fsck -f` forces the full check.
    - Max-Mounts before full check and mount-count Entries: Used by e2fsck.
If mount-count=Max-Mounts before full check then e2fsck does a full check.
    - Percent of full partition space reserved for root Entry: Normally 5%.
    - tune2fs can manipulate these above superblock entries.
Warning: Partition should NOT be mounted as ReadWrite if changing any of these entries.

Journaling Filesystems

EXT3: Is an EXT2 filesystem with a journal file and journalling functions.

Reiserfs: Stores a report of all transactions bigger than 1 block.
Up to 10 times faster than EXT3 when reading.

Xfs: Ported from IRIX system. Meant for handling very large files.
max 9,000 Peta Bytes.
Info: Kilo, Mega, Giga, Terra, Peta, Exa.

1.104.1 Create partitions and filesystems

Weight: 3

Description: Candidates should be able to configure disk partitions and then create filesystems on media such as hard disks. This objective includes using various mkfs commands to set up partitions to various filesystems, including ext2, ext3, reiserfs, vfat, and xfs.

- **Key files, terms, and utilities:**
  `fdisk`
  `mkfs`

Notes:

Max number of Primary partitions per hard disks: 4
Max number of extended partitions per hard disk: 1
Maximum number of logical partitions per hard disk: 11
Partitions names:

```
hda----IDE-Ctrl1------hdb      hdc----IDE Ctrl 2-----hdd
master              slave      master              slave
hda1(pri/ext)
hda2(pri/ext)
hda3(pri/ext)
hda4(pri/ext)
     hda5(logic)
     hda6(logic)
     ..........
     hda15(logic)
```

Partitioning:

`fdisk -l`	Display all hard disks and partitions recognized in the system.
`fdisk /dev/hda`	Partition hda hard disk

```
Command (m for help):m
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)
```

Creating a new partition:

```
     n --> primary-->.....
```

Changing its partition system id:

```
     t--->l(list)-->83(linux) or 82(swap)
```

List partitions:

```
     p Shows the partition table
```

When all finished:(warning: last change to verify and correct if needed)

```
     w Writes the partition table on disk!!!
```

Note: Linux does not need the activation of the bootable flag, but Windows does. So if Windows is installed and the flag is on for its partition, then leave it there.

Creating a filesystem.

```
      mkfs -t filesystem [options] device [blocks]
or    mke2fs [options] device [blocks]     (for ext2 filesystem)

      filesystems:
           ext2,ext3,vfat,msdos,reiserfs,xfs,minix,bfs,xiafs
```

Note: Possible commands and their synonyms:

```
mke2fs    =  mkfs.ext2        = mkfs -t ext2
mkfs.ext3 =  mkfs.ext2 -j     = mkfs -t ext3
mkdosfs   =  mkfs.msdos  = mkfs.vfat = mkfs -t vfat
mkfs.xfs  =  mkfs -t xfs
mkfs.bfs, mkfs.minix,, mkfs.xiafs
```

options:

```
-b     Block size. Valid values: 1024, 2048, 4096
-c     Before creating the filesystem, check the device for bad blocks
-i n   Specify the number(n) of bytes per inode. Min = Block size
       This helps to calculate the number of inodes to create.
       Number of i-nodes = Dependent on the size of partition.
 -N n  Specify the absolute number(n) of i-nodes to create.
```

device: /dev/*xxxx* *xxxx*=hda1.... hdc4 etc.

blocks: Optional. Size in blocks of the filesystem to create.
If not given the size is auto detected.

```
      mkreiserfs      options device For making a reiser filesystem.
or    mkfs -t reiserfs  " "     " "
or    mkfs.reiserfs     " "     " "

      tune2fs -j device     Converts an ext2 to ext3 filesystem.
```

1.104.2 Maintain the integrity of filesystems Weight: 3

Description: Candidates should be able to verify the integrity of filesystems, monitor free space and inodes, and repair simple filesystem problems. This objective includes the commands required to maintain a standard filesystem, as well as the extra data associated with a journaling filesystem.

- **Key files, terms, and utilities:**
  `du, df`
  `fsck,e2fsck`
  `mke2fs`
  `debugfs`
  `dumpe2fs`
  `tune2fs`

Disk Usage: du

du is recursive by default.

`du -sh /root`	Display amount of space used by `/root` dirrctory
`du -h --max-depth=0 /home`	Amount of space used by /home (non recursive)

Disk Free: `df`

`df`	List (in kilobytes) free & used space on mounted partitions
`df -h`	Same as above but in human readable format (`K,M,G`)
`df -i /dev/hda3`	Show number of free inodes on `hda3` Note: `df -i` doesn't show the inodes info for reiserfs or XFS, since they create inodes dynamically.

File system check: `fsck`

Shortcut aliases for fsck:

`e2fsck = fsck.ext2`	For EXT2 and EXT3
`reiserfsck`	For Reiserfs
`fsck.minix`	For minix
`fsck.msdos`	For Ms DOS FAT
`fsck.vfat`	For DOS VFAT
`fsck.xfs`	For XFS

Note: `fsck` should always be run on a non-mounted or read-only mounted filesystem.

Syntax:

```
fsck options filesystem
```

`options`:(mostly for the ext2/ext3 filesystem)

`-A`	Checks all filesystems listed for check in `/etc/fstab`
`-f`	Force checking even if the *Valid-Flag* is not set(filesystem ok)
`-p`	Auto Repair without asking
`-n`	NO-Simulation. No writing of any changes on disk
`-y`	YES- Answer yes to any questions coming up. Dangerous!!!

File system debugging: `debugfs`

Interactive command driven debugging program.
Created to fully control and manipulate the `ext2` filesystems.
Default is in read-only mode. `-w` option overrides this.
Command `help` shows all valid commands.

File system info dump: `dumpe2fs`

Displays lots of information about the structure of the ext2 filesystem.

Incl: Superblock content, free inodes categorized per block groups
Location(offsets) of the superblock backups, etc.

Can be useful to be saved in a file and used to recover a damaged system.

File system tuning-up: `tune2fs`

Allows to manipulate some of the parameters of the ext2 filesystem located in the superblock. Heare are few examples:

`tune2fs -l /dev/hda5`	List the content of the superblock
`tune2fs -j /dev/hda3`	Converts the ext2 filesystem to ext3
`tune2fs -c 30 /dev/hda2`	Change the max-mount-count to 30
`tune2fs -C 0 /dev/hda9`	Reset the number of mounts counter to 0.

Extra examples:

```
dumpe2fs /dev/hda7 | grep '[mM]ount count'
   dumpe2fs 1.19, 13-Jul-2000 for EXT2 FS 0.5b, 95/08/09
   Mount count:              7
   Maximum mount count:      20
```

```
tune2fs -C 9 /dev/hda6
  tune2fs 1.19, 13-Jul-2000 for EXT2 FS 0.5b, 95/08/09
   Setting current mount count to 9
```

1.104.3 Control mounting and unmounting filesystems Weight: 3

Description: Candidates should be able to configure the mounting of a filesystem. This objective includes the ability to manually mount and unmount filesystems, configure filesystem mounting on bootup, and configure user mountable removeable filesystems such as tape drives, floppies, and CDs.

- **Key files, terms, and utilities:**
  `/etc/fstab`
  `mount`
  `umount`

- **Systax of Mounting command**

```
mount -t <fstype>  <SourceDevice>  <MountPoint>
```

eg. `mount  /dev/hdc  /cdrom`

- **Mounting all the `fstab` -auto- (boot time only) mount points**
  `mount -a` tries to mount all the devices in fstab as it happens at boot time.

- **`/etc/fstab` file format**

	Device	Mount point	Files system	Options	Dump	fsck order
e.g.:	/dev/hda1	/boot	ext2	defaults	1	1
	/dev/hdb1	/	ext2	defaults	0	2
	/dev/hdb3	swap	swap	defaults	0	1
	/dev/cdrom	/cdrom	iso9660	ro,noauto,user	0	0
	/dev/floppy	/floppy	auto	noauto,user	0	0
	/dev/hdc1	/windows	vfat	user,umask=000	0	0

- **Options of 'defaults'**
  `rw,suid,dev,exec,auto,nouser,async,atime`
  (async=buffered)

- **List of all options**

auto	**noauto**	Mounting at boot time ?
exec	**noexec**	Execute binaries found on device ?
sync	**async**	Buffered data when writing ?
atime	**noatime**	Update inode access time when accessed ?
dev	**nodev**	Accept special character and block devices ?
suid	**nosuid**	Allow suid on mounted file system ?
user	**nouser**	Allow user to mount device ?
rw	**ro**	Read/Write(rw) or Read only(ro) ?
remount		Remount the already mounted device.
umask=		Sets the umask for writing on the partition (good for vfat)

Notes:

- The option **`user`** implies: `noexec,nosuid` and `nodev` unless overridden by subsequent contradictory options.
- Schreibeberechtigung als Benutzer auf einer **vfat Partition**:
  `user,umask=000`
- The option **`mount -w`** ... is the same as `mount -o rw`
- Almost all options can also be entered using `mount -o`
  E.g. `mount -o ro,umask=000 -t vfat /dev/hdd /windows`

- **Display already mounted devices**

- `mount`	Most complete info
- `cat /etc/mtab`	Not always refreshed immediately
- `cat /proc/mounts`	Always current
- `df -h`	Mounted devices and space used/free

1.104.4 Managing disk quota Weight: 3

Description: Candidates should be able to manage disk quotas for users. This objective includes setting up a disk quota for a filesystem, editing, checking, and generating user quota reports.

- **Key files, terms, and utilities:**

```
quota
edquota
repquota
quotaon
```

- **Terms of quota editing and reports:**
  1. The user is allowed to cross the *soft limit* for the length of time limited by the *grace period,* after which he's not allowed to write anything on the partition.
  2. The *hard limit* may never be exceeded by the user.
  3. The quota limits may be expressed in number of 1k blocks or in number of inodes (total number of files and directories) or both.

- **Procedure for installing quota for users and groups:**(short form)
  - Edit `/etc/fstab` and enter `usrquota`,`grpquota` in options field for filesystem

```
/dev/hda3  /home     ext2 defaults,usrquota,grpquota    1  1
```

  - Remount the filesystem:

```
mount -o remount /dev/hda3
```

  - Initialize the quota databases files(`aquota.user,aquota.group`)

```
quotacheck -avugm
```

  - Set quota for each user:

```
edquota -u paul or
edquota paul
```

  - Edit grace period for all the users:

```
edquota -tu
```

  - Turn quotas ON :

```
quotaon -u /dev/hda3
```

  - Check quota for user:

```
quota paul
```

  - Create a quota report for all users:

```
repquota -u /dev/hda3
```

  - Create a quota report for all groups:

```
repquota -g /dev/hda3
```

  - Turn quotas OFF(when needed)

```
quotaoff -u /dev/hda3
```

- **Detailed preparation of quotas.**

- Enter the following options in `/etc/fstab` for the partitions that must use quotas.

  eg.

```
/dev/hda2 /srv/www  ext2 defaults,usrquota,grpquota    1  1
/dev/hda3 /home     ext2 defaults,usrquota,grpquota    1  1
```

- Remount the filesystems:

```
mount -o remount /srv/www
mount -o remount /home
```

- Enter the following command to verify existing used space by each user and group:

```
quotacheck -avugm
```

  This command will also update two files in the `/home` directory:

  `quota.group`, and `quota.user`

  if version 2 of quotas is used then the two files will be:

  `aquota.group`, and `aquota.user`

- Start editing the quota for each user:

  eg. `edquota -u john` or `edquota john`

  Edits the filesystem quota for the user john

  The quota editor(`vi`) will appear and will allow changes to the soft and hard quota for user `john`. Note: The value 0 for `soft` or `hard` quota means <u>NO LIMIT</u>.

```
+-------------------------------------------------------------------+
| Filesystem      blocks      soft      hard      inodes     soft    hard   |
| /dev/hda7        3288       4000      6000       649       2000    3000   |
+-------------------------------------------------------------------+
```

- This above example means that `john`:

```
Already uses 3288 blocks(kb) of data on /dev/hda7 in 649 inodes (files)
The soft quota is set to 4000 kB and hard to 6000 kB
The soft limit is set to 2000 inodes and hard limit to 3000 inodes
```

- `edquota -tu` Edits grace period for <u>all</u> users.

  Not possible to set grace period for individual users

  eg. `(month(s),day(s),hour(s),min(utes),sec(onds))`

```
+-------------------------------------------------------------------+
| Filesystem      Block grace period       Inode grace period                |
| /dev/hda7               7days                   5days                      |
+-------------------------------------------------------------------+
```

  - To copy the quota for other users with the same limit values, easiest way is:

```
edquota -p john patrick
```

    This command will give `patrick` the same quota limits as `john.`

  - To verify the status of the quota for the user `john` use the commands:

```
su -
quota john
```

<u>The result:</u>

```
+-----------------------------------------------------------------+
| Disk quotas for user john (uid 5001):                           |
| Filesytem blocks  quota  limit grace  files  quota  limit  grace |
| /dev/hda7 3288     4000   6000          649   2000   3000        |
+-----------------------------------------------------------------+
```

This means that the user `john` has `649` files using `3288` Kb of hard disk space.
His soft limit is `4000` Kb or `2000` inodes and hard limit is `6000` kb or `3000` inodes

Repquota

`Repquota` produces summarized quota information for a file system.
Here is a sample of the output that `repquota` gives:

```
# repquota -a
     *** Report for user quotas on device /dev/hda7
     Block grace time: 7days; Inode grace time: 5days
                  Block limits                    File limits
     User          used    soft    hard  grace    used  soft  hard  grace
     root    -- 175419       0       0           14679    0     0
     john    +-    6000    4000    6000             650 2000  3000
     uucp    --     729       0       0              23    0     0
     user1   --   13046   15360   19200             806 1500  2250
```

`repquota -g /home` Report of <u>groups</u> quota
`repquota -u /home` same as `repquota /home` Report of <u>users</u> quota

- **`Quotaon and Quotaoff`**

`quotaon  -u /dev/hda2` turns ON quota accounting in kernel for users(`-u`)
`quotaoff -u /dev/hda2` turns it OFF.

Actually both files are similar. They are executed at system startup and shutdown.

<u>**Important Files involved in Quota**</u>

`quota` `(1)` Display disk usage and limits. `quota` reports the quotas of all filesystems listed in `/etc/mtab`. For mounted NFS filesystems , a call to `rpc.rquotad` on the server machine is performed to get the information.
`setquota` `(8)` Set disk quotas with one command without editing like edquota
`edquota` `(8)` Edit user quotas
`quotaoff` `(8)` `[`**`quotaon`**`]` Turn filesystem quotas on and off
`quotacheck` `(8)` Scan a file system for disk usage, create and check the files `aquota.user` and `aquota.group`
`repquota (8)` Summarize quotas for a filesystem

1.104.5 Use file permissions to control access to files Weight: 5

Description: Candidates should be able to control file access through permissions. This objective includes access permissions on regular and special files as well as directories. Also included are access modes such as suid, sgid, and the sticky bit, the use of the group field to grant file access to workgroups, the immutable flag, and the default file creation mode.

- **Key files, terms, and utilities:**

```
chmod
umask
chattr
```

- **File types**
  - (-) Regular files......(s) is unknow to me till now (eg. `/dev/gpmctl`)
  - (**l**) Symbolic Links (eg. `/sbin/init.d/rc2.d`......all files)
  - (**d**) Directories and sub-directories
  - (**b**) Block Device Files (eg. `/dev/hda1`...)
  - (**c**) Character Device Files (eg. `/dev/tty1`....)
  - (**p**) FIFO Named pipes (eg. `/dev/log`, `/dev/xconsole`)
  - (**s**) Socket File (eg. `/var/spool/postfix/private/bounce`...)

  Note. File and directory names that start with a Dot (**.**) are hidden from display by certain programs like `ls` etc.

- **Files and directories access rights**
  Access rights are restrictions applied to the content of a file or directory.
  It doesn't restrict the deletion of a file or directory. Only the parent directory's access rights controls that.

  - **Changing a file's access rights**

```
chmod [-R] [ugoa][+=-][rwx stXugo] or [0000 to 7777] file
eg.   chmod u+w,g-x,o=wx file1
      chmod 750 file2
      chmod 4755 program1            (SUID=ON)
      chmod u+s,g+s,o+t program2     (SUID=ON,SGID=ON,StickyBit=ON)
      chmod -R u=rwX,g=rX,o=rX dir1  Recursive 755 for directories.
                                     and 644 for files.
```

  - **Directory's access rights**
    - The read(`r`) without the search(`x`) access rights for directories makes no sense and the read is ignored.
    - Any file in a directory set to write access for everybody can be erased by anybody, regardless of who the current user is.

Extra access rights			**u**ser			**g**roup			**o**thers		
SUID (s)	**SGID (s)**	**Sticky Bit(t)**	**r**	**w**	**x**	**r**	**w**	**x**	**r**	**w**	**x**
4	**2**	**1**	**4**	**2**	**1**	**4**	**2**	**1**	**4**	**2**	**1**

– **SUID and SGID for programs** (`-rwsrwsrwx`)(`-rwSrwSrwx`)
  – SUID=ON: Effective user is the owner of the program
  – SGID=ON: Effective group is the group owner of the program

- **SGID for Directories**

  Forces subdirectories and files created in it to have the same group as the directory's group independent of the creating user's group. Subdirectories created within this directory will inherit the same SGID.

- **Sticky Bit for <u>Directories</u> :**

  Files in the directory can only deleted by their owner even if the directory is set to write access for all. Sticky bit is normally set on /tmp to prevent another user's processes from deleting your files.
  Note 1: Normally any file (belonging to the user or not) under a directory set to write access forgroup or others can be erased by any user.
  Note2: The **owner of the directory** can erase any file in it even if the sticky bit is set.

- **<u>Sticky Bit for programs</u>:**
  - Allows a running program to be stored in ram (buffers) until the system goes down. Advantage: Programs load faster..
    Disadvantage: Uses lots of RAM

    Note: Sticky bit for programs is obsolete. Linux has never used it and no modern Unix has used it for years – swap memory does the same thing more effectively

  - **Command :**

    `chmod o+t` (sets the sticky bit)
    result= (`-rwxrwxrwt`) or (`-rwxrwxrwT`)
    `chmod u+s` (sets the SUID)
    result= (`-rwsrwxrwt`) or (`-rwSrwxrwx`)
    `chmod g+s` (sets the SGID)
    result= (`-rwxrwsrwt`) or (`-rwxrwSrwx`)

    Note: When adding a sticky-bit to a file/dir with an **<u>x</u>** for Others, the sticky-bit is displayed as **<u>t</u>** otherwise as **<u>T</u>** if the x was not present. The same applies to SUID and SGID (-rw**<u>S</u>**rw**<u>S</u>rw<u>T</u>**)

- **Attributes(`chattr` & `lsattr`)**

  - Setting the 'append only' attribute on a directory or file.
  - Command : **`chattr +a`** *`filename`* or *`directoryname`*
  - User must necessarily not be root
  - A file with this attribute may be appended to, but may not be deleted, and the existing contents of the file may not be overwritten. If a directory has this attribute, any files or directories within it may be modified as normal, but no files may be deleted.

  - Setting the 'immutable' attribute on a directory or file.
  - Command : **`chattr +i`** *`filename`* or *`directoryname`*
  - User must be ´root´
  - A file or directory with this attribute may not be modified, deleted, renamed, or (hard) linked

  - **Display Attributes of files and directories**

    To list the (special) attributes of files and dirs, use the command **`lsattr`**

- **Attributes list:**

A Atime record is not modified. Prevents too much disk access for laptops. Still in testing mode

a Sets it to append mode only (can not erase it)
Only root can set this attribute

c The kernel compresses this file before writing to disk
The kernel decompresses it when reading it from disk.
NOT Implemented yet by kernel

d Will not be backed up by the program "dump"

i Cannot be modified
Cannot be erased
Cannot be renamed
Cannot be hard linked
Only root can change this attribute

s When this file is erased, the blocks it used are over-written with '0' to prevent recovery at a later date.

S Any change to this file will be immediately written to the disk instead of in the file system buffer.
(equivalent to 'sync' mount option)

u When this file is deleted, its content are saved. It can therefore be undeleted later.
NOT implemented yet by kernel.

- **`umask` for new files an directories**

Sets default access rights for newly created files and directories:
New file access rights = `666 !| umask` (`!|`=Logical NOR)
New directory access rights = `777 !| umask`
Note: umask specifies which attributes will NOT be applied

- examples:

666 access rights 777

umask	New files	New Directories
022	-rw-r--r--	-rwxr-xr-x
135	-rw-r---w-	-rw-r---w-
216	-r--rw----	-r-xrw---x

1.104.6 Manage file ownership Weight: 1

Description: Candidates should be able to control user and group ownership of files. This objective includes the ability to change the user and group owner of a file as well as the default group owner for new files.

- **Key files, terms, and utilities:**

```
chown
chgrp
chmod
```

chown : Change user and group ownership of a file or directory

Syntax:
```
chown [options] [user][:group] filename
chown [options] [user][:group] dirname
```

eg.

	chown	*user*:*group*	*filename*	Change user and group ownership of file
or	"	*user*	"	Change user ownership of file
or	"	*user*.	"	Change user and his group ownership of file
or	"	*user*:	"	" " "
or	"	.*group*	"	Change group ownership of file

Important Options:(from man page)

`-R --recursive` Recursive:affects all files and directories inside directory trees

`--dereference` affect the referent of each symbolic link, rather than the symbolic link itself.

`-h, --no-dereference`
affect symbolic links instead of any referenced file. (available only on systems that can change the ownership of a symlink)

`--from=`*`CURRENT_OWNER`*`:`*`CURRENT_GROUP`*
change the owner and/or group of each file only if its current owner and/or group match those specified here.
Either may be omitted, in which case a match is not required for the ommitted attribute.

`-f, --silent, --quiet` suppress most error messages

`-c, --changes` like verbose but report only when a change is made

`--reference=RFILE` use `RFILE`'s owner and group rather than the specified `OWNER:GROUP` values.

`-v, --verbose` output a diagnostic for every file processed

IMPORTANT:

`root` is the only user allowed to change ownership(`chown`) of files or directories.

chgrp : Change group ownership of a file or directrory

syntax: `chgrp [`*`options`*`]` *`newgroup filename`*

eg. `chgrp -R ftp /srv/www`
Changes recursively all the files and directories inside the dir. `/srv/www` to be owned by group `ftp`

`chgrp -R --reference=/home/hans /srv/ftp`
Changes recursively the group ownership of all the files and directories contained in `/srv/ftp` to the group owning the directory `/home/hans`

options:

`-R, --recursive` operate on files and directories recursively

`--dereference` affect the referent of each symbolic link, rather than the symbolic link itself

`-h, --no-dereference`
affect symbolic links instead of any referenced file (available only on systems that can change the ownership of a symlink)

`-f, --silent, --quiet` suppress most error messages

`--reference=RFILE` use `RFILE`'s group rather than the specified `GROUP` value

`-v, --verbose` output a diagnostic for every file processed

`-c, --changes` like verbose but report only when a change is made

chmod : Change the access rights of a files or directories

See 1.104.5 Use file permissions to control access of files

1.104.7 Create and change hard and symbolic links Weight: 1

Description: Candidates should be able to create and manage hard and symbolic links to a file. This objective includes the ability to create and identify links, copy files through links, and use linked files to support system administration tasks.

- **Key files, terms, and utilities:**
  `ln`

Creating a Symbolic link:

Syntax: `ln [options] -s` *`source destination`*(newlink)
or `cp -s` *`source destination`*(newlink)

eg. `ln -s /bin/cat /home/hans/bin/cat`
Creates a new symbolic link called `/home/hans/bin/cat`
pointing to `/bin/cat`

Creating a Hard Link: (files which have the same inode)

Syntax: `ln` *`source destination`*(newlink)
or `cp -l` *`source destination`*(newlink)
eg. `ln /bin/ping /home/hans/bin/ping`
Creates a new hard link called `/home/hans/bin/ping`
pointing to `/bin/ping`

Options:

`-f, --force` remove existing destination files

`-i, --interactive` prompt whether to remove destinations

`-s, --symbolic` make symbolic links instead of hard links

`--target-directory=`*`DIRECTORY`*
specify the *`DIRECTORY`* in which to create the links

`-v, --verbose` print name of each file before linking

Important Notes:

- Although the man page says that it is possible to make a hard link to a directory, in reality it is not possible...yet. Better us the special 'bind' mounting.
  eg. `mount /SourceDir /mountpoint -o bind`

- Hard links are limited to the same partition as the original

- Symbolic links are NOT limited to the same partition as the original

The command `cp` *`source destination`* copies the referenced file '
(the file that the sym link points to) when the *`source`* is a symbolic link.
eg. `cp linktest3 linktest5` (`linktest3` is a symbolic link---->`linktest`)
Copies the content of `linktest` to `linktest5` as a normal file.

- The field Nr.2 of the command `ls -l filename` shows how many files are hard linked to that inode.

eg. `ls -l linktest*`

```
-rw-r--r--  3 michel   video   0 2003-11-20 08:45 linktest
-rw-r--r--  3 michel   video   0 2003-11-20 08:45 linktest2
-rw-r--r--  3 michel   video   0 2003-11-20 08:45 linktest3
```

- The command `stat filename` also shows also how many files are hard linked to that inode.

```
eg.
stat linktest
File: `linktest'
Size: 0     Blocks: 0       IO Block: 4096   regular empty file
Device: 305h/773d       Inode: 876319      Links: 2
Access: (0644/-rw-r--r--) Uid: (  500/ michel)  Gid:( 33/  video)
Access: 2003-11-20 08:45:10.000000000 +0100
Modify: 2003-11-20 08:45:10.000000000 +0100
Change: 2003-11-20 08:45:22.000000000 +0100
```

1.104.8 Find system files and place files in the correct location Weight: 5

Description: Candidates should be thoroughly familiar with the Filesystem Hierarchy Standard, including typical file locations and directory classifications. This objective includes the ability to find files and commands on a Linux system.

- **Key files, terms, and utilities:**

```
find
locate
slocate
updatedb
whereis
which
/etc/updatedb.conf
```

find : Finding file recursively

Syntax:

```
find startdirectory [search_criteria_options] [-exec command \;]
```

Examples:

```
find . type d -maxdepth 1 | sort
```

Finds all directories located in the current directory and sort them

```
cd /etc/ ; find . -name "*XF*"
```

Recursively finds all files in directory `/etc` whose names include the pattern 'XF'

```
find /opt/kde -maxdepth 2 -type f -name "*edit"
```

Searches /opt/kde and subdirectories up to 2 levels deep for files whose names end with the word 'edit'

```
find . -follow -cmin -5
```

Search the current directory for files whose properties were changed less than 5 minutes ago

`-cmin +5`	Properties of file **c**hanged more than 5 minutes ago
`-amin -6`	Content of file **a**ccessed less than 6 minutes ago
`-mmin +8`	Content of file **m**odified more than 8 minutes ago
`-ctime +5`	Properties of file **c**hanged more than 5 days ago
`-atime -7`	Content of file **a**ccessed less than 7 days ago
`-mtime -3`	Content of file **m**odified less than 3 days ago

```
find /etc -type f -name '*.conf' -exec grep -H "hosts" {} \;
```

Searches the /etc directory for files with the extension .conf. Executes grep on those files looking for the string “hosts”. When found, also displays the filename it was found in.

```
find /etc -type f -name '*.conf' -ok grep -H "hosts" {} \;
```

Same actions as above except that `-ok` option asks `find` to prompt for confirmation (with **y**) of the command before executing it.

locate : Locate files in the whole system using a database of filenames.

Syntax:

`locate` *`filename`* Searches the locate database for the *`filename`*.

This database is in	`/var/lib/locatedb`
It is updated via the command:	`updatedb [`*`options`*`]`
The configuration file for updatedb is	`/etc/updatedb.conf`

`-d path, --database=path`

Instead of searching the default file name database, search the file name databases in path, which is a colon-separated list of database file names. You can also use the environment variable `LOCATE_PATH` to set the list of database files to search.
The option overrides the environment variable if both are used.

`-e, --existing` Only print out such names that currently exist (instead of such names that existed when the database was created).
Note that this may slow down the program a lot, if there are many matches in the database.

`-i, --ignore-case`

Ignore case distinctions in both the pattern and the file names.

slocate : Secure locate of file in system

Syntax:

`slocate [options] filename`

Secure Locate provides the same features as `locate` but it will also store file permissions and ownership so that users will not see files they do not have access to.

`slocate` database is not the same as `locate` database. It needs to be built by issuing the `slocate` command with proper options:

Database Build Options:

`-u` Create slocate database starting at path /

`-U <dir>` Create slocate database starting at path <dir>

`-e <dir1,dir2,...>`

Exclude directories from `slocate` database .

`-f <fstype1,...>`

Exclude files on specific file systems from the `slocate` database.

`-c` Parse `/etc/updatedb.conf` when updating the slocate database.

`-l <level>` Security level:

`0`	Turns security checks off. This will make searches faster.
`1`	Turns security checks on. This is the default.

`-o <file>` Specifies the database to create.
`--output=<file>`

`-v, --verbose` Verbose mode.
Display files when creating database.

Slocate Search Options:

`-i` Does a case insensitive search.

`-q` Quiet mode. Error messages are suppressed.

`-n <num>` Limit the amount of results shown to `<num>`.

`--regexp=<regexp>`
`-r <regexp>` Search the database using a basic POSIX regular expression.

`-d <path>` Specifies the path of databases to search.
`--database=<path>`

whereis: Search for a program and possibly its man pages from a predefined path.

Syntax:

`whereis` *`filename`* Searches a predefined (hard coded) list of directories for the filename and man pages. They must be in the path predefined during compilation of `whereis` program.

which : Search for the first occurance of a program in the `PATH`.

Syntax:

`which` *`filename`* Searches the `PATH` for the first occurance of the filename. The filename can be a list of files.

`type -p` *`filename`* Same as above `which` *`filename`*

Topic 110: The X Window System

1.110.1 Install & Configure XFree86 Weight: 5

Description: 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 customizing 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:**

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

- **X Server**

X-Server offers an empty display where programs that support the X-Protocol will be displayed and controlled via the mouse and keyboard.
The X-Server takes control of the local Graphic card, monitor, mouse and keyboard and possibly other devices like joystick, graphic tablet etc.
The X-Server is a network service for local or remote clients(X-Programs).
The X-Server has been developed for many hardware platforms. Most X-Server implementations are proprietary. XFree86is free. It is the one explained below.

`/etc/X11/XF86Config`	Main XF86 Configuration file. XFree86 looks for it by default.It is the first looked for. Before FHS it was often found at `/etc/XF86Config` If the user is root then XFree86 will first search for `~/XF86Config`

- **Search Path of XF86Config file.**

-When X is started as a normal user:

```
/etc/X11/$XF86CONFIG
/usr/X11R6/etc/X11/$XF86CONFIG
```

Then Common search path

-When X is started as the `root` user.

```
$XF86CONFIG
/etc/X11/$XF86CONFIG
/usr/X11R6/etc/X11/$XF86CONFIG
$HOME/XF86Config
```

Then Common search path

Common search path:

```
/etc/X11/XF86Config-4
/etc/X11/XF86Config
/etc/XF86Config
/usr/X11R6/etc/X11/XF86Config.<hostname>
/usr/X11R6/etc/X11/XF86Config-4
/usr/X11R6/etc/X11/XF86Config
/usr/X11R6/lib/X11/XF86Config.<hostname>
/usr/X11R6/lib/X11/XF86Config-4
/usr/X11R6/lib/X11/XF86Config
```

- **Note:** In the above 2 paths ***/X-Rootdir*** is normally `/usr/X11R6/`
  Depending on distributions the configuration files of X-Server Version 3 and Version 4 are located in different locations. Often used locations and names:

`/etc/XF86Config`	Version 3
`/etc/X11/XF86Config`	Verison 4

  or

`/etc/X11/XF86Config`	Version 3
`/etc/X11/XF86Config-4`	Verison 4

- **XF86 Configuration programs** (Helper programs that write an `XF86Config` file)

`xf86config` First text-based configuration program.
Provided and supported by the XFree86 development team.
Belongs to standard X-Server packages.

`XF86Setup` Graphic-based (640x480-VGA 16 colors) configuration program.
Also provided and supported by XFree86 development team.

`xf86cfg` Graphic-based configuration program. More complex and more for advanced administrators. Provides the possibility of dynamically tring some of the settings by pressing an 'Apply' button.
Provides auto-detection of graphic cards.
Also provided and supported by XFree86 development team.

SAX & SAX2 SuSE graphics-based configuration programs.
SAX is for XFree86 Version 3 and SAX2 for XFree86 Version 4.
Provides auto-detection of graphic cards.

`Xconfigurator`
RedHat text-based configuration programs. Improved version of `xf86config`. It does auto-detection of graphic cards. Works in interactive mode or in automatic-install mode.

`dexconf` Background Debian system installation program.
No user startable program. To reconfigure the X-Server execute:
`dpkg-reconfigure xserver-xfree86`

All of the above configuration programs do 2 things:
- Configuration of the `XF86Config` file.
- Creation of a symbolic link to the configured X-Server

- **Configuration of the `XF86Config` file** :
  Sections of XF86Config file:

`ServerFlags`	Server flags
`Module`	Dynamic module loading
`InputDevice`	Input device description
`Device`	Graphics device description
`VideoAdaptor`	Xv video adapter description
`Monitor`	Monitor description
`Modes`	Video modes descriptions
`Screen`	Screen configuration

`ServerLayout`	Overall layout
`DRI`	DRI-specific configuration
`Vendor`	Vendor-specific configuration

Note: Although the next 2 sections are recognized by version 4
it is recommended to use InputDevice section instead.

`Keyboard`	Keyboard configuration(Version 3)
`Pointer`	Mouse configuration(Version 3)

- **Creation of a symbolic link to the configured X-Server**

  For Version 3

  ```
  /usr/X11R6/bin/X ==> /var/X11R6/bin/X ==>
                               /usr/X11R6/bin/XF86_Servertype
  ```

  For Version 4

  ```
  /usr/X11R6/bin/X ==> /var/X11R6/bin/XFree86
  ```

- **Starting the Xserver and Windowmanager:**

  `startx` (script)

- **Fine Tuning the monitor settings:**

  - Manually with monitor's buttons
  - Via the `xvidtune` program

- **X11 Fonts and Fonts server:**

  Fonts are listed in XF86Config through the Keyword: `FontPath`
  eg.

  ```
   FontPath "/usr/X11R6/lib/X11/fonts/TrueType"
   FontPath "/usr/X11R6/lib/X11/fonts/75dpi:unscaled"
  ```

  Font servers can also be listed but MUST be first in the list:

  `FontPath "unix/:7100"` Font server on local Unix socket

- **`xset`: Temporarily Changing the X-Server's `FontPath` settings as it runs:**

  Example of adding and deleting FontPaths (non permanent).

  ```
     xset  +fp /usr/X11R6/lib/X11/fonts/TrueType   Adds a FontPath
  or xset  fp+/usr/X11R6/lib/X11/fonts/TrueType

     xset  -fp /usr/X11R6/lib/X11/fonts/TrueType   Deletes a FontPath
  or xset  fp-/usr/X11R6/lib/X11/fonts/TrueType
  ```

- **Setting-up a Font server:XFS**

  `xfs` is the standard Font Server which listens for requests on port `7100`(Standard).

  - Settings of client's `XF86Config` configuration file:

    ```
    FontPath "unix/:7100"                  Local Font server on Unix socket
    #FontPath "tcp/myserver.fd.com:7100"  Remote font server
    ```

  - xfs Configuration file:`/etc/X11/fs/config` or `/etc/X11/xfs.conf`

  - Starting the font server as a daemon:

    ```
    xfs -config /etc/X11/fs/config -daemon
    ```

- Example xfs configuration file:

```
no-listen = tcp
port = 7100
client-limit = 10
clone-self = on
use-syslog = on
deferglyphs = 16

catalogue = /usr/X11R6/lib/X11/fonts/misc:unscaled,
            /usr/X11R6/lib/X11/fonts/75dpi:unscaled,
            /usr/X11R6/lib/X11/fonts/100dpi:unscaled,
            /usr/X11R6/lib/X11/fonts/japanese:unscaled,
            /usr/X11R6/lib/X11/fonts/baekmuk:unscaled,
            /usr/X11R6/lib/X11/fonts/Type1,
            /usr/X11R6/lib/X11/fonts/URW,
            /usr/X11R6/lib/X11/fonts/Speedo,
            /usr/X11R6/lib/X11/fonts/CID,
            /usr/X11R6/lib/X11/fonts/PEX,
            /usr/X11R6/lib/X11/fonts/cyrillic,
            /usr/X11R6/lib/X11/fonts/latin2/misc,
            /usr/X11R6/lib/X11/fonts/latin2/75dpi,
            /usr/X11R6/lib/X11/fonts/latin2/100dpi,
            /usr/X11R6/lib/X11/fonts/latin2/Type1,
            /usr/X11R6/lib/X11/fonts/latin7/75dpi,
            /usr/X11R6/lib/X11/fonts/kwintv,
            /usr/X11R6/lib/X11/fonts/truetype,
            /usr/X11R6/lib/X11/fonts/uni,
            /usr/X11R6/lib/X11/fonts/ucs/misc,
            /usr/X11R6/lib/X11/fonts/ucs/75dpi,
            /usr/X11R6/lib/X11/fonts/ucs/100dpi,
            /usr/X11R6/lib/X11/fonts/hellas/misc,
            /usr/X11R6/lib/X11/fonts/hellas/75dpi,
            /usr/X11R6/lib/X11/fonts/hellas/100dpi,
            /usr/X11R6/lib/X11/fonts/hellas/Type1

# in decipoints
default-point-size = 120
default-resolutions = 75,75,100,100
# font cache control, specified in KB
cache-hi-mark = 2048
cache-low-mark = 1433
cache-balance = 70
```

- **Fonts names Format.**

```
Author       Weight   Width     Pixels  XRes   Spacing   ISO-Standard
    |           |        |         |      |     |        |
  -b&h-lucida-medium-r-normal-sans-18-180-75-75-p-106-iso8859-1
         |           |          |      |     |     |          |
     Fontname     Attribute    Style   Points YRes Average Options
                  (i or* =Italic)       (1/72 in)   Width
                   (r =roman)
```

- **Installing new fonts:**
New fonts need some preparation before they can be used.
Besides the font files (with extensions `.bdf .snf .pcf`)
located in the font directories, some extra files need attention:

`fonts.dir` Contains the number of fonts available in this directory (on first line) and one line per font description. The Format is:

First line: Number of fonts listed in this file.(eg. 439)
Rest of file: *`FontFilename Font_Description`*

eg.

```
439
putbi.pfa -adobe-Utopia-bold-i-normal--0-0-0-0-p-0-adobe-standard
putbi.pfa -adobe-Utopia-bold-i-normal--0-0-0-0-p-0-iso10646-1
putbi.pfa -adobe-Utopia-bold-i-normal--0-0-0-0-p-0-iso8859-1
...........................
```

To create this file the program `mkfontdir` must be run:
Syntax:

`mkfontdir` *`/path/to/font/directory`*

Valid font types: PCF (`.pcf`), SNF (`.snf`) and BDF (`.bdf`) `fonts.alias`

List entered by manually assigning a non existing font name to an existing one

Format: *`fictive_name existing_name`*

eg.

```
fixed         -misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso8859-1
variable      -*-helvetica-bold-r-normal-*-*-120-*-*-*-*-iso8859-1
5x7           -misc-fixed-medium-r-normal--7-70-75-75-c-50-iso8859-1
5x8           -misc-fixed-medium-r-normal--8-80-75-75-c-50-iso8859-1
6x9           -misc-fixed-medium-r-normal--9-90-75-75-c-60-iso8859-1
6x10          -misc-fixed-medium-r-normal--10-100-75-75-c-60-iso8859-1
6x12          -misc-fixed-medium-r-semicondensed--12-110-75-75-c-60-iso8859-1
6x13          -misc-fixed-medium-r-semicondensed--13-120-75-75-c-60-iso8859-1
6x13bold      -misc-fixed-bold-r-semicondensed--13-120-75-75-c-60-iso8859-1
7x13          -misc-fixed-medium-r-normal--13-120-75-75-c-70-iso8859-1
7x13bold      -misc-fixed-bold-r-normal--13-120-75-75-c-70-iso8859-1
7x13euro      -misc-fixed-medium-r-normal--13-120-75-75-c-70-iso8859-15
7x13eurobold -misc-fixed-bold-r-normal--13-120-75-75-c-70-iso8859-15
7x14          -misc-fixed-medium-r-normal--14-130-75-75-c-70-iso8859-1
7x14bold      -misc-fixed-bold-r-normal--14-130-75-75-c-70-iso8859-1
8x13          -misc-fixed-medium-r-normal--13-120-75-75-c-80-iso8859-1

etc................
```

`fonts.scale`

List of fonts that are scalable. The format is:

First line: Number of fonts listed in this file.(eg. `439`)
Rest of file: *`FontFilename Font_Description`*

eg.

```
439
putbi.pfa -adobe-Utopia-bold-i-normal--0-0-0-0-p-0-adobe-standard
putbi.pfa -adobe-Utopia-bold-i-normal--0-0-0-0-p-0-iso10646-1
putbi.pfa -adobe-Utopia-bold-i-normal--0-0-0-0-p-0-iso8859-1
..................
```

- **Controlling X-Server settings.**
  - <u>Dynamic settings:</u> X-Server can be dynamically (non permanently) controlled via the command `xset`.
    eg. To immediately change the keyboard's delay (250 ms) and repeat rate (30/s):
    `xset r rate 250 30`
  - <u>User controlled settings:</u> X-Server can also be controlled to provide certain configurations when X-Programs are started using the `~/.Xresources` file.
    Note: In SuSE `~/.Xresources` is a symbolic link to `~/.Xdefaults`
  - <u>Sequence for reading resource files</u>: Global config files for each separate X-program are first read from the directory:
    `/usr/lib/X11/app-defaults/*` and then the `~/.Xresources` file is read.
    Any user- and machine-specific resources may be specified by setting the `XENVIRONMENT` environment variable to the name of a resource file to be loaded by all applications. If this variable is not defined, a file named:
    `~/.Xdefaults-`*`hostname`* is looked for instead, where *`hostname`* is the name of the host where the application is executing.

- File Format for resources files ..../`app-defaults/*` and `~/.Xresources`:
  `X-ProgramName*attribute: value`
  eg.(commented lines start with a '`!`')

```
xterm*background:        LightYellow2
xterm.eightBitInput:     true
! xterm*font:    -adobe-courier-bold-r-normal--14-140-75-75-m-90-iso8859-1
```

These parameters can be overridden by starting an X-Program with arguments.

```
eg. xterm  -fn 9x15bold -geometry 100x40+30+40 -bg LightYellow2\
        -T "Test_Xterm" -sb -rightbar
```

1.110.2 Setup a display manager Weight: 3

Description: Candidate should be able setup and customize 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:**
  `/etc/inittab`
  `/etc/X11/xdm/*`
  `/etc/X11/kdm/*`
  `/etc/X11/gdm/*`

- **Methods of starting an X session**
  An X session can be started in 2 ways:
  - **Log in from a virtual terminal** (text based) and then run the script `startx`.
    `startx` in turns starts `xinit`.
    `xinit` configuration file:
    `$HOME/.xinitrc` if found otherwise,
    `/var/X11R6/lib/xinit/xinitrc`
  - **Via an X-Display-Manager (XDM):** The user log-in in is done graphically.
    The display manager is started at boot time (runlevel 5) in the background as a daemon and provides graphical logins to users.
    Note: For this we need to make sure that the /etc/inittab has 5 as default runlevel:
    `id:5:initdefault:`

  - **Popular display managers:**

XDM	Config files Directory	Description
xdm	/etc/X11/xdm/	Provided by XFree86
kdm	***kde_rootdir***/share/config/kdm/	Provided by KDE
gdm	***gnome_rootdir***/gdm/	Provided by Gnome

kde_rootdir = Main root directory for kde desktop system
for kde3: `/etc/opt/kde3`

gnome_rootdir= Main root directory for Gnome desktop system
for Gnome 2: `/etc/opt/gnome`

- **Properties of the Display Managers:**
  kdm is based on xdm and uses many of its configuration files.
  gdm is ta new development and is therefore independant from xdm.

- **xdm configuration:**
  xdm is a typical X11 program and offers only a logo, background and login fields.
  The parameters to change its behaviour are in :
  `/etc/X11/xdm/Xresources`

```
eg.
xlogin*greeting:   Welcome at CLIENTHOST  <---(replaced automatically by $HOSTNAME)
xlogin*namePrompt:      \040\040\040\040\040\040\040Login:
xlogin*fail:            Login incorrect
xlogin*login.greetFont: *-FAMILY-bold-SLANT-normal--*-140-*-*-*-*-iso8859-1
xlogin*login.promptFont:    *-FAMILY-bold-r-normal--*-120-*-*-*-*-iso8859-1
xlogin*login.Font:        *-FAMILY-medium-r-normal--*-120-*-*-*-*-iso8859-1
xlogin*logoFilename       /xxxxx.xpm
xlogin*borderWidth
xlogin*useShape:               true
xlogin*greetColor:             CadetBlue
xlogin*failColor:              red
xlogin*borderWidth:            0
xlogin*frameWidth:             5
xlogin*innerFramesWidth:       2
xlogin*Foreground:             black
xlogin*Background:             #c0c0c0
xlogin*shdColor:               #828282
xlogin*hiColor:                #e0e0e0
```

  xdm runs a script called `/etc/X11/xdm/Xsetup` each time it presents a login window. There we can run programs that change the background etc.
  Some examples of programs: xpmroot , xsetbg etc
  eg. `/usr/sbin/xpmroot /etc/X11/xdm/background.xpm`

- **kdm configuration**
  kdm works quite similar to xdm and uses many of its configuration files in:
  `/etc/X11/xdm/`
  The main kdm configuration file is:
  kde_rootdir/share/config/kdm/**kdmrc**
  kde_rootdir = Main root directory for kde desktop system
  for kde3: /etc/opt/kde3

  The pictures of the users shown in kdm login are(valid formats: .xpm or .png):
  kde_rootdir/share/apps/kdm/pics/users/*username*.png
  The default is default.png

- **gdm configuration**
  gdm has its own configuration files separate from xdm/kdm.
  - Main configuration file:
  gnome_rootdir/gdm/gdm.conf
  gnome_rootdir= Main root directory for Gnome desktop system
  for Gnome 2: /etc/opt/gnome

  - Method of configuring gdm.conf:
  manual (editor) and (much better) through the config program: gdmconfig

  - Other tool for configuring individual user's pictures in gdm login:
  gdmphotosetup

- **XTerminals using the xdm/kdm Display Managers:**

1) Activate the XDMCP (XDM Control Protocol):

- Edit the file `/etc/X11/xdm/xdm-config`

add a '!' at the beginning of the following line (normally the last line):

```
!DisplayManager.requestPort:    0
```

- Edit the file `/etc/opt/kde3/share/config/kdm/kdmrc`:

Enable Xdmcp and restrict the shutdown to only Root

Important note: The `AllowShutdown=Root` is with a BIG 'R'

```
[Xdmcp]
     Enable=true
[X-*-Core]
     AllowShutdown=Root
```

2)Allow access through the network:

- Edit the file `/etc/X11/xdm/xaccess`

----> For direct query from a client:

The client uses a command like: `X -query kdmserver :1`

On server: Enter or activate (remove the '#') the following lines:

```
*                      #Allow any host to remotely login
```
or
```
*.linux.local          #Allow any host from my domain
```
or
```
myhost.linux.local   #allow only myhost to remotely login
```

----> For Broadcast or indirect queries from clients:

The client uses the command:

```
          X -broadcast :2
     or   X -indirect kdmserver :2
```

On server: Enter or activate (remove the '#') the following lines:

```
*               CHOOSER BROADCAST
```
or
```
*.linux.local  CHOOSER BROADCAST
```
or
```
myhost.linux.local CHOOSER BROADCAST
```

----> For Unattended x-login (xdm/kdm actively initiates the contact with the client. The client doesn't have to make a request: He only needs to start his X-Server on the right display port is necessary.

The client uses the command:

```
X :2
```

On server: Edit the file

kde_rootdir`/share/config/kdm/Xservers`

kde_rootdir`=/etc/opt/kde3`

enter the follwoing line:

```
XTerminalName:2 foreign
```

where *XTerminalName*=Client Host name or IPNr.

Note 1: The display port number (`:2` etc) can be chosen at will from the client as long as the same port is not chosen multiple times in the same client host. This number can also be eg. `:2.0` which means the first graphic card used (`0`). Since it's mostly the case we only use eg. `:2` and it's enough.

Note 2: For these configuration files changes to take effect `kdm/xdm` needs to be restarted.

- **`gdm` XDMCP configuration:**
  Use the program gdmconfig ---> Expert sub-menu ---> Activate XDMCP
  or
  - Edit the file ***`gnome_rootdir`***`/gdm/gdm.conf`
    `gnome_rootdir`= Main root directory for Gnome desktop system
    for Gnome 2: `/etc/opt/gnome/`
  - Enable the Xdmcp:

```
[xdmcp]
  Enable=true
```

1.110.4 Install & Customize a Window Manager Environment Weight: 5

Description: Candidate should be able to customize a system-wide desktop environment and/or window manager, to demonstrate an understanding of customization 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:**
  `.xinitrc`
  `.Xdefaults`
  `xhost`
  `DISPLAY` environment variable

- **Window manager:**
  The window managers allow application windows to be moved, resized or iconified. Most display a window title bar, some also display a menu system or allow drag-&- drop between applications.

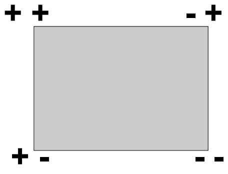
  Window managers names:
  `twm, mwm, olwm, fvwm, kwin, windowmaker` etc:

  Configuration files of window managers:
  Different for each one but most seem to have a `.`*xxxx*`rc` format.
  They are normally in the $HOME directory.
  eg. `.mwmrc, .fvwm2rc, .olwmrc` etc.

- **Configuration of X Clients (X programs):**
  Many X Clients will accept many of the following X11 standard parameters: eg.
  `xterm -T "Title" -fn 9x15 -display :0 -geometry 100x40+30+40`

  - **Positioning and size of window when starting an X Client**: `-geometry`

    Syntax:
    `-geometry` *`<Hsize>`*`x`*`<Vsize><Hpos><Vpos>`*
    `<Hsize>` and *`<Vsize>`* are numbered in characters
    `<Hpos>` `'+'` is right, `'-'` is left (in screen pixels)
    eg. **`+10`** is <u>right</u> 10 pixels
    `<Vpos>` `'+'` is up, `'-'` is down(in screen pixels)
    eg. **`+10`** is <u>down</u> 10 pixels



Examples:
`-geometry 1x1+0+0`
1 char Horiz, 1 char Vert, top left corner
`-geometry 5x20-10+30`
5 char Horiz,
20char Vert,
top right corner
10 Pixels Horiz.to the left
30 Pixels vertical down

Note: The geometry can also be set for individual X clients by editing
`~/.Xresources`
eg. `Xterm*geometry: 90x30`

- **Selecting a font for X Clients:** `-fn` *`fontname`*

  Short list of some fixed-sized fonts:

  `7x14   6x10   6x13    8x13   9x15   10x20`

  eg.

  ```
  xterm -fn 10x20
  ```

  or

  ```
  xterm -fn -misc-fixed-medium-r-normal--20-200-75-75-c-100-iso8859-1
  ```

Note: Fonts can also be set for individual X clients by editing `~/.Xresources`
eg. `Xterm*font: 90x30`

- **`~/.Xresources` or `~/.Xdefaults` file:**

  Note: in SuSE `~/.Xresources` is a symbolic link to `~/.Xdefaults`

  File syntax:

  `ProgramName*Resource`*: *`Value`

  Examples of Xterm settings in `~/.Xresources`

  ```
  xterm*background:       LightYellow2
  xterm*Foreground:       Blue
  xterm.eightBitInput:    true
  xterm*multiScroll:      on
  xterm*jumpScroll:       on
  xterm*font:   -adobe-courier-bold-r-normal--14-140-75-75-m-90-iso8859-1
  xterm*ScrollBar: on
  xterm*SaveLines: 2000
  xterm*VisualBell: true
  xterm.eightBitOutput:   true
  Xterm*geometry: 90x30
  ```

  Actualizing changes made in the `~/.Xresources` file
  without restarting the X Server:

  ```
  xrdb -merge .Xresources
  ```

- **Starting an X session with `startx`:**

  Log in from a virtual terminal (text based) and then run the script `startx`.
  `startx` in turns starts `xinit`.
  `xinit` startx the X Server then starts the `xinitrc` script file:
  (`$HOME/.xinitrc` if found otherwise `/var/X11R6/lib/xinit/xinitrc`)

  Content of `xinitrc` script:
  - System wide configured key definitions are loaded:
    Definitions are in:
    `/etc/X11/Xmodmap` and `~/.Xmodmap`
  - System wide configured Resources definitions are loaded:
    Definitions are in:
    `/etc/X11/Xresources, ~/.Xresources, ~/.Xfefaults`
  - A user's manually entered programs may start here
  - The selected window manager is started.

- **Starting an X session with xdm/kdm/gdm:**
  When a user does login via a display manager, a similar process to `startx` will occur, the difference is the script that will be run is:
  `/etc/X11/xdm/Xsession` and `~/.Xsession` if it exists.

  Note: Some distributions are running the `~/.xinitrc` from `Xsession` to keep the same environment consistent.

- **X11 in Network**
  Xclient preparation:
  Since almost all X Client programs can us the argument `-display :xx`, we can start a client program and send its output to any existing X server that will allow the connection.
  All X Client programs need to know where the X Server intended to host the program is located. This information is given to the program when we start it either via the above argument (-display :xx.xx) or via the environment variable `DISPLAY`.
  To do so the following command prepares the content of this variable:
  ```
     export DISPLAY=X-ServerHost:xx
  ```
  then run the X client program.

  Xserver preparation:
  X-Servers by default allow only the local user's programs to be displayed. For other users or hosts to be allowed to display their X-Client programs on it, the X-Server needs to be told to do so. The notification is done with command `xhost`.
  Syntax:
  ```
     xhost [+|-] ClientHostName
  ```
  eg. `xhost + localhost` Allows other users X-clients on the local host to use this X-server.
  `xhost +` Allows everybody from anywhere to use this X-Server. Hmmm dangerous!!!
  `xhost + myfriend` Allows XClient programs on the host `myfriend` to use this local X-Server.
  `xhost - bugger` Disallows host “bugger” to use this X-Server.

  Note 1: Only the owner of the X-Server process is allowed to issue the `xhost` command.

  Permanent allowance of access:
  There are 2 regular methods to permanently a list of hosts access to the local X-Server:
  1) Edit the file enter the comand xhost for all the hosts allowed in `~/.xinitrc` script or
  2) Create a file called `/etc/X`*`n`*`.hosts` and enter all the hosts allowed to use the local X-Server. *n*=X-Server display port number.

- **Checking the libraries dependencies for X-Client programs:**
  In the matter of library dependencies, there is no difference between normal programs and X-Client programs. The program `ldd` does the job. See chapter 1.102.4