LPI Level 1

Test 101
Certification Study Guide

By Victor Mendonça

Detailed Objectives:
http://www.lpi.org/eng/certification/the_lpic_program/lpic_1/exam_101_detailed_objectives

October 22nd 2007 (Updated on January 2010)

License

Bibliography

- O'Reilly - LPI Linux Certification in a Nutshell, 2nd Edition
http://oreilly.com/catalog/9780596005283/index.html
- IBM - Linux Professional Institute (LPI) exam prep
www.ibm.com/developerworks/linux/lpi/

Study Material

- MCMCSE Forum
http://www.mcmcse.com/linux/lpic1.shtml
- Practice Exam
http://www.linux-praxis.de/lpisim/lpi.html
- IRC
irc.freenode.net #lpi

Errata

I'm willing to make major corrections, however I cannot guarantee the response time.

Please send an email to victorbrca@yahoo.ca with the following info:
Subject: LPI 101 correction
Include:
- Content to be corrected
- Page
- Source that proves you are right

Thanks!! ;)

Author

Victor Mendonça
http://wazem.org

Index

Topic 1.101 - Hardware and Architecture

Objective 1 - Configure Fundamental BIOS Settings ..... Page 4
Objective 3 - Configure Modems and Sound Cards ..... Page 6
Objective 4 - Set Up Non-IDE Devices ..... Page 6
Objective 5 - Set Up Different PC Expansion Cards ..... Page 8
Objective 6 - Configure Communications Devices ..... Page 9
Objective 7 - Configure USB Devices ..... Page 9

Topic 1.102 - Linux Installation and Package Management

Objective 1 - Design a Hard Disk Layout ..... Page 11
Objective 2 - Install a Boot Manager ..... Page 12
Objective 3 - Make and Install Programs from Source ..... Page 16
Objective 4 - Manage Shared Libraries ..... Page 17
Objective 5 - Use Debian Package Manager ..... Page 19
Objective 6 - Use Red Hat Package Manager ..... Page 22

Topic 1.103 - GNU and UNIX Commands

Objective 1 - Work on the Command Line ..... Page 25
Objective 2 - Process Text Streams Using Filters ..... Page 31
Objective 3 - Perform Basic File Management..... Page 43
Objective 4 - Use Unix Streams, Pipes and Redirects ..... Page 48
Objective 5 - Create, Monitor and Kill Processes ..... Page 50
Objective 6 - Modify Process Execution Priority ..... Page 57
Objective 7 - Search Text Files Using Regular Expressions ..... Page 58
Objective 8 - Perform Basic File Editing Operations Using vi ..... Page 63

Topic 1.104 - Devices, Linux Filesystems, and the Filesystem Hierarchy

Objective 1 - Create Partitions and Filesystems ..... Page 66
Objective 2 - Maintain the Integrity of Filesystems ..... Page 70
Objective 3 - Control Filesystem Mounting and Unmounting ..... Page 74
Objective 4 - Managing Disk Quotas ..... Page 78
Objective 5 - Use File Permissions to Control Access to Files ..... Page 84
Objective 6 - Manage File Ownership ..... Page 87
Objective 7 - Create and Change Hard and Symbolic Links ..... Page 89
Objective 8 - Find System Files and Place Files in the Correct Location ..... Page 91

Topic 1.101

Hardware and Architecture

Objective 1: Configure Fundamental BIOS Settings

BIOS

- Usually includes system initialization, testing memory (and other) and locating OS.

Date and time

- Configured in the BIOS and passed on to the OS.
- It can also be configured on the OS (eg: NTP server or daemon)

Disks and boot devices

- Booting choices, NIC booting, configuring devices, etc...

Buses

- PCI - Peripheral Component Interconnect (8 and 16-bit devices)
- ISA - Industry Standard Architecture (32-bit devices)
- The files /proc/pci contains informations on current system PCI devices. This file is (has) becoming obsolete and being replaced by the command 'lspci' (/sbin/lspci on RedHat, /usr/bin/lspci on Debian)

Resource Assignments

- Resource assignments can be identified via boot messages (dmesg), the specific /proc/ subsystem or other utilities

DMA (Direct Memory Access)

- Provides hardware direct access to the memory bypassing CPU
- Most devices only request a DMA channel when IO is actually happening
- See /proc/dma for current system DMA assignments (usually empty, see previous statement)

I/O

- Devices addresses on CPUs memory map
- The file /proc/ioports contain information on current system I/O assignments

IRQ (Interrupt Requests)

- Interrupt priority to the CPU
- See /proc/interrupts for current system IRQ assignments
- Today's devices share IRQs

Note: memorize parallel and serial IRQs

I/O, IRQ and DMA

Device	I/O	IRQ	DMA
ttyS0 (COM1)	3f8	4	N/A
ttyS1 (COM2)	2f8	3	N/A
ttyS2 (COM3)	3e8	4	N/A
ttyS3 (COM4)	2e8	3	N/A
lp0 (LPT1)	378-37f	7	N/A
lp1 (LPT2)	278-27f	5	N/A
fd0,fd1	3f0-3f7	6	2

Plug And Play

- Was developed to allow a a device to tell the system it's resource requirements and allow the BIOS to tell the device which resources to use
- Prior to kernel 2.4 a package called isapnptools allowed users to configure PnP
  . The command pnpdump scans PnP devices during boot and dumps a list of resources that devices need or would like to use in a configuration file (usually /etc/isapnp.conf)
  . User would edit the file and uncomment the commands the he would like to use
  . isapnp would read the conf file and configure the system
- A tool called lspnp can also be used to display information on PnP devices (which is the same as /proc/bus/pnp)

IDE Hard Drives

- Integrated Drive Electronics are the most common nowadays
- Usually come in two formats:
  . AT Attachment - ATA
  . Serial AT Attachment - SATA
- Disk capacities are measured in powers of 10 (not 1024)

Size Limitation

- Original CHS (Cylinder, Head, Sector) design only allows up to 137 GB
- Logical Block Address (LBA) was designed to overcome the size limits from CHS. The system ignores the geometry and leaves it to the drive to figure out by referring to a LBA instead of a real location
- An older device that does not have LBA support might need the boot from within the the first 1024 cylinders

1024-Cylinder Limit

- Boot loader on Linux can be placed either on MBR or on the root partition. Some BIOS can not read over the 1024th cylinder, which can cause a problem.
- Older version of LILO requires the Kernel to be within the first 1024 cylinders as well

Linux Disk Names

- The /dev/ filesystem is a pseudo filesystem (just like /proc/)
- Naming is done as the devices are found:
  . Controller 1 - Master HD - /dev/hda
  . Controller 1 - Slave HD - /dev/hdb
  . Controller 2 - Master HD - /dev/hdc
  . Controller 2 - Slave HD - /dev/hdd
- A drive can have up to 4 primary partitions and up to 512 logical (however the OS might limit it to 63)
- Nowadays USB and SATA devices appear as sd rather than hd (also true for hard drives)

Legacy Peripherals

- Many systems do not use interrupts for printers, so the IRQ may or may not be used
- Parallel ports can usualy be configured with different signaling modes (bi-directions, EPP, ECP):
  . Enhanced Capabilities Port (ECP) - Designed to use with printers
  . Enhanced Parallel Port (EPP) - Devices that require a large amount of data to flow either way (CD-ROM, tape)

Objective 3: Configure Modems and Sound Cards

Modems

- Also have digital compression and error correction capabilities

Modem Types

- Modems discussed in the exam are asynchronous modems, where data is transmitted in individual bytes (synchronous modems transmit data in blocks of data, like the IP protocol)
- Winmodems are no compatible (unless a Linux-Specific driver is available)
- Linmodems are software assisted modems that work under Linux (linmodems.org)

Modern Hardware Resources

- External modems usually connect to port RS-232 (ttyS0 and S1)
- PCI modems usually show as ttyS2
- The program 'setserial' can be used to view modem parameters that will be used with the serial driver (it does not probe the hardware, it only tells the serial driver what parameters to use)
- Some modems also require flow control, which sends signal from each end advising of the device transmission state (CTS - Clear to send and RTS - Ready to send)
- Be careful when configuring a modem manually due to IRQ and I/O conflicts

Sound Devices

- Devices can come with different types of connection interface (even USB)
- I/O ports can vary between 0220-022f, 240, 260 or 280
- IRQs vary from 2, 5, 7, 10 (5 is usually the default)

Support and Configuration

- Kernels 2.4 and 2.6 have support for a variety of drivers built-in as modules
- On 2.4 module configuration is stored under /etc/modules.conf, while 2.6 keeps it under /etc/modprobe.conf
- Drivers:
  . 2.4 and older kernels - Open Sound System (OSS)
  . 2.6 - Advanced Linux Sound Architecture (ALSA)
- For kernels with OSS support a tool called 'sndconfig' was used. The utility probes the sound card, plays a sound and updates /etc/modules.conf

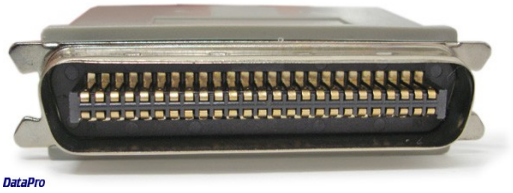
Objective 4: Set Up Non-IDE Devices

Overview

- Comparison between SCSI and IDE
  . More expensive (2-3x more), more expandability, flexibility and throughput

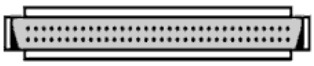
SCSI Types

- SCSI-1 - Original SCSI, 8-bit, 5Mbps Centronics 50-pin connector



- SCSI-2 - 8-bit, 5Mbps, Micro-D 50-pin connector. Still used for low range tape drives. Interchangeable with SCSI-1

- Wide SCSI - 16-bit, 10Mbps, Micro-D 68-pin connector



- Fast SCSI - 8-bit, 10Mpbs, Micro-D 50-pin connector
- Fast Wide SCSI - 16-bit, 20Mbps, Micro-D 68-pin connector
- Ultra SCSI - 8-bit, 20Mbps, Micro-D 50-pin connector
- Ultra Wide SCSI (SCSI-3) - 16-bit, 40Mbps

- Ultra2 - 8-bit, 40Mbps
- Wide Ultra2 - 16-bit, 80Mbps

SCSI IDs

- Can be configured via jumpers, switches or software
- Devices using Single Connector Attachment (SCA) gets their ID assigned automatically as they are hot-plugged
- Priority goes from 7-0 15-8 (high to low), and slower devices usually get higher priority
- The amount of available ids is based on a binary reading of address line
  . 8-bit SCSIs have 3 address lines which result in 8 devices (0-7), 7 usually being for the controller (2expn3=8)
  . 16-bit SCSIs have 4 address lines resulting in 16 devices (0-15), 7 usually being for the controller (2expn4=16)
- The full ID of a device consists of an adapter number, channel number, device ID and a LUN

SCSI Logical Unit Numbers

- LUNs (logical unit numbers) are used by some SCSIs to provide the controller with an address to a logical partition
- Tapes and disk drives either do not report a LUN or use LUN 0

Linux SCSI Disk Device Files

- Partition numbers have nothing to do with SCSI ID, but is instead related to SCSI ID/LUN = device
  . IDE - /dev/hda (disk1), /dev/hdb (disk2), /dev/hda1 (first partition on disk 1)
  . SCSI - /dev/sda (disk1), /dev/sdb (disk2), /dev/sda1 (first partition on disk 1)
- It's important to know that numbering is redone at each reboot. A install or removal of a device could cause major system problems if fstab refer to device names instead of label or UUID

Support And Configuration

- File /proc/scsi/scsi will provide information on scsi devices
- 'scsi_info' will display information on which real device corresponds to system device (/dev/sdx*n*). Some systems do not include this utility
- Newer systems have moved over to the SCSI Generic driver (sg driver). Information can be collected from /proc/scsi/sg, and a

few utilities are also provided:
  . sg_map - Provides a map between the sg name and another device
  . sginfo - Same as scsi_info

Termination
- SCSI buses must have a terminator on each end. They usually are the controller, an external terminator or a disk internal terminator
- Special attention is needed when terminating a bus with 8-bit and 16-bit devices.

SCSI BIOS And BOOT Sequence
- SCSI controller's have their own firmware and a BIOS which display a boot menu option after the main BIOS has loaded. Sometimes they have to be configured:
  . Controller SCSI address – Usually 7, however this can be changed
  . Default boot device – Older card require this to be 0, new devices allows you to choose any device
  . On board termination – Internal, external or both
  . SCSI bus speed - For back compatibility with older devices

Objective 5: Set Up Different PC Expansion Cards

Plug And Play
- Jumper era - Settings were made through configuration of jumpers
- Nonvolatile era - Settings were stored into a nonvolatile memory space. Software was proprietary and usually based on MS-DOS, which required Linux users to have a MS-DOS machine to provide initial configuration
- Modern era - PCI bus automatically configures devices prior to OS loading.

Using The /proc Filesystem
- The /proc filesystem can be used to find many hardware information. The only problem is that it only shows info on devices that have their device/drivers on open/active
- It's important to know how to use the /proc filesystem and the basic I/O, IRQ and DMA

Interrupts
```
$ cat /proc/interrupts
           CPU0
  0:      53128   IO-APIC-edge      timer
  1:       1838   IO-APIC-edge      i8042
  8:          3   IO-APIC-edge      rtc
  9:          4   IO-APIC-fasteoi   acpi
 12:        675   IO-APIC-edge      i8042
 14:      15558   IO-APIC-edge      libata
 15:          0   IO-APIC-edge      libata
 16:      95186   IO-APIC-fasteoi   uhci_hcd:usb1, ehci_hcd:usb5, HDA Intel,
i915@pci:0000:00:02.0
 17:          0   IO-APIC-fasteoi   uhci_hcd:usb4
 18:      15300   IO-APIC-fasteoi   uhci_hcd:usb2, ipw2200
 19:          0   IO-APIC-fasteoi   uhci_hcd:usb3, eth0
NMI:          0
LOC:      23704
ERR:          0
MIS:          0
```

DMA
```
$ cat /proc/dma
 4: cascade
```

I/O

```
$ cat /proc/ioports
0000-001f : dma1
0020-0021 : pic1
0040-0043 : timer0
0050-0053 : timer1
0060-006f : keyboard
0070-0077 : rtc
0080-008f : dma page reg
00a0-00a1 : pic2
00c0-00df : dma2
00f0-00ff : fpu
0170-0177 : 0000:00:1f.1
  0170-0177 : libata
01f0-01f7 : 0000:00:1f.1
  01f0-01f7 : libata
0376-0376 : 0000:00:1f.1
```

Objective 6: Configure Communication Devices

- Be familiar with the /proc/ filesystem
- Check to make sure that the required drivers are part of your distro, if not install it.
- For some modem connection protocols (like ppp) a synchronous driver might be needed as the default it for asynchronous (for bit/character mode transmission instead of block)

Objective 7: Configuring USB Devices

Overview

- Each computer system may have on or more controllers/hubs
- Each hub supports up to 7 direct devices and a total of up to 127 chained devices.
- The Universal Serial Bus System is a layered system
  . Bus Interface - Provides the hardware connection (signaling, data transfer)
  . Device Layer - Allows the host to identify characteristics of the device (manufacturer, model, speed, etc...)
  . Function layer - Provides device specific functions (storage, sound, video, etc...)
- USB speeds
  . USB 1.1 - 12Mbps
  . USB 2.0 - Theoretical max of 480Mbps

USB Devices

- There are several classes:
  . Human Interface Device - Input devices, like mice, keyboard, etc...
  . Communications Device - Modems
  . Mass Storage Devices - Disk devices
  . Audio - Sound devices
  . IrDA - Infrared devices
  . Printer - Printers and USB-to-parallel cables

Controllers, Drives And Manipulation

USB Controllers

- There are tree types of USB controllers:
  . Open Host Controller Interface (OHCI) - USB 1.1 - Drivers: usb-ohci.o
  . Universal Host Controller Interface (UHCI) - USB 1.1 - Drivers: usb-uhci.o, uhci.o
  . Enhanced Host Controller Interface (EHCI) - USB 2.0 - Drivers: ehci-hcd.o

USB Drivers

- USB support was added on Kernel 2.3.x and back ported to 2.2.x (minus support to USB mass storage devices due to SCSI changes on 2.3.x). The back port was also included in the 2.2.18. Kernel 2.0.x had no USB support.
- Support is provided via kernel modules due to the hot-pluggable nature of USB
- USB 1.1 driver depends on the controller's chipset, while 2.2 drivers depend on the 1.1 driver and an EHCI driver
- Linux kernel USB drivers are divided into three categories:
  . Host Controller Drivers - usb-ohci.o (OHCI driver), usb-uhci.o (UHCI driver), uhci.o (old alternate UHCI driver) and ehci-hcd.o (EHCI driver)
  . Class Drivers - hid.o, usb-storage.o (mass storage), acm.o (Automated Control Mode, deals with modems that emulate standard serial modem AT command interface), printer.o and audio.o
  . Other Device Drivers - These devices don't fit with the standard USB classes or don't work with the standard class drivers. Eg: rio500.o (Rio mp3 payer) and pwc.o (Phillips webcams)

USB Hotplug

- There are usually two commands that may handle hot-plugging of the devices:
  . usbmgr - Config file is at /etc/usbmgr
  . hotplug - Config file is at /etc/hotplug
- Newer systems will most likely have hotplug

Displaying USB Information

- Information can be acquired from /proc/bus/usb/devices
- USB devices are attached to a host in a tree trough some number of host devices. The command “lsusb -t” shows how devices are physically attached

```
$ lsusb -t
Bus#  2
`-Dev#   1 Vendor 0x0000 Product 0x0000
  `-Dev#   2 Vendor 0x05f3 Product 0x0081
    |-Dev#   3 Vendor 0x05f3 Product 0x0007
    |-Dev#   4 Vendor 0x05bc Product 0x0102
    `-Dev#   5 Vendor 0x045e Product 0x0039
Bus#  1
`-Dev#   1 Vendor 0x0000 Product 0x0000
  |-Dev#   3 Vendor 0x0409 Product 0x0058
  | `-Dev#   5 Vendor 0x0fe9 Product 0x9010
  `-Dev#   4 Vendor 0x07cc Product 0x0501
```

Topic 1.102

Hardware and Architecture

Objective 1: Design a Hard Disk Layout

Partitions

- IDE drives are limited to 63 partitions, and SCSI to 15
- A partitioning program may report an error if it has a different understanding of the disk geometry than the partitioning program that created the partition
- The nominal disk geometry can be displayed from /proc/ide/hda/geometry
- The partition table is located on the MBR of a disk. Due to its small size (MBR) primary partitions are limited to a number of 4
- Logical partitions start from 5 (/dev/sda5)

System Considerations

Limited disk space

- Knowing how to divide the HD with system partitions is a must. For example, dividing a 1GB HD:
  . /boot - 50MB, ensures that he kernel will be within the first 1024 cylinders
  . / - 850MB, holds everything that is not in /boot
  . swap - 100MB
- On this case, the boot partition could be combined with root as long as the entire partition fits the 1024-cylinder.

Larger systems

- On larger platforms, different functionality requirements will dictate the layout of the system. For example, a 100GB server that will provide executable data files to users via NFS:
  . /boot - 100MB, keeps kernel within the 1024-cylinder
  . swap - 1GB
  . / - 500MB minimum
  . /usr - 4GB, all executables are shared to workstations via read-only NFS
  . /var - 2GB, keeping log files in their own partition helps making sure that their size will not affect the system
  . /tmp - 500MB, same as /var
  . /home - 90GB, used to provide the /home directory to all the users

System Role

- A server would keep all executable files for the workstations and share via NFS. This was very used when workstation storage was an issue. It also helped distributing updates

Backup

- Different partitions were backed up individually into also individuals backup devices. This also can sometimes dictate the size of the partition

Mounting

- When mounting filesystem on a exiting non-empty folder, any files or subdirectories within the original mounted folder no longer are shown until filesystem is unmounted

Swap Space

- Usually is kept to twice the size of RAM, but nowadays has changed a lot due to cheap HD and memory price
- When in doubt use twice the size of RAM
- A swap file can also be used, however a dedicated partition performs better

General Guidelines

- Keep “/” simple by dividing large portions of the directory to other partitions
- Separate a small /boot partition for boot loader kernel
- Separate /var and make sure it's big enough for you rotation schema
- Separate /tmp. It should be large enough to handle file for all users simultaneously
- Separate /usr and make it big enough for kernel building. It can also be used for sharing via NFS
- Separate /home for machines with multiple users or any machines that you don't want these files affected during system upgrades. For better performance put home on a disk array
- Set swap to at least same size as RAM (preferably double)

Objective 2: Install a Boot Manager

- Consists of two processes:
  . Run the boot loader from the boot device - It's LILO's job to find the kernel and load it into memory
  . Launch the Linux kernel and start the process - Kernel takes over and controls the hardware

Overview

Common DOS Boot Process

1- PC boots and loads the MBR
2- The MBR (which is a space of 512 bytes that holds the partition table as well as the stage 1 boot loader) checks table to find the primary partition marked a active and loads the first sector of that partition. Note that the code on the MBR is also the same code that can display the message “file system not found”
3- The partition boot record (also a stage 1 loader) loads a set of block from the partition, which is the stage 2 loader

Linux

- Linux provides LILO and GRUB, which allow the booting of multiple operating systems
- They can both be installed into MBR, partition boot record or even removable devices
- They differ and many aspects, but the one with the most highlight is that LILO needs to be re-run whenever there's a kernel upgrade or certain changes to the system, while GRUB only needs a configuration file to be changed

LILO

- Most popular boot loader utility. Consists of two parts:
  . The boot loader - A two-stage program. The fist part is a small code located in the MBR of the disk. It's job is to launch the second stage, that presents user with boot options (kernel image selection and boot time) and loads the kernel into memory
  . The LILO command - Also called the map installer. Used to install and configure LILO. It also reads a configuration file that tells where to find the kernel images, video information, default boot disk, and so on. It uses the information and write in files for use of the boot loader
- The primary function of the 'lilo' command is to write a stage 1 boot loader and create a map file (/boot/map) using configuration information that is available in the LILO's config file (/etc/lilo.conf)
- The boot loader and any information associated must be installed by LILO map installer utility. The LILO command writes the portion of LILO that resides on MBR specifically for your system. You have to do it manually when you compile a new kernel
- It's advisable to create a config file to install the boot loader on a floppy first. Once that has been created you can change the “boot” parameter on the config file to reflect your bootable hard disk

The Boot Loader Prompt

- When LILO launches, press Tab for an option of images for LILO to boot. You can also pass on parameters to the kernel

```
LILO: <press Tab>
linux* linux_586-smpexperimental
```

The LILO map installer

- Needs to be re-run every time the configuration file is changed, when a partition is changed or any other system change that may invalidate information in the configuration file

```
Name
       lilo

Syntax
       lilo [options]

Description
       lilo installs a boot loader that will be activated next time you boot. It has lots of
options.

       -C config_file
              Read the config_file instead of the default /etc/lilo.conf
       -m map_file
              Write map_file in place of the default as specified on the config file
       -q
              Query the surrent configuration
       -v
              Increase verbosity
```

. -q - Will display information from the MAP file
. -R - Will reboot the using the specified file system (only for the next immediate reboot)
. -l - Displays information about the path o kernel
. -u - Uninstalls LILO and restores the previous boot record

LILO Configuration File

- LILO's configuration file contains a series of options and kernel image information. Some options are global and other affect only the listed kernel image

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
read-only
message=/boot/message
lba32
default=linux

image=/boot/vmlinuz-2.4.0-0.43.6
        label=linux
        initrd=/boot/initrd-2.4.0-0.43.6.img
        root=/dev/hda5

other=/dev/hda1
         label=dos
```

. boot - Name of the HD partition that contains the boot sector
. timeout - Time in tenths of a second to wait for user input
. prompt - Set boat loader to prompt the user. It can also be done by pressing Shif, Ctrl or Alt when LILO starts
. read-only - Sets the root filesystem to read-only. The system startup utility will set to read-write after
. map - Sets location to the map file. Default is /boot/map
. install - Sets the file to install as the new boot sector. Default is /boot/boot.b
. image - The kernel image location
. label - A label for the kernel
. root - Sets the device to mounted as root
. compact - Attempts to read requests for adjacent sectors. This speeds up load time and keeps the map smaller
. message - Displays a message before boot prompt. It must be less than 65535 bytes
. lba32 - Sets LILO to use LBA32 instead of CHS

. password - Specifies a password that must be entered before boot. This is a clear text password stored in /etc/lilo.conf, so file permission should be watched for
. restricted - Configures password to be required only for special boot parameters (like single user)
. loader - Specifies the loader to be run. LILO allows the feature chain.b, which call a loader from another partition

LILO Locations
- LILO can be placed either on the boot sector of the disk or in the root partition, but should be placed on the boot sector when using multiple OSs

GRUB
- Multistage boot loader more flexible than LILO
- Supports serial console and much more
- Is also a small but powerful shell
- Configuration files is stored in /boot/grub/grub/menu.lst. On some systems this is a sym link to /boot/grub/grub.conf

```
# Red Hat
[root@centos ~]# ll /boot/grub/menu*
lrwxrwxrwx 1 root root 11 Jan 31 18:21 /boot/grub/menu.lst -> ./grub.conf
# Debian
victux ~ $ ll /boot/grub/menu.lst
-rw-r--r-- 1 root root 4.2K 2009-02-01 02:08 /boot/grub/menu.lst
```

Grub Device Naming
- Uses the following naming system, [xdn,m]
. x - Type of device (HD, floppy)
. d - Stands for drive
. n - Device number (as seen by the BIOS. Starts at 0)
. m - Partition number on the device (starts at 0)
- GRUB does not distinguish IDE and SCSI disks, it refers to the devices as seem by the BIOS

Installing GRUB
- When using grub device naming, it's important to use quotes to avoid shell interpretation
- There are two ways to install Grub.
. Run the command bellow. It will check /boot/grub/device.map to find the proper mapping from BIOS drives to Linux devices:

```
$ cat /boot/grub/device.map
(hd0)  /dev/sda
# for Linux devices
$ grub-install /dev/sda
# for BIOS drives
$ grub-install '(hd0)'
```

. Using grub command

```
$ grub
grub> root (hd0,0)
grub> setup (hd0)
```

- Use the command 'grub-install '(fd0)" to regenerate a device.map file after system changes have been made

Booting Grub
- In case there's no configuration file, or the configuration file did no specify a kernel to load, grub will display a prompt “grub>” waiting for the following information:
. 1- Root device
. 2- Kernel filename [options]
. 3- initrd filename (optional)
. 4- boot
- Here's an example of a HedHat with boot on device hda1 and hda2

```
grub> root (hd0,0)
grub> kernel /vmlinuz-2.4.18-14 ro root=/dev/hda2
grub> initrd /initrd-2.4.18-14.img
grub> boot
```

The Grub configuration file

- Grub also has a graphical menu that can bypass the previous steps. The same could've been achieved by editing /boot/grub/menu.lst:

```
default=0
timeout=3
title Red Hat Linux (2.4.18-14)
        root (hd0,0)
        kernel /vmlinuz-2.4.18-14 ro root=/dev/hda2
        initrd /initrd-2.4.18-14.img
```

. default - Entry number to load. “0” loads the first entry, “1” loads the second, and so on
. savedefault - When used with a system choice will make that system the default
. password - Password to be provided for user to edit boot parameters. This can be in a clear text or as MD5 digest and can be assigned to different entries so (so each has a different password)
. lock - Locks an entry from being edited at boot
. rootnoverify - Grub will not attempt to mount partitions or check filesystems (useful for mounting non Linux systems like NTFS)
. chainloader - Specifies that another file will be loaded as stage 1 file

Recovery

- If you don't a have a recovery floppy, a normal system recover disk (like a live CD) can also be used. The chroot command can the be used to mouth the filesystem as '/'
- A good of avoiding problems is to set /boot/ on a separate partition that will hardly change. This is also necessary if your system uses a filesystem that is not supported by the boot loader (like LVM)
- Multiple systems must not share the same /boot/ partition

Building A Boot Diskette

- A floppy doesn't have much space or idea of cylinders, so to install stages 1 and 2 of GRUB on it a user needs to install stage 1 and copy stage 2 into the immediate sectors
- The original stage 1 and 2 files can found in /usr/share/grub/
- Here's how to create a boot disk without a configuration file (this will remove the filesystem on the disk)

```
$ cd /usr/share/grub/
dd if=stage1 of=/dev/fd0 bs=512 count=1
dd if=stage2 of=/dev/fd0 bs=512 seek=1
```

- With this device you can now boot grub in any system (even non-Linux)
- A second option is to install grub in a disk as well as a configuration file

```
mkdosfs -R 210 /dev/fd0
mount /dev/fd0 /mnt/floppy
mkdir -p /mnt/floppy/boot/grub
cd /boot/grub/
cp stage1 stage2 grub.conf /mnt/floppy/boot/grub/
umount /dev/fd0
grub
grub> root (fd0)
grub> setup (fd0)
```

Objective 3: Make and Install Programs from Source

- When not binary, the package must be compiled

Getting Open Source And Free Software

- Source code is available in many forms and for many OSs

Tape Archive (tar)

- Tape ARchive (tar) is used to manipulate archives from the files in a directory tree. The entire code file three is stored in one file
- The tar command does not compress any data, it only stores in a form that all file properties can be restored

Tarball definition

- It's a compressed tar
- Compression programs can be compress, gzip or bzip2
- Tarball files usually contain the source code, a makefile and some documentation
- Common extensions are .tar.gz, .tgz, .bz2
- Gunzip will handle .Z, .tar.gz and .tgz

Opening a tarball

- Involves two processes, uncompress it with gunzip and then extract it with tar
- There different ways of doing it:

```
#1
$ gzip -d tarball.tar.gz
$ tar xvf tarball.tar
#2
$ gunzip tarball.tar.gz
$ tar xvf tarball.tar
#3 - Does not decompress file (only stdout)
gzip -dc tarball.tar.gz | tar xv
#4 - Does not decompress file (only stdout)
tar zxvf tarball.tar.gz
#5 - Does not decompress file (only stdout)
bzip2 -dc tarball.tar.bz2 | tar xv
#6
tar jxvf tarball.tar.bz2
```

- The option '-c' is what redirects to stdout (same as zcat)

Other Compressing Tools

- Zip files - Can be associated to the 'unzip' tool
- CVS trees - Concurrent Version System

Compiling Open Source Software

- After unpacking the source code, it's necessary to compile it. The system must have the appropriate tool to do so (like gcc and make)

Configure

- Configure is made by the programmer using the autoconf utility
- Sits on the top folder of the tar file
- Intended to create a makefile that it's customized to the system
- When executed it checks for a compiler, libs, utilities and other. If something is missing it will present user with a detailed error message. If everything is ok it will create a custom Makefile for the software package base on your particular system
- Some source files do not include a configure script. These packages usually have a makefile that will work on most systems
- Configure scripts usually also have a '--prefix' option to allow the install into another location other than the default

Config.cache
- First tieme configure is run it creates a 'config.cache' file within the same directory. It's recommended to remove this file before re-running the configure file

Make And Makefiles
- Called makefile because the program that runs it is called Make
- Contains rules that tells the program how to build things
- Defines targets and their dependencies as specified on the Makefile
- This is when the code is compiled
- Can also generate errors that need to addressed before installation

Installing the compiled software - make install
- “Make install” points to a default location where the software should be installed (may differ according to distro)
- User needs to be root
- Makes sure that installed files have the proper ownership and permission

Objective 4: Manage Shared Libraries

Types

Statically Linked
- A program that contains executable code from the required libraries for it to run. It has few pros, but it can become very heavy and take a lot off the memory and multiple of these programs are running at the same time

Dynamically Linked
- Does not have the library code built-in. It links to the existing system libraries at run time. Runs faster and the executables are smaller. Dynamic linked libraries are shared between many apps, and are called shared libraries

Note: Some programs may have two instances, static and dynamic, in case of system failure

```
[root@centos ~]# ll /bin/ln
-rwxr-xr-x 1 root root 29872 May 24  2008 /bin/ln
[root@centos ~]# ll /sbin/sln
-rwxr-xr-x 1 root root 535118 May 23  2008 /sbin/sln
```

Shared Library Dependencies
- All programs that are dynamic linked require at least a few shared libraries. If they are no found, the program will fail to run
- To check what libs are necessary for a particular executable, use the “ldd” tool:

```
$ man ldd

LDD(1)

NAME
       ldd - print shared library dependencies

SYNOPSIS
       ldd [OPTION]...  FILE...

DESCRIPTION
       ldd  prints  the  shared libraries required by each program or shared library specified
on the
       command line.
```

Required libraries for bash

```
$ ldd /bin/bash
        linux-gate.so.1 =>  (0xffffe000)
        libncurses.so.5 => /lib/libncurses.so.5 (0xb7ea0000)
        libdl.so.2 => /lib/tls/i686/cmov/libdl.so.2 (0xb7e9c000)
        libc.so.6 => /lib/tls/i686/cmov/libc.so.6 (0xb7d51000)
        /lib/ld-linux.so.2 (0xb7ef2000)
```

Reference to the previous ln/sln example:

```
[root@centos ~]# ldd `which ln` `which sln`
/bin/ln:
        linux-gate.so.1 =>  (0x00bfa000)
        libc.so.6 => /lib/libc.so.6 (0x00889000)
        /lib/ld-linux.so.2 (0x00866000)
/sbin/sln:
         not a dynamic executable
```

Linking Shared Libraries

- The ld.so/ld-linux.so (shared object dynamic linker) is what examines the dynamic linked executables at the runtime. It checks for dependencies in the exec file header (Executable and Linking Format - ELF) and attempts to satisfy any unresolved link
- The necessary shared libraries needed by the program are searched for in the following order:
  . LD_LIBRARY_PATH - Path with a list of folders tat may contain libraries
  . /etc/ld.so.cache - A binary file that ld.so can read quickly
  . In the default path /lib and /usr/lib
- ldconfig - When is run it updates /etc/ld.so.cache with a list of libraries found in /etc/ld.so.conf, in the default directories (/lib and /usr/lib) as well as the directories given with the command.

man ld.so

```
$ man ld.so

LD.SO(8)

NAME
       ld.so/ld-linux.so - dynamic linker/loader

DESCRIPTION
       ld.so  loads  the  shared libraries needed by a program, prepares the program to run,
and then runs it.  Unless explicitly specified via the -static option to  ld  during
compilation,  all
       Linux programs are incomplete and require further linking at run time.

       The necessary shared libraries needed by the program are searched for in the following
order

       o      Using  the  environment  variable  LD_LIBRARY_PATH (LD_AOUT_LIBRARY_PATH for
a.out pro-
              grams).  Except if the executable is a  setuid/setgid  binary,  in  which  case
it  is
              ignored.

       o      From  the  cache  file  /etc/ld.so.cache  which  contains  a compiled list of
candidate
              libraries previously found in the augmented library path.

       o      In the default path /usr/lib, and then /lib.
```

man ldconfig

```
$ man ldconfig

ldconfig(8)
```

```
NAME
       ldconfig - configure dynamic linker run-time bindings

SYNOPSIS
       ldconfig [OPTION...]

DESCRIPTION
       ldconfig  creates, updates, and removes the necessary links and cache (for use by the
run-time  linker, ld.so) to the most recent shared libraries found in the directories specified
on  the  command line, in the file /etc/ld.so.conf, and in the trusted directories (/usr/lib
and /lib).

       ldconfig checks the header and file names of the  libraries  it  encounters  when
determining  which versions should have their links updated.  ldconfig ignores symbolic links
when scanning  for libraries.

       ldconfig will attempt to deduce the type of ELF libs (ie. libc 5.x or libc 6.x (glibc))
based  on  what  C  libraries  if  any  the library was linked against, therefore when making
dynamic  libraries, it is wise to explicitly link against libc (use -lc).  ldconfig is capable
of storing  multiple  ABI  types of libraries into a single cache on architectures which allow
native  running of multiple ABIs, like ia32/ia64/x86_64 or sparc32/sparc64.

       Some existing libs do not contain enough information to allow the  deduction  of  their
type, therefore  the /etc/ld.so.conf file format allows the specification of an expected type.
This  is only used for those ELF  libs  which  we  can  not  work  out.  The  format  is  like
this  "dirname=TYPE",  where type can be libc4, libc5 or libc6.  (This syntax also works on the
com mand line).  Spaces are not allowed. Also see the -p option.

       Directory names containing an = are no longer legal unless they also  have  an  expected
type  specifier.

       ldconfig  should  normally be run by the super-user as it may require write permission
on some root owned directories and files. If you use -r option to change the root directory,
you don't have to be super-user though as long as you have sufficient right to that directory
tree.
```

ld.so.conf

```
$ cat /etc/ld.so.conf
include /etc/ld.so.conf.d/*.conf
```

Objective 5: Use Debian Package Management

- A group of tools used to get and manage software packages on Debian systems
- dpkg is the main package manager covered on 101

Debian Package Management Overview

- Debian packages usually contain program, configuration files, documentation and noted dependencies on other packages

Package name

- Short and descriptive. Usually contain a hyphen when multiple words.

Version number

- Usually show as "major.minor.patchlevel"

A file extension

- .deb

Managing Debian Packages

- Original tool is dpkg, which operates directly on the .deb package
- Alternative tools are:
  . apt-get - Uses package names and obtain them from an external source (CD, ftp, etc...)
  . dselect - Uses an interactive menu (similar to aptitude)
  . alien - Allows the install of non Debian packages, like RPM

Dpkg

man dpkg

```
$ man dpkg
dpkg(1)

NAME
       dpkg - package manager for Debian

SYNOPSIS
       dpkg [options] action

DESCRIPTION
       dpkg is a tool to install, build, remove and manage Debian  packages.  The  primary  and
more
       user-friendly front-end for dpkg is dselect(1). dpkg itself is controlled entirely via
command
       line parameters, which consist of exactly one action and zero or  more  options.  The
action-
       parameter tells dpkg what to do and options control the behavior of the action in some
way.

ACTIONS
       dpkg -i | --install package_file...
       dpkg --configure package ... | -a | --pending
       dpkg -r | --remove | -P | --purge package ... | -a | --pending
       dpkg --update-avail | --merge-avail Packages-file
       dpkg -I | --info filename [control-file]
       dpkg -l | --list package-name-pattern ...
       dpkg -s | --status package-name ...
       dpkg -L | --listfiles package ...
       dpkg -S | --search filename-search-pattern ...
       dpkg -p | --print-avail package
       -R | --recursive
       -G     Don't install a package if a newer  version  of  the  same  package  is  already
installed. This is an alias of --refuse-downgrade.
       -E | --skip-same-version
              Don't  install  the  package  if  the  same  version  of  the package is already
installed.
```

Install a package

```
$ dpkg -i <folder/package name>
```

Upgrade a package

```
$ dpkg -G <folder/package name>
```

Purge or remove a package

```
$ dpkg -r <package name>
$ dpkg --purge <package name>
```

Find a package with specific files

```
$ dpkg -S apt-get
apt: /usr/bin/apt-get
```

Get status information on a package

```
$ dpkg -s apt
Package: apt
Status: install ok installed
Priority: important
Section: admin
Installed-Size: 4444
Maintainer: Ubuntu Core Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Architecture: i386
Version: 0.7.6ubuntu14
```

List files in a package

```
$ dpkg -L apt | grep '^/usr/bin'
/usr/bin
/usr/bin/apt-cache
/usr/bin/apt-cdrom
/usr/bin/apt-config
/usr/bin/apt-get
/usr/bin/apt-key
/usr/bin/apt-mark
```

Apt-get

man apt-get

```
APT-GET(8)

NAME
       apt-get - APT package handling utility -- command-line interface

DESCRIPTION
       apt-get is the command-line tool for handling packages, and may be considered the user's
       "back-end" to other tools using the APT library. Several "front-end" interfaces exist,
such as
       dselect(8), aptitude, synaptic, gnome-apt and wajig.

OPTIONS
       -d, --download-only
           Download only; package files are only retrieved, not unpacked or installed.
       -s, --simulate, --just-print, --dry-run, --recon, --no-act
           No action; perform a simulation of events that would occur but do not actually
change the system.
      -y, --yes, --assume-yes
           Automatic yes to prompts; assume "yes" as answer to all prompts and run non-
interactively.
```

Options

- (-s) - Simulates installs

Apt-setup

- Interactive tool to configure repositories
- Not available on all distros

apt-cdrom add

- Add CD-ROM to the repositories

The /etc/apt/apt.conf File

- Allows users to customize options for apt-get
- The command apt-config can be used by scripts to call for info from /etc/apt/apt.conf

Alien

man alien

```
NAME
     alien - Convert or install an alien binary package

SYNOPSIS
    alien [--to-deb] [--to-rpm] [--to-tgz] [--to-slp] [options] file [...]

DESCRIPTION
alien is a program that converts between Redhat rpm, Debian deb, Stampede slp, Slackware tgz,
and Solaris pkg file formats. If you want to use a package from another linux distribution than
the one you have installed on your system, you can use alien to convert it to your preferred
package format and install it. It also supports LSB packages.

OPTIONS
     -i, --install
         Automatically install each generated package, and remove the package file after it has
been installed.
     -r, --to-rpm
         Make rpm packages.
     -t, --to-tgz
         Make tgz packages.
     -d, --to-deb
         Make debian packages. This is the default.
```

Objective 6: Use Red Hat Package Manager (RPM)

RPM Overview

- RPM keeps a database of all installed packages. This database is consulted when packages are removed, queried and installed
- There not as many programs available as for apt
- Has extensive capabilities
- Does not keep pkage information to the same extend that dpkg does
- RPM packages have four elements:
  . Name - If multiple words are used they will have a hyphen separating them.
  . Version - major.minor.patchlevel
  . Revision - Release number for the package
  . Architecture - Depends on the PC architecture. Types usually are i386 for Intel 80386; i586 for Intel Pentium; i686 for Intel Pentium Pro, II, Celeron, III an 4; athlon for AMD; alpha for Alpha; ia64 for Itanium; ppc for Power PC; sparc for SPARC; noarch for any architecture

gcc-2.96-113.i386.rpm
 | | | | |------ extension
 | | | |----------- architecture
 | | |---------------- revision
 | |--------------------- version
 |-------------------------- name

Running RPM

- RPM has the option of using extra options within what's called modes. For example, when using “rpm -i”, a series of options are enabled to be used with that mode. eg:

```
$ rpm -i [install or mode options] PACKAGE_FILE...
```

man rpm

```
NAME
    rpm -- RPM Package Manager

SYNOPSIS

Querying:
    rpm {-q | --query} [PACKAGE_NAME]

Maintaining installed 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} [options] PACKAGE_NAME...
    rpm {-V | --verify} [options] PACKAGE_NAME
```

Install/upgrade Mode (-i-U)
- Requires full name of package file
- Upgrade mode is a variant of install mode. Another variant is the freshen mode (-F), which upgrades old installed packages to a more recent one without handling dependencies
- Frequently used options:
  . --force - Can be used to install an old package over a newer one *
  . -h (--hash) - Prints a progress bar in the form of 50 “#”
  . --nodeps - Installs packages without checking for dependencies (makes dependency database inconsistent) *
  . --test - Only tests install
  . -v - Verbose
  . -vv - Extra verbose

Note: If a install fails due to dependency, install the dependencies either before the desired package or with the package

```
$ rpm –i <dependency 1> <dependency 2> <dependency 3> <desired package>
```

Uninstall Mode (-e)
- Requires package name only
- By default rpm uninstalls a package only if no other package depends on it
- Frequently used options
  . --nodeps - Removes packages without checking for dependencies (makes dependency database inconsistent). This is similar to 'rpm -i --force' *
  . --test - Only tests removal

Query mode (-q)
- Can be used on installed packages and raw package files
- Non-root users can query package for information
- Frequently used query package selection options
  . -a (--all) - Displays all packages installed *
  . -f - Displays the package with the filename *
  . -p - Query a package file (complete name) *
- Frequently used query information selection options
  . -c (--configfiles) - Lists configuration files only
  . -d (--docfiles) - Lists documentation files only *
  . -i - Displays information about a package *
  . -l (--list) - Lists all files contained in a package *
  . -R (--requires) - Lists packages on which this package depends *
  . --whatrequires - Lists packages that depend on this *

Package Integrity
- Digital signature and digiest verygfication
  . --checksig (-k) - Checks the integrity of an RPM package *
  . --import - Imports signature key for package *

Verify mode (-V)
- Compares an installed package against it's expected configuration from RPM database
- Frequently used options:
  . --nofiles - Ignores missing files
  . --nomd5 - Ignores MD5 errors
  . --nopgp - Ignores PGP checking

Configuring RPM
- RPM can be changed to control run-time operation
  . Old config file - /etc/rpmrc
  . New location - /usr/lib/rpm/rpmrc (this file gets overwritten when the RPM package is upgraded)

Topic 1.103

GNU and Unix Commands

Objective 1: Work on the Command Line

- The shell provides an interface layer between the Linux kernel and user
- Steve Bourne wrote the first shell which was called sh
- Bourne Again Shell is a variant of sh and the default Linux shell (bash). It provides many improvements and function capabilities, while still substantially compatible with sh
- tcsh is a variant of the csh (cshell) which also came from sh

Shell Basics

- When you first start it does some "housekeeping" and them it's ready to take commands from a *standard output* device
- In order for bash to execute a program, there needs to be one of the following:
  . A bash built-in command
  . An executable on the $PATH
  . Explicitly defined

Entering Commands at the Command Prompt

- Most commands issued to the shell will require the following:
  . A valid command (built-in to shell or within $PATH);
  . Command option;
  . Argument;
  . Line acceptance (hitting enter).
- As mentioned before, some of the commands are built-in within shell (cd, break, exec)
- Usually a dash is expected before the option. It also helps differentiate the option from the argument, but some times its not necessary. eg:

```
$ tar czvf backup.tar.gz file1 file2 file3
$ tar -czvf backup.tar.gz file1 file2 file3
```

Entering Commands not in the $PATH

- Most of the times it's easier to add the program directory to $PATH if it's a program that you use frequently.
- Other option is to execute the complete path (fully qualified filename) to the program. eg:

```
$ /usr/bin/ls
```

Entering Multiple-line Commands Interactively

- Usually when you enter a multi-line command, like a loop (for, until, if, etc...), bash will ask you for the subsequent lines of the command until a command has been completed. The prompt shown on this case is a shell variable called PS2, or ">".

```
$ <commands for arg1>
command output
$ <commands for arg2>
command output
$ <commands for arg3>
command output
```

- This can be done easier by using bash's loop construction

```
$ for var in arg1 arg2 arg3
>do
>echo var
><command>
>done
```

```
arg1
command output
arg2
command output
arg3
command output
```

Entering Command Sequences
- To add more than one command you can use the ";" sign. eg:

```
$ ls -1; pwd
Desktop
documents
/home/victor
```

Variables

Shell Variable Basics
- Set of important variables maintained by shell that has important information for the execution of bash
- Example:
  . $PS1 - Prompt String. Content of command prompt that will be displayed when bash is ready to accept commands

Ubuntu
```
victor@victor-laptop:~$ echo $PS1
${debian_chroot:+($debian_chroot)}\u@\h:\w\$
```

Red Hat
```
[user@hostname user]$ echo $PS1
[\u@\h \w]\$
```
  . u = User
  . h = Hostname
  . w = Working Directory

- $PATH - Leads bash to a list of directories where most programs are. Save you from type "/usr/bin/less"
- Some shell variables are also available to programs executed on the shell. They are called environment variables. To create/export a shell variable to an environment variable, you run the following code:

```
$ export <variable name>
```

- Environment variable are usually displayed in capital, this is however a guideline only
- Exported variables are not available to it's parent shell
- $SHELL - Holds information to the current shell
- $PPID - Parent process ID of the current process
- Variables, environment variables and child process:

```
[root@centos ~]# VAR1=1 ; VAR2=2 ; export VAR3=3
[root@centos ~]# echo $VAR{1,2,3} $$ $PPID
1 2 3 23033 23011
[root@centos ~]# bash -c 'echo $VAR{1,2,3} $$ $PPID'
3 25641 23033
```

Quoting
- Shell expands double quotes (“), however single quotes (')are not expanded

```
[root@centos ~]# echo "$SHELL" '$SHELL'
/bin/bash $SHELL
```

- When passing to a child shell, double quotes are expended before passing the command (in the parent shell), while single quotes are expended in the child process

```
[root@centos ~]# bash -c echo 'parent $$ $PPID'

[root@centos ~]# bash -c "echo parent $$ $PPID"
parent 23033 23011
[root@centos ~]# bash -c 'echo child $$ $PPID'
child 25798 23033
```

Braces And Variable Names
- Making sure that name gets expanded

```
[root@centos ~]# echo "$USER_/this is my user name"
/this is my user name
[root@centos ~]# echo "${USER}_/this is my user name"
root_/this is my user name
```

Env
- The env command can be used to display current environment variables (without argument) or execute a command in a custom environment
  . -i - Clear the current envritonment before running the command
  . -u - Unsets environment variables that you do not wish to pass

```
[root@centos ~]# ksh -c 'echo $SHELL'
/bin/bash
[root@centos ~]# env -i ksh -c 'echo $SHELL'
/bin/sh
```

Set And Unset
- unset - Removes variable (including environment variables)
- set - Controls many different shell aspects.
  . Example - (-u) will add an error for variables that are not set

```
[root@centos ~]# set -u
[root@centos ~]# echo $var1
bash: var1: unbound variable
[root@centos ~]# set +u
[root@centos ~]# echo $var1
```

- declare - Similar to set.

```
[root@centos ~]# declare > declare1
[root@centos ~]# set > set1
[root@centos ~]# diff set1 declare1
57c57
< _=declare
---
> _=--color=tty
```
Note: The last line in the variable list is related to your last command

Exec
- Runs a command that replaces the current shell

```
[root@centos ~]# echo $$
27316
[root@centos ~]# bash
[root@centos ~]# echo $$
27369
[root@centos ~]# exec ls
anaconda-ks.cfg   clearlooks.tar.gz  Desktop  install.log         set1  um2
bluecurve.tar.gz  declare1           icons    install.log.syslog  um1   umdois
[root@centos ~]# echo $$
27316
```

Command History And Editing

- This part can be divided into 3 different groups: command history, expansion and editing.

Command History

- Before bash interprets each command, it stores them in the history list. Once the session is closed bash writes the last commands to the history file
- Can be turned off with 'set +o history' and turned on with 'set -o history'
- The $HISTISIZE variable controls the history list. The list is usually set up to a limit of 500
- Commands from a previous bash session are also stored on ~/.bash_history (or where the $HISTFILE variable points to)
  . Note: if you use multiple tabs on one shell window, only the last tab closed will write to ~/.bash_history

```
$ echo $HISTSIZE
500

$ echo $HISTFILE
/home/user/.bash_history
```

- You can also use the command “history” to view the history:

```
$ history
    1  ls
    2  sudo rm -r Examples
    3  ls
    4  xkill
    5  df
    6  sudo apt-get update
    7  sudo apt-get upgrade
```

- Displays last *N* of history

```
$ history 2
  509  pwd
  510  history 2
```

- Deletes *N* lines from history

```
$ history -d 2
  509  pwd
  510  history 2
```

History Expansion

- Uses either a line number from the history or a portion of a previous command to re-execute that command.

Command History Expansion Designators

Designator	Description
!!	Most recent command (bang-bang)
!n	Refer to command n from history
!-n	Current command -n from history
!#	Current command being typed
!string	Most recent command starting with string
!?string	Most recent command containing string
^string1^string2	Repeat the last command replacing with first occurrence of string1 with string2

Basic Command History Editing (Emacs)

Key	Description	Variant
Ctrl+p	Previous line	up
Ctrl+n	Next line	down
Ctrl+b	Back on char	left
Ctrl+f	Forward on char	right
Ctrl+a	Beginning of the line	
Ctrl+e	End of line	
Ctrl+l	Clear screen and keep current line	
m+<	Top history	
m+>	Bottom history	
Ctrl+d	Delete char from right	
Ctrl+k	Delete from cursor to end of line	
Ctrl+y	Paste previously cut	
m+d	Delete word	
Ctrl+r <text>	Reverse search	
Ctrl+s <text>	Forward search	

- A colon can also be used to modify a previous command (similar to '^string1^string2')

```
$ env -i bash -c 'echo $$'
9559
victux ~ $ !en:s/$$/$PPID/
env -i bash -c 'echo $PPID'
9272
```

Paths

- The two most common programs used to find program paths are 'type' and 'which'

```
$ which env set ls
/usr/bin/env
/bin/ls

$ type env set ls
env is hashed (/usr/bin/env)
set is a shell builtin
ls is aliased to `ls --color=auto'
```

- A tilde (~) can also be used to link to a user's home directory
  . ~/ - Users home directory
  . ~username - Also users home directory
- CDPATH is a variable that holds colon separated paths that will be searched when using relative paths

```
$ pwd
/
$ cd images/
$ pwd
/home/victor/images
```

- Another variable that can be used is OLDPWD (-)

```
$ pwd
/home/victor/images
$ cd -
$ pwd
/
```

Command Substitution

- Can be used by aliases on .bashrc or by using `` on command `command` or $(command)

Note: The later option is better when posting commands on web forums for easier understanding

Applying Commands Recursively Though A Directory Tree

- Extra care is required when executing commands recursively (even more when acting as root)
- Using recursive option to descend into the subfolders. eg:

```
$ chmod -R 500 /var/www/
```

- In case you wanted to change only HTML files within the folder and subfolders, you can use the -exec option from find

```
$ find /var/www/ -name *.html -exec chmod 500 {} \;
```

. { } will be replaced by the file name
. \ is used to comment out the ";"
. ; is used to end the command

Commands And Sequences

Echo

- Using quotes on strings will preserve additional white spaces
- Enabling escape sequences (-e)
  . \a - Alert (system bell)
  . \b - Backspace
  . \n - New line
  . \t - Tab

Escapes And Line Continuation

- A backslash can be interpreted as a line break for a command

```
[root@centos ~]# this is my command\
> and here's the continuation
```

Metacharacters And Control Operators

- A metacharacter is interpreted in a different way by shell. To use one as a normal character you will need to escape it
- Some metacharacters have special attributes, like control operator. Here are two of them:
  . && - Second command is executed if first command returns a 0
  . || - Second command is executed if first command returns a non 0

```
[root@centos ~]# echo $USER && ls dir || echo $PS1
root
ls: dir: No such file or directory
[\u@\h \W]\$
```

Man Pages

- There are 8 common manual pages
- Manual pages are usually installed with it pertaining package

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]

- For some items that are within multiple sections you can specify the sections number before the item name

```
$ man 7 man
```

Other Commands
- whatis - Searches man's page and displays description
- apropos - Searches for item on man pages names and descriptions (very similar to 'man -k')
- info - Online documentation. Provides option to navigate between sections

Objective 2: Process Text Streams Using Filters

- Multiple commands can be used to produce a text stream, which are modified at each step in a pipeline formation
- Text filtering is the process of taking a text stream and process it before sending it to an output
- Some commands require a hyphen (-) instead of a filename to read the output from another command

Commands

cat (catenate)
- Takes output from stdin (-) if no file name is specified

```
CAT(1)

NAME
       cat - concatenate files and print on the standard output

SYNOPSIS
       cat [OPTION] [FILE]...

DESCRIPTION
       Concatenate FILE(s), or standard input, to standard output.

       -A, --show-all
              equivalent to -vET

       -b, --number-nonblank
              number nonempty output lines

       -e     equivalent to -vE

       -E, --show-ends
              display $ at end of each line

       -n, --number
              number all output lines
```

cut
- Removes (cut out) selected columns or fields from one or more files (extracts fields from txts)
- Default delimiter is a tab character

man cut

```
CUT

NAME
       cut - remove sections from each line of files
```

```
SYNOPSIS
       cut [OPTION]... [FILE]...

DESCRIPTION
       Print selected parts of lines from each FILE to standard output.
      Mandatory arguments to long options are mandatory for short options too.

       -b, --bytes=LIST
              select only these bytes
      -c, --characters=LIST
              select only these characters
      -d, --delimiter=DELIM
              use DELIM instead of TAB for field delimiter
      -f, --fields=LIST
              select  only  these  fields
```

- Confirms a user name on the /etc/passwd file:

```
$ cat /etc/passwd | grep user | cut -f1 -d:
user
```

expand

- Converts tabs to spaces (usually by default, 1 tab = 8 spaces)

man expand

```
EXPAND

NAME
       expand - convert tabs to spaces

SYNOPSIS
       expand [OPTION]... [FILE]...

DESCRIPTION
       Convert  tabs in each FILE to spaces, writing to standard output.
      Mandatory arguments to long options are mandatory for short options too.

       -i, --initial
              do not convert tabs after non blanks
      -t, --tabs=NUMBER
              have tabs NUMBER characters apart, not 8
      -t, --tabs=LIST
              use comma separated list of explicit tab positions
```

fmt

- Formats text to a specific width by filling lines and removing newline characters (so it fits margin)

man fmt

```
FMT

NAME
       fmt - simple optimal text formatter

SYNOPSIS
       fmt [-DIGITS] [OPTION]... [FILE]...

DESCRIPTION
       Reformat each paragraph in the FILE(s), writing to standard output.
       Mandatory arguments to long options are mandatory for short options too.

       -c, --crown-margin
              preserve indentation of first two lines
      -p, --prefix=STRING
              reformat only lines beginning with STRING, reattaching the prefix to reformatted
```

```
lines
      -s, --split-only
              split long lines, but do not refill
      -t, --tagged-paragraph
              indentation of first line different from second
      -u, --uniform-spacing
              one space between words, two after sentences
      -w, --width=WIDTH
              maximum line width (default of 75 columns)
```

head

- Prints first lines of one or more files

man head

```
HEAD

NAME
       head - output the first part of files

SYNOPSIS
       head [OPTION]... [FILE]...

DESCRIPTION
       Print  the  first 10 lines of each FILE to standard output.
       Mandatory arguments to long options are mandatory for short options too.

       -c, --bytes=[-]N
              print  the  first N bytes of each file
      -n, --lines=[-]N
              print the first N lines instead of the first 10
```

join

- Prints a pair of input lines, from file1 and file2, that have and identical field (or join field)

man join

```
JOIN

NAME
       join - join lines of two files on a common field

SYNOPSIS
       join [OPTION]... FILE1 FILE2

DESCRIPTION
       For  each  pair  of  input lines with identical join fields, write a line to standard
output.
       The default join field is the first, delimited by whitespace.

       -a FILENUM
              print  unpairable lines coming from file FILENUM, where FILENUM is 1 or 2,
corresponding to FILE1 or FILE2
      -e EMPTY
              replace missing input fields with EMPTY
      -i, --ignore-case ignore differences in case when comparing fields
      -j FIELD
              equivalent to '-1 FIELD -2 FIELD'
      -o FORMAT
              obey FORMAT while constructing output line
      -t CHAR
              use CHAR as input and output field separator
      -v FILENUM
              like -a FILENUM, but suppress joined output lines
      -1 FIELD
              join on this FIELD of file 1
      -2 FIELD
```

```
              join on this FIELD of file 2
```

nl

- Numbers the lines of files
- Can also work with headers and footers, as well as headers and footer styles

man nl

```
NL

NAME
       nl - number lines of files

SYNOPSIS
       nl [OPTION]... [FILE]...

DESCRIPTION
       Write  each  FILE to standard output, with line numbers added.
       Mandatory arguments to long options are mandatory for short options too.

       -b, --body-numbering=STYLE
              use STYLE for numbering body lines
      -d, --section-delimiter=CC
              use CC for separating logical pages
      -f, --footer-numbering=STYLE
              use STYLE for numbering footer lines
      -h, --header-numbering=STYLE
              use STYLE for numbering header lines
      -i, --page-increment=NUMBER
              line number increment at each line
      -l, --join-blank-lines=NUMBER
              group of NUMBER empty lines counted as one
       -n, --number-format=FORMAT
              insert line numbers according to FORMAT
      -p, --no-renumber
              do not reset line numbers at logical pages
      -s, --number-separator=STRING
              add STRING after (possible) line number
      -v, --first-page=NUMBER
              first line number on each logical page
      -w, --number-width=NUMBER
              use NUMBER columns for line numbers
```

od

- Dump files in octal and other formats

man od

```
OD

NAME
       od - dump files in octal and other formats

SYNOPSIS
       od [OPTION]... [FILE]...
       od [-abcdfilosx]... [FILE] [[+]OFFSET[.][b]]
       od --traditional [OPTION]... [FILE] [[+]OFFSET[.][b] [+][LABEL][.][b]]

DESCRIPTION
       Write  an  unambiguous  representation,  octal  bytes by default, of FILE to standard
output.

USAGE
       -t type
              Specifies the type of output
       A
              Name character
       c
```

```
              ACII or backslash escape
       O
              Octal (default)
       x
              Hexadecimal
```

paste

- Pastes together lines on one or more files into vertical columns
-

man paste

```
PASTE(1)

NAME
       paste - merge lines of files

SYNOPSIS
       paste [OPTION]... [FILE]...

DESCRIPTION
       Write  lines  consisting of the sequentially corresponding lines from each FILE,
separated by TABs, to standard output.
      Mandatory arguments to long options are mandatory for short options too.

       -d, --delimiters=LIST
              reuse characters from LIST instead of TABs
      -s, --serial
              paste one file at a time instead of in parallel
```

- Examples

```
$ cat test1
1
2
3
4
5
6

$ cat test2
a
b
s
c
d
e

$ paste test1 test2
1       a
2       b
3       s
4       c
5       d
6       e

$ paste -s test1 test2
1       2       3       4       5       6
a       b       s       c       d       e

$ paste -s test1 test2 | expand | tr -s '[:blank:]'
1 2 3 4 5 6
a b s c d e
```

pr

- Converts a text file into paginated
- The default header includes a file name, file creation date and time, page number and two lines of a blank footer

man pr

```
PR

NAME
       pr - convert text files for printing

SYNOPSIS
       pr [OPTION]... [FILE]...

DESCRIPTION
       Paginate or columnate FILE(s) for printing.

USAGE
       -d, --double-space
              double space the output
      -D, --date-format=FORMAT
              use FORMAT for the header date
       -h HEADER, --header=HEADER
              use  a  centered  HEADER instead of filename in page header,
       -l PAGE_LENGTH, --length=PAGE_LENGTH
              set  the page length to PAGE_LENGTH (66) lines (default number of lines of text
56,
       -w PAGE_WIDTH, --width=PAGE_WIDTH
              set  page  width  to  PAGE_WIDTH  (72) characters for multiple text-column output
only,
       -W PAGE_WIDTH, --page-width=PAGE_WIDTH
              set page width to PAGE_WIDTH (72) characters always
```

- Example

```
$ cat /etc/apt/sources.list | pr -h Tutorial -l 15


2007-11-27 20:30                    Tutorial                    Page 1


deb cdrom:[Ubuntu 7.10 _Gutsy Gibbon_ - Release i386 (20071016)]/ gutsy main restricted
# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.

deb http://ca.archive.ubuntu.com/ubuntu/ gutsy main restricted
```

sort

- Writes input to stdout sorted alphabetically
- Fields are delimited by blank spaces or tabs
- Two texts can be sorted at the same time

man sort

```
SORT

NAME
       sort - sort lines of text files

SYNOPSIS
       sort [OPTION]... [FILE]...

DESCRIPTION
       Write sorted concatenation of all FILE(s) to standard output.

       Mandatory arguments to long options are mandatory for short options too.  Ordering
options:

USAGE
       -b, --ignore-leading-blanks
              ignore leading blanks
```

```
      -f, --ignore-case
             fold lower case to upper case characters
      -M, --month-sort
             compare (unknown) < 'JAN' < ... < 'DEC'
      -n, --numeric-sort
             compare according to string numerical value
      -r, --reverse
             reverse the result of comparisons

       Other options:

       -k, --key=POS1[,POS2]
              start a key at POS1, end it at POS2 (origin 1)
       -o, --output=FILE
              write result to FILE instead of standard output
       -t, --field-separator=SEP
              use SEP instead of non-blank to blank transition
```

- Example. Sorts items by PID

```
$ ps u | sort -k 2 -n
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
victor    5662  0.0  0.2   5668  3052 pts/0    Ss+  19:41   0:00 bash
victor    6287  0.0  0.3   5700  3116 pts/1    Ss   20:11   0:00 bash
victor    6440  0.0  0.0   2620  1000 pts/1    R+   20:38   0:00 ps u
victor    6441  0.0  0.0  28532   680 pts/1    S+   20:38   0:00 sort -k 2 -n
```

split

- Splits a file into different groups/files

man split

```
SPLIT

NAME
       split - split a file into pieces

SYNOPSIS
       split [OPTION] [INPUT [PREFIX]]

DESCRIPTION
       Output fixed-size pieces of INPUT to PREFIXaa, PREFIXab, ...;

       Mandatory arguments to long options are mandatory for short options too.

       -a, --suffix-length=N
              use suffixes of length N (default 2)
      -b, --bytes=SIZE
              put SIZE bytes per output file
       -C, --line-bytes=SIZE
              put at most SIZE bytes of lines per output file
      -d, --numeric-suffixes
              use numeric suffixes instead of alphabetic
      -l, --lines=NUMBER
              put NUMBER lines per output file
```

- Example. Divide the file into groups of 3 with the default suffix (aa, ab, ac, ad, etc...)

```
$ cat test3
1  a
2  b
3  s
4  c
5  d
6  e

$ split -3 test3 test-
```

```
$ cat test-aa
1  a
2  b
3  s

$ cat test-ab
4  c
5  d
6  e
```

tac

- Opposite of *cat*. It prints file in reverse order

man tac

```
TAC

NAME
       tac - concatenate and print files in reverse

SYNOPSIS
       tac [OPTION]... [FILE]...

DESCRIPTION
       Write each FILE to standard output, last line first.  With no FILE, or when FILE is -,
read standard input.

       -b, --before
              attach the separator before instead of after
      -r, --regex
              interpret the separator as a regular expression
      -s, --separator=STRING
              use STRING as the separator instead of newline
```

tail

- Prints the last few lines of one or more files

man tail

```
TAIL

NAME
       tail - output the last part of files

SYNOPSIS
       tail [OPTION]... [FILE]...

DESCRIPTION
       Print the last 10 lines of each FILE to standard output.

       --retry
              keep trying to open a file even if it is inaccessible when tail starts or if it
becomes inaccessible later
       -c, --bytes=N
              output the last N bytes
       -n, --lines=N
              output the last N lines, instead of the last 10
       -f, --follow[={name|descriptor}]
              output appended data as the file grows
```

tr

- Translate characters from string 1 to the corresponding characters on string 2

man tr

```
TR

NAME
       tr - translate or delete characters

SYNOPSIS
       tr [OPTION]... SET1 [SET2]

DESCRIPTION
       Translate, squeeze, and/or delete characters from standard
input, writing to standard output.

USAGE
       -c, -C, --complement
              first complement SET1
      -d, --delete
              delete characters in SET1, do not translate
      -s, --squeeze-repeats
              replace each input sequence of a repeated character
that is listed in SET1 with a single occurrence of that character
      -t, --truncate-set1
              first truncate SET1 to length of SET2
       [:alnum:]
              all letters and digits
      [:alpha:]
              all letters
      [:blank:]
              all horizontal whitespace
      [:cntrl:]
              all control characters
      [:digit:]
              all digits
      [:graph:]
              all printable characters, not including space
      [:lower:]
              all lower case letters
      [:print:]
              all printable characters, including space
      [:punct:]
              all punctuation characters
      [:space:]
              all horizontal or vertical whitespace
      [:upper:]
              all upper case letters
      [:xdigit:]
              all hexadecimal digits
```

- Examples

```
# Change case
$ tr a-z A-Z
# or
$ tr '[:lower:]' '[:upper:]'

# Replace contigious space with one space
$ tr -s '[:blank:]'

# Delete blank space
$ tr -d '[:blank:]'
```

unexpand

- Converts spaces to tab. Default is 8 spaces = 1 tab

man unexpand

```
UNEXPAND
```

```
NAME
       unexpand - convert spaces to tabs

SYNOPSIS
       unexpand [OPTION]... [FILE]...

DESCRIPTION
       Convert blanks in each FILE to tabs, writing to standard output.  With no FILE, or when
FILE is -, read standard input.

USAGE
       -a, --all
              convert all blanks, instead of just initial blanks
      --first-only
              convert only leading sequences of blanks (overrides -a)
      -t, --tabs=N
              have tabs N characters apart instead of 8 (enables -a)
```

uniq

- Removes consecutive duplicate lines (first line must be right before second)

man uniq

```
UNIQ

NAME
       uniq - report or omit repeated lines

SYNOPSIS
       uniq [OPTION]... [INPUT [OUTPUT]]

DESCRIPTION
       Discard all but one of successive identical lines from INPUT (or standard input),
writing to OUTPUT (or standard output).

USAGE
       -c, --count
              prefix lines by the number of occurrences
      -d, --repeated
              only print duplicate lines
      -D, --all-repeated[=delimit-method] print all duplicate lines
              delimit-method={none(default),prepend,separate} Delimiting is done with blank
lines.
      -i, --ignore-case
              ignore differences in case when comparing
      -u, --unique
              only print unique lines
```

wc

- Prints counts of bytes, characters, words and lines of a file

man wc

```
WC(1)                                                                 User Commands
WC(1)

NAME
       wc - print the number of newlines, words, and bytes in files

SYNOPSIS
       wc [OPTION]... [FILE]...

DESCRIPTION
       Print  newline,  word,  and byte counts for each FILE, and a total line if more than one
FILE is specified.

USAGE
```

```
      -c, --bytes
              print the byte counts
      -m, --chars
              print the character counts
      -l, --lines
              print the newline counts
      -L, --max-line-length
              print the length of the longest line
      -w, --words
              print the word counts
```

xargs

- Executes command based on a previous command. Very similar to -exec, however its more efficient. Example:
  . The first example will pause every time it finds a result, pass the information to -exec and wait for it to be done before it continues the search.
  . On example B, find will not stop, and xargs will collect data at an approximate group of 20-50 names

```
# Example A
$ find / -name <whatever> -exec chmod 777 {} \;

# Example B
$ find / -name <whatever> | xargs chmod 777
```

man xargs

```
XARGS

NAME
       xargs - build and execute command lines from standard input

SYNOPSIS
       xargs  [-0prtx]  [-E  eof-str]  [-e[eof-str]] [--eof[=eof-str]] [--null] [-d delimiter]
[--delimiter delimiter] [-I replace-str] [-i[replace-str]] [--replace[=replace-str]]  [-l[max-
lines]] [-L  max-lines] [--max-lines[=max-lines]]  [-n   max-args] [--max-args=max-args] [-s
max-chars] [--max-chars=max-chars]   [-P  max-procs] [--max-procs=max-procs] [--interactive]
[--verbose] [--exit] [--no-run-if-empty] [--arg-file=file] [--show-limits] [--version] [--help]
[command [initial-arguments]]

DESCRIPTION
       This manual page documents the GNU version of xargs.  xargs reads items from the
standard input, delimited by blanks (which can be protected  with double  or single quotes or a
backslash) or newlines, and executes the command (default is /bin/echo) one or more times with
any initial-arguments followed by items read from standard input. Blank lines on the standard
input are ignored.

       Because Unix filenames can contain blanks and newlines, this default behaviour is often
problematic; filenames containing blanks  and/or  newlines are  incorrectly  processed  by
xargs. In these situations it is better to use the `-0' option, which prevents such problems.
When using this option you will need to ensure that the program which produces the input for
xargs also uses a null character as a separator.  If that program  is GNU find for example, the
`-print0' option does this for you.
```

sed (stream editor)

- Loads the input into pattern space, applies the command options to the text and then move it to standard out
- Also has a hold buffer which can replace, be added or exchange the pattern space
- Lines numbers can be used and multiple commands can be separated by a semi-colon (;)

```
# cat 3
1 apple
2 pear
3 banana
# sed '2d;s/a/A/g' 3
1 Apple
3 bAnAnA
```

- A range of lines can also be given and separated by a coma (,). They can be either a number or a position pointer (^$)

```
# sed '2,3s/a/A/g' 3
1 apple
2 peAr
3 bAnAnA
```

- Multiple commands can also be given inside curly bracket { }

```
# sed '2{ s/r/R/g; s/e/E/g }' 3
1 apple
2 pEaR
3 banana
```

- Line numbers can be added with ('=')

```
# sed '=' 1
1
 pear
2
 apple
3
 banana
```

- Two lines can be added into the pattern space with (N)

```
# sed '=' 1 | sed 'N;s/\n//g'
1 pear
2 apple
3 banana
```

```
SED(1)                          User Commands                          SED(1)

NAME
       sed - stream editor for filtering and transforming text

SYNOPSIS
       sed [OPTION]... {script-only-if-no-other-script} [input-file]...

DESCRIPTION
       Sed  is a stream editor.  A stream editor is used to perform basic text
       transformations on an input stream (a file or input from  a  pipeline).
       While  in  some  ways similar to an editor which permits scripted edits
       (such as ed), sed works by making only one pass over the input(s),  and
       is consequently more efficient.  But it is sed's ability to filter text
       in a pipeline which particularly distinguishes it from other  types  of
       editors.

       -n, --quiet, --silent
              suppress automatic printing of pattern space

       -e script, --expression=script
               add the script to the commands to be executed

       -f script-file, --file=script-file
              add the contents of script-file to the commands to be executed

       -i[SUFFIX], --in-place[=SUFFIX]
              edit files in place (makes backup if extension supplied)

       -c, --copy
              use  copy  instead  of  rename  when  shuffling files in -i mode
               (avoids change of input file ownership)

       -l N, --line-length=N
              specify the desired line-wrap length for the 'l' command
```

Objective 3: Perform Basic File Management

Filesystem Objects

- As most OS, Linux keeps its objects organized in a hierarchy layout
- The most common objects in Linux (although not the only ones) are directories and files
- Information about each object is stored in a table, and each object has a unique number in that table

Directories And Files

- A directory is a container to hold files and another directories
- A file exists within a directory and it's responsible for raw data storage

Inodes

- Carries information about objects (where they are located, modification time, security settings, etc...)
- Filesystems (like ext2 and ext3) have finite numbers of inodes depending on the size of the system (and other options)
- Multiple objects can share the same inode by linking
- Is also known as the file serial number
- When listing directories with long listing the second field tells us the amount of hard links to that file

```
# ls -li
2278809 -rw-r--r-- 2 root root 21 Feb 28 17:05 1
2278809 -rw-r--r-- 2 root root 21 Feb 28 17:05 1.1
2269220 lrwxrwxrwx 1 root root  1 Mar  1 17:01 1.2 -> 1
```

File And Directory Management Commands

- LPI1 tests on the command-line tools, not the GUI
- When using 'ls', if a directory name Is given it's contents will be displayed instead of the folder itself (unless using the -d option)

```
# ll lpi/
-rw-r--r-- 1 root root 21 Feb 28 17:05 1
-rw-r--r-- 1 root root  6 Feb 28 17:04 2
-rw-r--r-- 1 root root 24 Feb 28 16:02 3

# ls -ld lpi/
drwxr-xr-x 2 root root 4096 Mar  1 17:05 lpi/
```

cp

- Copies fileA to fileB
- Overwrites target if it already exists (as long as it's writable). In this case a new timestamp is created and permissions will be of the user making the copy

man cp

```
CP

NAME
       cp - copy files and directories

SYNOPSIS
       cp [OPTION]... [-T] SOURCE DEST
       cp [OPTION]... SOURCE... DIRECTORY
       cp [OPTION]... -t DIRECTORY SOURCE...

DESCRIPTION
       Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

USAGE
      -a, --archive
              same as -dpR
      --backup[=CONTROL]
              make a backup of each existing destination file
```

```
      -b     like --backup but does not accept an argument
       -d     same as --no-dereference --preserve=link
      -f, --force
              if an existing destination file cannot be opened, remove it and try again
       -i, --interactive
              prompt before overwrite
      -H     follow command-line symbolic links
      -l, --link
              link files instead of copying
      -p     same as --preserve=mode,ownership,timestamps
       --no-preserve=ATTR_LIST
              don't preserve the specified attributes
      -R, -r, --recursive
              copy directories recursively
       -s, --symbolic-link
              make symbolic links instead of copying (deprecated for 'ln')
      -u, --update
              copy  only when the SOURCE file is newer than the destination file or when the
destina-
              tion file is missing
      -v, --verbose
              explain what is being done
```

- Examples:

```
# Copies file to local directory
$ cp /etc/apt/sources.list .

# Copies files log1, log2, log3, log4, log5, log6 and log7 to home directory
$ cp log1 log2 log[34567] ~
```

mkdir

- Creates one or more directories

man mkdir

```
MKDIR

NAME
       mkdir - make directories

SYNOPSIS
       mkdir [OPTION] DIRECTORY...

DESCRIPTION
       Create the DIRECTORY(ies), if they do not already exist.

USAGE
       -m, --mode=MODE
              set permission mode (as in chmod), not rwxrwxrwx - umask
      -p, --parents
              no error if existing, make parent directories as needed
      -v, --verbose
              print a message for each created directory
```

-Example. Creates directories recursively

```
$ mkdir /dir1/dir2/dir3
```

mv

- Moves or renames files and directories
- When moving to another filesystem, a copy of the files will be made and then the original will be deleted
- Does not overwrite file if target already exists

man mv

```
MV

NAME
       mv - move (rename) files

SYNOPSIS
       mv [OPTION]... [-T] SOURCE DEST
       mv [OPTION]... SOURCE... DIRECTORY
       mv [OPTION]... -t DIRECTORY SOURCE...

DESCRIPTION
       Rename SOURCE to DEST, or move SOURCE(s) to DIRECTORY.

USAGE
       --backup[=CONTROL]
              make a backup of each existing destination file
      -f, --force
              do not prompt before overwriting
      -i, --interactive
              prompt before overwrite
      -u, --update
              move  only when the SOURCE file is newer than the destination file or when the
destina-
              tion file is missing
      -v, --verbose
              explain what is being done
```

rm

- Removes one or more files from the system

man rm

```
RM

NAME
       rm - remove files or directories

SYNOPSIS
       rm [OPTION]... FILE...

DESCRIPTION
       rm removes each specified file.

USAGE
       -d, --directory
              unlink FILE, even if it is a non-empty directory (super-user only; this works
only  if
              your system supports ‘unlink’ for nonempty directories)
      -f, --force
              ignore nonexistent files, never prompt
      -i, --interactive
              prompt before any removal
       -r, -R, --recursive
              remove directories and their contents recursively
      -v, --verbose
              explain what is being done
```

rmdir

- Remove empty directories
- Also has a '-p' options for removing parents

man rmdir

```
RMDIR

NAME
       rmdir - remove empty directories

SYNOPSIS
       rmdir [OPTION]... DIRECTORY...

DESCRIPTION
       Remove the DIRECTORY(ies), if they are empty.

USAGE
       -p, --parents
              Remove  DIRECTORY and its ancestors.  E.g., 'rmdir -p a/b/c' is similar to 'rmdir
a/b/c a/b a'.
```

Note: 'rm -R' removes directories recursively even if they are not empty

touch

- Changes the access and/or modification time in a file

man touch

```
TOUCH

NAME
       touch - change file timestamps

SYNOPSIS
       touch [OPTION]... FILE...

DESCRIPTION
       Update the access and modification times of each FILE to the current time.

USAGE
       -a     change only the access time
      -c, --no-create
              do not create any files
      -d, --date=STRING
              parse STRING and use it instead of current time
      -m     change only the modification time
      -t STAMP
              use [[CC]YY]MMDDhhmm[.ss] instead of current time
```

Example:

```
# Creates a new file
$ touch file
```

File-Naming Wildcards And Globbing

- Used to match a part of a filename
- Wildcards are expanded by shell and not the commands. Shell first expand all the wildcards and then passes the result to the command
- Using wildcards to identify multiple files is called globbing (file glob)
- (!) in a position other than the first matches itself
- manpage 7 of glob has more information

Wildcards

Wildcard	Description	Example
*	Matches a multiple chars or none	file* = file, file1, file123
?	Matches a single char	file? = file1, file2 but not file or file123
[characters]	Matches a single char within the characters listed	[123] = file1, file2, file3 but not file 4 or file 123
[!characters]	Matches a single char not listed in the wildcards	[123] = file4, file5 but not file1, file2 or file456
[a-z]	Matches a single char within the range of characters listed	[1-3] = file1, file2, file3 but not file 4 or file 123 [a-zA-Z] = filea, filez, fileD, fileF, etc...
[!a-z]	Matches a single char not in the range of characters listed	[!1-3] = file4, file5 but not file1, file2 or file456
{sting1, string2, string3}	Adds string to the end of a file	touch file{1,2,3} = file1, file2 and file3

- Example

```
$ ls /et*/*/*.log
# The first * matches /etc folder, the second * any folder within /etc, and the third * matches
all log files
```

Touching Files

- The 'touch' cmmand can be used with a few interesting options
  . (-c) - Does not create a file, only updates modification timestamp
  . (-t) - Updates a file modification timestamp using MMDDhhmm format (hour and minute are optional)
  . (-d) - Updates a file modification timestamp using a extremely flexible format
  . (-a) - Updates access time
- The options '-l' and '-u' of 'ls' can be used to display modification (same as 'date -r') and access time

```
# modification
$ ls -l 1
-rw-r--r-- 1 root root 21 Mar  1 17:23 1
$ date -r 1
Sun Mar  1 17:23:01 EST 2009

# access
$ ls -lu 1
-rw-r--r-- 1 root root 21 Mar  1 17:27 1
```

Finding Files

- Useful options
  . (-path) - Similar to '-name' however it user absolute path
  . (-type) - Finds file types (-d for directories, -f for regular files, -l for sym links, etc...)
  . (-empty) - Finds empty files (same as '-size 0')
  . (-mtime and -atime) - Finds based on modification time and access time
  . (-exec) - Executes a command based on results. Must be terminated by a escaped semi-colon to shell interpreting first

```
$ ll
-rw-rw-r-- 1 user1 user1 0 Mar  1 17:40 1.htm
-rw-rw-r-- 1 user1 user1 0 Mar  1 17:40 2.htm
-rw-rw-r-- 1 user1 user1 0 Mar  1 17:40 3.htm

$ find . -name '*.htm' -exec mv '{}' '{}l' \;

$ ll
-rw-rw-r-- 1 user1 user1 0 Mar  1 17:40 1.html
-rw-rw-r-- 1 user1 user1 0 Mar  1 17:40 2.html
-rw-rw-r-- 1 user1 user1 0 Mar  1 17:40 3.html
```

Objective 4: Use Stream, Pipes and Redirects

- Everything on Linux is a file. All the devices (HD, partitions, mouse, audio, etc..) are mapped into a file making easier to interact with the device
- Each device receives a device file (number) and a device driver (which is associated to the device file)

```
$ lss /dev/sda*
brw-rw---- 1 root disk 8, 0 2009-02-01 17:40 /dev/sda
brw-rw---- 1 root disk 8, 1 2009-02-01 17:41 /dev/sda1
brw-rw---- 1 root disk 8, 2 2009-02-01 17:40 /dev/sda2
brw-rw---- 1 root disk 8, 5 2009-02-01 17:40 /dev/sda5
brw-rw---- 1 root disk 8, 6 2009-02-01 17:41 /dev/sda6
```

Standard I/O An Default File Descriptors

- The ability of to shell to control input, output and error information:
  . Standard Input (stdin) - Information typed on the keyboard. Also known as *file descriptor 0*
  . Standard output (stdout) - What's shown on the terminal. Also known as *file descriptor 1*
  . Standard error (stderr) - Information related to errors that is also shown on the terminal. Also known as *file descriptor 2*

Pipe

- There's no difference for a program between reading stdin from you or from a file or another program. This can be accomplished by the use on pipe “l”, which joins two or more commands together

```
$ ps x | grep nm-applet
```

- Pipelines does not redirect stderr alone

Redirection

- Pipe is a form of redirection, but standard I/Os can also be redirected to a file
  . > - Writes data to the a file. It creates a file if was not created before and will overwrite everything on it
  . >> - Will append information to the end of the file. It also creates a file if was not created before but it will not overwrite it
  . < - Writes standard input to a program from a file
- The overwrite option can be disabled with the shell built-in 'set -o noclobber'. When using noclobber files can still be overwritten with >| (like 'echo “” >| file1')
- You can use “&” to send both stdout and stderr to a file (&> or &>>)
- This sends an email with a body from a file. Usually the mail program would prompt use to type a msg:

```
$ mail -s "inode list" jdean < message.txt
```

- A here-document can be used to determine the end of an input (<<[word])

```
$ cat <<EOF > file1
> 1
> 2
> EOF
$ cat file1
1
2
```

- Another option to use with a here-document '<<- ' to remove leading tabs

Note: Shell reads the and creates the output redirectors first and them the command. If you tried something like the example bellow, it will make the file completely blank:

```
$ cat test-aa
1  a
2  b
3  s
$ grep 1 test-aa > test-aa
$ cat test-aa
```

Standard I/O Redirections

Redirection Function	Syntax for bash
Writes stdout to file	>, 1>
Writes stderr to file	2>
Writes stdout and stderr to file	cmd > file 2>&1
sdout to file1 and sterr to file2	cmd > file1 2> file2
stdin from file	<
Append stdout to file	>>, 1>>
Append stderr to file	2>>
Append both stdou and stderr to file	cmd >> file 2>&1
Pipe stdout	|
Pipe stdout and stderr from cmd1 to cmd2	cmd1 2>&1 | cmd2

Using find and xargs

- find has an option that separates file names with a null character instead of a newline ('find -print 0'). Xargs can read from this output by using the '-0' option
- Other options:
  . (-maxdepth) - Determines how far down the folder the search should go
  . (-mindepth) - Does not search at levels less than the specified

Using The Tee Command

- Writes to a file and displays the output on the terminal as well
- The option '-a' can be used to append instead of overwrite

```
TEE

NAME
       tee - read from standard input and write to standard output and files

SYNOPSIS
       tee [OPTION]... [FILE]...

DESCRIPTION
       Copy standard input to each FILE, and also to standard output.

USAGE
       -a, --append
              append to the given FILEs, do not overwrite
      -i, --ignore-interrupts
              ignore interrupt signals
```

- First example writes all output to file1, while second example writes output of cmd1 to file2 and all outputs to file1

```
$ cmd1 | cmd2 | cmd3 > file1

$ cmd1 | tee file2 | cmd2 | cm3 > file1
```

Objective 5: Create, Monitor and Kill Process

Processes

- Every running program (command, application or script) is a process (process is a running instance of a program)
- Bash is a process, and every command executed on the shell is also a process (also called child process or subprocess)
- Process attribute:
  . Lifetime - Processes have execute their commands in different time frames (eg: Firefox and ls). When a process finished is also said that it died (hence the kill command)
  . Process ID (PID) - An unique integer number assigned to the process when it starts
  . User ID (UID) and Group ID (GID) - Associated to the user's privilege who started the process
  . Parent Process - Process who initiated the process in question (eg: initd, or PID 1, is the parent process of all processes)
  . Parent process ID (parent PID) - PID of the process who initiated the process in question
  . Environment - Each process holds a list of variables and its values so it can use. This list is known as environment variables. Child process inherit variable from its parent process unless told otherwise
  . Current working directory - Default directory associated with process. The process will read and write file in this directory (unless specified elsewhere)

Process Monitoring

ps (process status)

- Takes a snapshot of all processes that are currently running

```
PS

NAME
       ps - report a snapshot of the current processes.

SYNOPSIS
       ps [options]

DESCRIPTION
       ps displays information about a selection of the active processes.

SIMPLE PROCESS SELECTION
       -A              Select all processes. Identical to -e.

       -N              Select all processes except those that fulfill thE specified conditions.
(negates the selection) Identical to --deselect.

       T               Select all processes associated with this terminal. Identical to the t
option without any argument.

       -a              Select all processes except session leaders

       a               All processes with controlling terminal

       -d              Select all processes except session leaders.

       x               All processes without a controlling terminal

       -C cmdlist      Select by command name. This selects the processes whose executable name
is given in cmdlist.

       -G grplist      Select by real group ID (RGID) or name. This selects the processes whose
real group name or ID is in the grplist list. The real group ID identifies the group of the
user who created the process, see getgid(2).

       U userlist      Select by effective user ID (EUID) or name.This selects the processes
whose effective user name or ID is in userlist. The effective user ID describes the user whose
file access permissions are used by the process (see geteuid(2)). Identical to -u and --user.
```

```
       -U userlist     select by real user ID (RUID) or name. It selects the processes whose
real user name or ID is in the userlist list. The real user ID identifies the user who created
the process, see getuid(2).

       -g grplist      Select by session OR by effective group name. Selection by session is
specified by many standards, but selection by effective group is the logical behavior that
several other operating systems use. This ps will select by session when the list is completely
numeric (as sessions are). Group ID numbers will work only when some group names are also
specified. See the -s and --group options.

       p pidlist       Select by process ID. Identical to -p and --pid.

       -t ttylist      Select by tty. This selects the processes associated with the terminals
given in ttylist. Terminals (ttys, or screens for text output) can be specified in several
forms: /dev/ttyS1, ttyS1, S1. A plain "-" may be used to select processes not attached to any
terminal.

       -u userlist     Select by effective user ID (EUID) or name. This selects the processes
whose effective user name or ID is in userlist. The effective user ID describes the user whose
file access permissions are used by the process (see geteuid(2)). Identical to U and --user.

       --ppid pidlist  Select by parent process ID. This selects the processes with a parent
process ID in pidlist. That is, it selects processes that are children of those listed in
pidlist.
```

- ps can understand options into three different forms:
  . Unix98 options - May be grouped and must be preceded by a dash
  . BSD options - These may be grouped and must not be used with a dash
  . GNU long options - Preceded by two bashes
- Other useful options are '-j' for jobs and '--forest'

```
$ ps -j --forest
  PID  PGID   SID TTY          TIME CMD
16902 16902 11526 pts/2    00:00:00 bash
20016 20016 11526 pts/2    00:00:00  \_ sleep
20017 20017 11526 pts/2    00:00:00  \_ sleep
20018 20018 11526 pts/2    00:00:00  \_ ps
```

pstree

- Displays a hierarchical list of processes in a tree format

```
PSTREE

NAME
       pstree - display a tree of processes

SYNOPSIS
       pstree [-a] [-c] [-h|-Hpid] [-l] [-n] [-p] [-u] [-Z] [-A|-G|-U] [pid|user]
       pstree -V

DESCRIPTION
       pstree  shows  running  processes  as a tree. The tree is rooted at either pid or init
if pid is omitted.

OPTIONS
       -a     Show  command line arguments to launch the process
       -A     Use ASCII characters to draw the tree.
      -G     Use VT100 line drawing characters.
      -h     Highlight the current process and its ancestors. This is a no-op if the terminal
doesn't support highlighting or  if  neither  the  current process nor any of its ancestors are
in the subtree being shown.
      -H     Like -h, but highlight the specified process instead. Unlike with -h, pstree fails
when using -H if highlighting is not available.
      -n     Sort processes with the same ancestor by PID instead of by name. (Numeric sort.)
```

```
      -p     Show PIDs. PIDs are shown as decimal numbers in parentheses after each process
name. -p implicitly disables compaction.
```

- Example:

```
$ pstree
init─┬─NetworkManager───3*[{NetworkManager}]
     ├─NetworkManagerD
     ├─acpid
     ├─atd
     ├─avahi-daemon───avahi-daemon
     ├─bonobo-activati───{bonobo-activati}
     ├─console-kit-dae───61*[{console-kit-dae}]
     ├─cron
     ├─cupsd
     ├─2*[dbus-daemon]
     ├─dd
     ├─deskbar-applet───2*[{deskbar-applet}]
     ├─dhcdbd───dhclient
     ├─epiphany-browse───4*[{epiphany-browse}]
     ├─evolution-data-───{evolution-data-}
     ├─evolution-excha───{evolution-excha}
     ├─gconfd-2
     ├─gdm───gdm─┬─Xorg
     │           └─x-session-manag─┬─bluetooth-apple
     │                             ├─compiz─┬─compiz.real
     │                             │        └─emerald
     │                             ├─glipper
     │                             ├─gnome-panel
     │                             ├─nautilus
     │                             ├─python
     │                             ├─ssh-agent
     │                             ├─trackerd───2*[{trackerd}]
     │                             ├─update-notifier
     │                             ├─vino-session
     │                             └─{x-session-manag}
     ├─6*[getty]
     ├─gnome-keyring-d
     ├─gnome-power-man
     ├─gnome-screensav
     ├─gnome-settings-───{gnome-settings-}
     ├─gnome-terminal─┬─bash───pstree
     │                ├─gnome-pty-helpe
     │                └─{gnome-terminal}
     ├─gnome-vfs-daemo───{gnome-vfs-daemo}
     ├─gnome-volume-ma
     ├─hald───hald-runner─┬─hald-addon-acpi
     │                    ├─hald-addon-cpuf
     │                    ├─4*[hald-addon-keyb]
     │                    └─hald-addon-stor
     ├─hcid───2*[bluetoothd-serv]
     ├─klogd
     ├─mapping-daemon
     ├─mixer_applet2───{mixer_applet2}
     ├─notification-da
     ├─python───{python}
     ├─soffice───soffice.bin───5*[{soffice.bin}]
     ├─ssh
     ├─syslogd
     ├─system-tools-ba───dbus-daemon
     ├─totem───6*[{totem}]
     ├─trashapplet
     ├─tsclient───vncviewer
     ├─udevd
     └─wpa_supplicant
```

top

- Similar to ps, however information shown keeps getting refreshed every 3 seconds until user quits the program
- Displays a header with uptime, load, cpu, memory, etc...
- top must be able to understand how to control the terminal. It gets information on the terminal from $TERM. If the variable is not set top may not open

```
TOP

NAME
       top - display Linux tasks

SYNOPSIS
       top -hv | -bcHisS -d delay -n iterations -p pid [, pid ...]

       The traditional switches '-' and whitespace are optional.

DESCRIPTION
       The  top program provides a dynamic real-time view of a running system.  It can display
system
       summary information as well as a list of tasks currently being managed by  the  Linux
kernel.
       The  types  of  system  summary information shown and the types, order and size of
information
       displayed for tasks are all user configurable and that configuration can  be  made
persistent
       across restarts.

1. COMMAND-LINE Options
        The typically mandatory switches ('-') and even whitespace are completely optional.

       -b : Batch mode operation
            Starts  top  in  'Batch mode', which could be useful for sending output from top to
other programs or to a file.
       -c : Command line/Program name toggle
            Displays command line instead of command name
       -d : Delay time interval as:  -d ss.tt (seconds.tenths)
            Specifies  the  delay  between  screen  updates, and overrides the corresponding
value in one's personal configuration file or the startup default.
       -i : Idle Processes toggle
            Starts top with the last remembered 'i' state reversed.  When this toggle is  Off,
tasks that are idled or zombied will not be displayed.
      -n : Number of iterations limit as:  -n number
            Specifies  the maximum number of iterations, or frames, top should produce before
ending.
       -U : Monitor by user as:  -U somebody
            Monitor only processes with a UID or user name matching that given.  This  matches
real, effective, saved, and filesystem UIDs.
      -p : Monitor PIDs as:  -pN1 -pN2 ...  or  -pN1, N2 [,...]
            Monitor  only  processes  with  specified process IDs.  This option can be given up
to 20 times, or you can provide a comma delimited list with up to 20  pids.
      -s : Secure mode operation
            Starts  top  with  secure mode forced, even for root.  This mode is far better
controlled through the system configuration file (see topic 5. FILES).
      -S : Cumulative time mode toggle
            Starts top with the last remembered 'S' state reversed.  When 'Cumulative mode'  is
On, each  process  is  listed with the cpu time that it and its dead children have used.

INTERACTION
       ^L     Redraw the screen.
       <?> or <h> :Help
             There are two help levels available.  The first will provide a reminder  of  all
the basic interactive commands.
      <k> :Kill_a_task
             You will be prompted for a PID and then the signal to send.
      n<nprocs>
             Only display up to <nprocs> processes.  0 is treated as  infinity.
       q      Quit.
```

```
       s<delay>
              Set the delay between updates to <delay> seconds.
       <r> :Renice_a_Task
              You will be prompted for a PID and then the value to nice it to.
       <d> or <s> :Change_Delay_Time_interval
              You  will  be  prompted  to  enter  the  delay time, in seconds, between display
updates.
```

Signaling Active Processes

- A asynchronous way of communicating with processes
- Each process running listen to signals, sent either by the kernel or by users
- Signals have a name and a number. Name are for human reference and number is what is sent to the process
- There are more than 32 signals available in Linux

Frequently Used Interactive Signals

Signal Name	Number	Meaning and Use
HUP	1	Hang up. Can be used to log out or disconnect a modem, or to make daemon reread conf file
INT	2	Interrupt, stope running (same as Ctrl+c)
KILL	9	Kill, stop unconditionally and imediatelly
TERM	15	Terminate nicelly if possible
CONT	18	Continue execution. Used to start a process stopped by SIGTSTP or SIGSTOP. Same as 'bg'
STOP	19	Suspends execution until a CONT is received (same as Ctrl+d)
TSTP	20	Stop executing, ready to continue (same as Ctrl+z)

Notes:
- Signal names will usually be referred with a preceding SIG (SIGHUP, SIGINT)
- Kill command is both a shell command and a binary
- The command 'nohup' make a process immune to HUP

kill

- Sends a signal to a process

```
KILL

NAME
       kill - send a signal to a process

SYNOPSIS
       kill [ -signal | -s signal ] pid ...
       kill [ -L | -V, --version ]
       kill -l  [ signal ]

DESCRIPTION
       The  default  signal  for  kill is TERM. Use -l or -L to list available signals.

EXAMPLES
       kill -9 -1
              Kill all processes you can kill.
      kill -l 11
              Translate number 11 into a signal name.
      kill -L
              List the available signal choices in a nice table.
      kill 123 543 2341 3453
              Send the default signal, SIGTERM, to all those processes.
```

- Examples:
  . All these will accomplish the same:

```
$ kill 1000 1001
$ kill -15 1000 1001
$ kill -SIGTERM1000 1001
$ kill -sigterm 1000 1001
$ kill -TERM 1000 1001
$ kill -s 15 1000 1001
$ kill -s SIGTERM 1000 1001
```

  . Some daemons will respond to HUP, HTTPD is one of them. This can be useful if you made changes to the configuration file and want the process to re-read it.

```
kill -HUP `cat /var/run/httpd.pid
```

Terminating Processes

- Commonly used only on process that are not responding. Usually you should use -15 (SIGTERM) and then escalate to -9 (SIGKILL) if necessary
- Processes terminated will in most cases kill it's child processes
- A process displayed with a status of zombie is a process that are stuck in the process of terminating. These processes can not be killed and could represent a system bug if they keep reappearing

Shell Job Control

- Allows you to place processes started in a shell in the background so you can have access to the shell again
- The terminal which the jobs was created is called controlling terminal
- Controlling terminal has no way of sending stdin to background jobs
- If you run a process from inside the shell, it will wait until you are done with the process and provide stdout and stderr to the screen. You can also use a "&" after the process so shell doesn't wait for it to be finished. This will place the process in the background (bg).
- Other method to put the process in the background is by starting the process and them hitting Ctrl+z. This will pause the process, show a job number for the process and provide you with a prompt. Here you can them use "bg" to make the process run in the background. Example:

```
$ glxgears
# hit Ctrl+z
[1]+  Stopped                 glxgears
$ bg %glxgears
[1]+ glxgears &
```

- The command "jobs" will show all processes that are running in the background:

```
$ jobs
[1]+  Running                 glxgears &
```

- Jobs in the background can be called upon by using fg %<job number> or by fg %<process name> (as long as there is only one process with the same name
- Processes can also be terminated with "kill %<job number>"

bg

- Places a job in the background

```
BG

NAME
       bg - run jobs in the background

SYNOPSIS
       bg [job_id ...]
```

```
DESCRIPTION
       If job control is enabled (see the description of set -m), the bg utility shall resume
suspended jobs from the current environment (see Shell Execution Environment ) by running them
as background jobs. If the job specified by job_id is already a running background job, the bg
utility shall have no effect and shall exit successfully.

OPTIONS
       None.

OPERANDS
       job_id
           Specify the job to be resumed as a background job. If no job_id operand is given, the
most recently suspended job shall be used
```

fg
- Places a job in the foreground

```
FG

NAME
       fg - move jobs to the foreground

SYNOPSIS
       fg [ options ] [job ...]

DESCRIPTION
       fg places the given jobs into the foreground in sequence and sends them a CONT signal to
start each running.

       If job is omitted, the most recently started or stopped background job is moved to the
foreground.

USAGE
       Each job can be specified as one of the following:

       number
           number refers to a process id.
       -number
           number refers to a process group id.
       %number
           number refer to a job number.
       %string
           Refers to a job whose name begins with string.
       %?string
           Refers to a job whose name contains string.
       %+ or %%
           Refers to the current job.
       %-
           Refers to the previous job.
```

jobs
- Lists the active jobs for the shell in question
- Common used options:
  . (-l) - Prints PID of the related process
  . (-p) - Prints PID of the process group leader for each job

nohup
- Used to start a command that will ignore hangup signals and write stdout and stderr to a file (defaults are nohup.out or ~/nohup.out)

Objective 6: Modify Process Execution Priority

- There are a total of 40 priorities available in Linux
- The higher the priority, more CPU time the kernel offers to the process
- By default user processes are created with a nice number of 0
- Any user can start a process with a positive nice number, but only root can lower a process's nice number
- The PRI column (top or ps -l) shows the priority of a process

Nice

- One of the parameters used by the kernel to assign process priority to processes. It causes a program to execute with less priority than other processes, thus being nice
- A positive value means the process is being nicer, thus running with less priority
- Numbers can go from -20 to +19
- Process started with nice default to a nice number of +10
- You cannot use a command list or pipeline with nice

nice

- Used to alter another command nice number at start up

```
NICE

NAME
       nice - run a program with modified scheduling priority

SYNOPSIS
       nice [OPTION] [COMMAND [ARG]...]

DESCRIPTION
       Run  COMMAND  with an adjusted niceness, which affects process schedul-
       ing.  With no COMMAND, print the current  niceness.   Nicenesses  range
       from -20 (most favorable scheduling) to 19 (least favorable).

       -n, --adjustment=N
              add integer N to the niceness (default 10)
```

Example:

```
# These two forms execute the same thing, they add a value of -10 to a process
$ nice --10 gedit &
$ nice -n -10 gedit &
```

renice

- Used to change the nice value on process that are already running

```
NAME
     renice - alter priority of running processes

SYNOPSIS
     renice priority [[-p] pid ...] [[-g] pgrp ...] [[-u] user ...]

DESCRIPTION
     Renice alters the scheduling priority of one or more running processes.

USAGE
     -g      Force who parameters to be interpreted as process group ID's. Changes all process
running by that group.
    -u      Force the who parameters to be interpreted as user names. Changes all process
running by that user.
```

Notes:

- Renice does not use "--" like nice

- Renice can also be used under top with the "r" interaction command

Objective 7: Search Text Files Using Regular Expressions

- LPI-1 only goes into simple regular expressions using “sed” and “grep”
- Regular expressions are also known as regexp or regex
- Has two forms of syntax used by grep: basic and extended

Regular Expression Syntax

- Regular expressions have been added to different tools over the years without following rules or guidelines. After a while, in a attempt to make them more consistent, a definitions was implemented
- They are patterns that can be divided into two groups:
  . Literal - A literal is any character we use in a search or matching expression, for example, to find ind in windows the ind is a literal string - each character plays a part in the search, it is literally the string we want to find.
  . Metacharacter - A metacharacter is one or more special characters that have a unique meaning and are NOT used as literals in the search expression, for example, the character ^ (circumflex or caret) is a metacharacter.
  . Escape Sequence - A escape sequence is a way of indicating that we want to use one of our metacharacters as a literal. In a regular expression a escape sequence involves placing the metacharacter \ (backslash) in front of the metacharacter that we want to use as a literal, for example, if we want to find ^ind in w^indow then we use the search string \^ind and if we want to find \\file in the string c:\\file then we would need to use the search string \\\\file (each \ we want to search for as literal is preceded by an escape sequence \).
  . Target String - Describes the string that we will be searching, that is, the string in which we want to find our match or search pattern.

Text taken from http://www.zytrax.com/tech/web/regex.htm#intro

Position Anchors

Metacharacter	Description
^	Match at the beginning of a line
$	Match at the end of a line
\<, \>	Match word boundaries. The \ is to allow the use of a special character

- Used to specify a position of one or more characters in relation to the entire line of text

Characters Set (regex)

Metacharacter	Description
[abc]	Match a single character within the specified list
[a-z]	Match a single character within the range specified by the list
[^abc]	Match a single character **NOT** within the specified list
[^a-z]	Match a single character **NOT** within the range specified by the list
.	Match a single character int the position ony. eg: ton. will find tons and toneau but not anton

- Matches text (either literal, metacharacters that matches individuals or multiple characters, or a combination of both)
- Does not need to be escaped, neither does special character within brackets (like [*.])
- Posix characters can also be used:
  . [:alnum:] - Alphanumeric characters
  . [:blank:] - Space and tab characters
  . [:digit:] - Same as 0-9
  . [:upper:] and [:lower:] - Upper and lower case characters

Modifiers

Basic Expression	Extended Expression	Description
*	*	Match none, single or multiple characters from the receding character (or regex)
\?	?	Match zero or one instance of the preceding regex
\+	+	Match one or more instance of the preceding regex
\{n,m\}	{n,m}	\{n\} matches n occurences; \{n,\} matches at least n occurrences; \{n,m\} matches any number of occurences from n to m
|	\	Match regex either before or after pipe
\(regex\)	(regex)	Used for grouping together patterns

- Follows a character set and indicates the number of times it should appear

grep

- Grep derivated from a command used on a old line editor called ed (global regular expression print)
- Global regular expression print was used to displays lines of a file being edited that matched a given expression

```
GREP

NAME
       grep, egrep, fgrep, rgrep - print lines matching a pattern

SYNOPSIS
       grep [options] PATTERN [FILE...]
       grep [options] [-e PATTERN | -f FILE] [FILE...]

DESCRIPTION
       grep  searches the named input FILEs (or standard input if no files are named, or the
file name - is given) for lines containing a match to the given PATTERN.  By default, grep
prints the matching lines.

OPTIONS
       -c, --count
              Suppress normal output; instead print a count of matching  lines  for  each
input  file.  With the -v, --invert-match option (see below), count non-matching lines.
       -E, --extended-regexp
              Interpret PATTERN as an extended regular expression (see below).
       -h, --no-filename
              Suppress the prefixing of  filenames  on  output  when  multiple files are
searched.
      -i, --ignore-case
              Ignore case distinctions in  both  the  PATTERN  and  the  input files.
       -n, --line-number
              Prefix each line of output with the line number within its input file.
       -v, --invert-match
              Invert the sense of matching, to select non-matching lines.
```

- Examples - Inside a folder there are 5 files (file1-5) with a single line text in each that reads “filea” to file1, “fileb” to file2, and so on. Let's say we want to read the text inside the files:

```
# list the files
$ echo file*
file1 file2 file3 file4 file5
# wrong form for grep
$ grep file* file*
file1 file2 file3 file4 file5 file1 file2 file3 file4 file5
# proper form
$ grep 'file*' file*
file1:filea
file2:fileb
file3:filec
```

```
file4:filed
file5:filee
```

sed

- Long name is “Stream Editor”
- Commonly used to automate repetitive editing tasks or to process text pipes

```
SED

NAME
       sed - stream editor for filtering and transforming text

SYNOPSIS
       sed [OPTION]... {script-only-if-no-other-script} [input-file]...

DESCRIPTION
       Sed  is a stream editor.  A stream editor is used to perform basic text transformations
on an input stream (a file or input from  a  pipeline). While  in  some  ways similar to an
editor which permits scripted edits (such as ed), sed works by making only one pass over the
input(s),  and is consequently more efficient.  But it is sed’s ability to filter text in a
pipeline which particularly distinguishes it from other  types  of editors.

USAGE
       -e script, --expression=script
              add the script to the commands to be executed
      -f script-file, --file=script-file
              add the contents of script-file to the commands to be executed

COMMANDS
       d
               delete lines
       s
               make substitutions. s/pattern/replacement/[flags]
               g - replace all instances of pattern, not just the first
               n - replace nth instance of pattern. 1 is the default
               p - print the line to the file if a successful substitution is done
               w file - print line to file if a successful substitution is done
               y - Translate characters
```

- Examples:

```
# 1- Delete lines 3 throug 5
$ sed '3,5d' file

# 2- Delete lines that start wit #
$ sed '^# ' file

# 3- Translate characters; a to x, b to y and c to z
$ sed 'y/abc/xyz' file

# 4- Writes @ to all empty lines
$ sed 's/^$/@/' file

# 5- Removes all double quotation
$ sed 's/”//g' file

# 6- Gets sed command from an external file. The file will replace the 1st and 2nd quotation marks
for single quotes on lines 1 to 2
$ cat script-file
1,5{
s/”/'/1
s/”/'/2
}
$ cat file
“”something”
“”something else”
$ sed -f script-file file
'”something'
```

```
'"something else'
#note: sed starts counting again after firs " has been substituted
```

Using Regular Expressions With Sed

- A pattern match (addressing) can also be used with sed

```
# replaces "abc" for "ABC" only on lines that have "123"
sed /123/abc/ABC/g
```

Extended Regular Expression

- Eliminates the need to escape some characters
- grep has the options 'egrep' or 'grep -e'
- sed has the option 'sed -r'

Item Match Results

- grep has the option of providing an amount of matches as well:
  . (-n) - What line numbers have a match
  . (-l) - List of lines with matches
  . (-c) 0 How many lines had a match

Regular Expressions Examples

Anchors

- Describe position information

```
# 1- Displays all lines that start with Linux
$ grep '^Linux' file

# 2- Displays all lines that end with an x
$ grep '$x' file

# 3- Displays the number of empty lines
$ grep -c '^$' file

# 4- Displays all lines containing the word null by itself
$ grep '^null$' file
```

Groups and Ranges

```
# 1- Displays all lines containning Linux, linux, turbolinux, LinuxOS
$ grep '[Ll]inux' file

# 2- Displays files that contain three consecutive digits
$ grep '[0-9][0-9][0-9]' file

# 3- Displays lines that do no begin with a numeral
$ grep '^[^0-9]' file

# 4- Displays all lines containning Linux, linux but not turbolinux, LinuxOS
$ grep '\<[Ll]inux\>' file

# 5- Matches a line with 5 or more characters
$ grep '.....' file

# 6- Displays all non blank lines
$ grep '.' file

# 7- Displays all lines that have a period
$ grep '\.' file
```

Modifiers

- Changes the meaning of other characters in a regular expression

```
# 1- Displays all lines that have ab, abc, abcc, abcccc, etc...
$ grep 'abc*' file

# 2- Displays all lines that have abc, abcc, abcccc, etc...
$ grep 'abcc*' file

# 3- All lines that have two or more numeral digits
$ grep '[0-9][0-9][0-9]*' file
     or
$ grep '[0-9]\{2,\}' file

# 4- All lines that have file, file1 or file2
$ grep 'file[12]\?' file

# 5- All lines that have one or more digits
$ grep '[0-9]\+' file

# 6- All lines that have 111, 1111 or 11111 by itself
$ grep '^1\{3,5\}$' file

# 7- All lines that contain a 3, 4 or 5 digit number
$ grep '\<[0-9]\{3,5\}\>' file

# 8- All lines that contain hapy, Happy; sad, Sad or angry, Angry
$ grep -E '[hH]appy|[sS]ad|[aA]ngry' file

# 9- All lines that have two or more instances of abc (abcabc, abcabcabc, abcabc abcabc etc...)
$ grep '\(abc\)\{2,\}' file
```

Basic Regular Expressions Patterns

```
# 1- Match any letter
$ [a-zA-Z]

# 2- Any non alphanumeric  character
$ [^0-9A-Za-z]

# 3- An uppercase character followed by zero o lower case characters
$ [A-Z][a-z]*

# 4- Matches numbers on the pattern xx-xxxx-xxx
$ [0-9]\{2}-[0-9]\{4\}-[0-9]\{3\}

# 5- Matches "$ xx.xx", whereas the chracters following $ could be a space or digits
$ \$[ 0-9]*\.[0-9]\{2\}

# 6- Matches June or Jun (? matches zero or an instance of e)
$ june\?
```

Using regular expressions as addressed in sed

```
# 1- Delete blank lines
$ /^$/d

# 2- Delete lines that doesn't have #keepme
$ /#keepme/!d

# 3- Delete lines that contain white space or tab
$ /^[ tab]*$/d
     or
$ /^[[:blank:]]*$/d

# 4- Delete lines beginning with . or #
$ /^[\.#]/d
```

```
# 5- Substitute multiple spaces for a single space
$ s/  */ /g
      or
$ s/ \{2,\}/ /g

# 6- Substitutes abc for def on lines 1 to 4
$ 1,4s/abc/def/g file

# 7- Translate characters a, b, or c (any word) on lines 11 through 20 for @
$ 11,20y/abc/@@@/
```

Objective 8: Perform Basic File Editing Operations Using VI

- vi is the most common editor used in Linux. If a system has one editor it will most likely be vi
- The original version of vi did not have syntax highlighting

Invoking Vi

- Type vi followed by a file name in a terminal window
- Lines with a "~" are not yet edited lines (empty lines with not even blank spaces)

```
$ vi file
     or
$ vi file1 file2
```

Vi Basics

- vi has two modes:
  . Command - Lets you navigate and enter commands (press Esc to change into command mode)
  . Insert - Used to enter new text (press i to change into insert mode)
- Command mode lets you use a number before the command to make the command repeat itself multiple times
- Commands are usually a set of one or two keys

vi Commands

Key Command	Description
h	Left
j	Down
k	Up
l	Right
H (Caps)	Top of the screen
L (Caps)	Bottom of the screen
G (Caps)	End of line
w	Forward one word
b	Backward one word
0 (zero)	Beginning of current line
^	Firs non-whitespace character on the current line
$	Move to the end of the current line
Ctrl-B	Move up (back) one screen
i	Insert at the current cursor position
I (Caps)	Insert at the beginning of current line
a	Append after the cursor position
A (Caps)	Append to the end of the current line
o	Start a new line after the current line
O (Caps)	Start a new line before the current line
r	Replace the character at the current position
R (Caps)	Start replacing at the current position
x	Delete character at current position
X (Caps)	Delete the character to the left of cursor
s	Delete character at current position and go into insert mode
S (Caps)	Delete the contents of the current line and go into insert mode
dX	Given a movement command (X), cut the appropriate number of characters, words or lines from the cursor
dd	Cut entire line
D (Caps)	Cut from cursor to the end of line
cX	Given a movement command (X), cut the appropriate number of characters, words or lines from the cursor and go into insert mode
cc	Cut entire line and go into insert mode
C (Caps)	Cut from cursor to the end of line and go into insert mode
yX	Given a movement command (X), copy the appropriate number of characters, words or lines from the cursor
yy or Y	Copy current line
p	Paste after cursor
P (Caps)	Paste before cursors
.	Repeat last command
u	Undo last command
/**regex**	Search forward for regex

Key Command	Descriptor
?regex	search backward for regex
n	Find next match
N (Caps)	Find previous match
:n	Go to next file (use :n! if current has unsaved chages)
:efile	Load file in place of current (force with :!efile)
:e!	Reload files from disk loosing all changes made
:rfile	Insert contents of file after cursor
:q	Quit
:wfile	Write to files (:w >>file to append and :w! to force)
:wq	Write and quit (force with :wq!)
:w!	Write file, no matter if changes were made or not
:x	Write and quit
ZZ	Write file contents and quit
:!command	Execute command in a subshell

Note: Need to know how to switch between modes and how to perform basic navigation and editing tasks

Topic 1.104

Devices, Linux Filesystems, and the Filesystem Hierarchy

Objective 1: Create Partitions and Filesystems

- Filesystem can have two different meanings:
  . The way files and directories are physically stored in a disk or medium. Filesystems can vary, and there are many supported by Linux (eg: etx2, ext3, vfat, msdos, JFS, etc...)
  . The structure and contents of a “filesystem”. A filesystem needs to be mounted for the system to be able to access it's data. In Linux, a filesystem can be mounted from a network and seem like local for a user (even / can be mounted over the network if configured properly)

Disk Drives Under Linux

- Devices that will work with Linux, including IDE, SCSI, floppy, CD-ROM, CD-Rs, ZIP and Jaz disks
- IDE are usually primary master, primary slave, secondary master and secondary slave
- SCSI offer lower CPU utilization and up to 15 devices per bus. It's however three times the cost of IDEs

Hard disk devices

- IDE
  . Primary master - /dev/hda
  . Primary slave - /dev/hdb
  . Secondary master - /dev/hdc
  . Secondary slave - /dev/hdd
- SCSI
  . First drive - /dev/sda
  . Second drive - /dev/sdb
  . Third drive - /dev/hdc

Note: A PC with a HD on primary master and a CD-ROM on secondary master would have /dev/hda and /dev/hdc

Disk partitions

- Linux by default uses the MS-DOS partitioning system (it can use others). It allows for up to 4 primary partitions. Out of the 4 primary partitions, one can be replaced with a logical partition, which allow for up to 12 new partitions, totaling 15 usable partitions (+1 which is the extended partition, or the container of the logical partition, and can not have any data)
- Primary partitions (one can be active which will be used by BIOS to boot) - /dev/hda1, /dev/hda2, /dev/hda3, etc...
- Extended partition has same numbering as primary, but there can only be one per disk:
  . /dev/hda1 (primary)
  . /dev/hda2 (extended)
- Logical partitions are numbered from 5 to 16
  . /dev/hda1 (primary)
  . /dev/hda2 (extended)
  . /dev/hda5 (logical)
  . /dev/hda6 (logical)
  . /dev/hda7 (logical)

The root filesystem and mount points

- Not all directories need to be on the same partition as /. As a matter of fact, mounting the different directories from different partitions into / helps avoid a problem in the system due to space
- Guideline to mounting partitions:
  . / - First directory mounted. The following directories are required for the boot process, thus must be on the same partition
    /bin and /sbin - System binary programs
    /dev - Device files
    /etc - Configuration files (some are required on boot)

/lib - Shared libraries

. Following directories are part of the single / partition and are better placed on a different partition than /

/boot - Contains static files used by the boot loader, including the kernel image. On systems used for kernel development this folder can fill up influencing the / directory if on the same partition

/home - User files and the bigger partition in the system

/tmp - Usually a separate partition to prevent temporary files from filling /

/var - Log files

/usr - Extra utilities that don't fit under /bin (games, printers, softwares, libraries, shared files, etc...) . Package manager installs software here. Folder can sometimes be static, allowing some users to mount this directory as read only to avoid corruption

An Example of Partition Scheme

Partition	Type	Mount	Size
/dev/hda1	Primary	/boot	100 MB
/dev/hda2	Primary	/	500 MB
/dev/hda3	Extended	-	-
/dev/hda5	Logical	/usr	4 GB
/dev/hda6	Logical	/var	2 GB
/dev/hda7	Logical	/opt	1 GB
/dev/hda8	Logical	/tmp	500 MB
/dev/hda4	Primary	(swap partition)	1 GB
/dev/hdb1	Primary	/home	60 GB

- Resizing partitions is possible but risky

Managing Partitions

- There are mainly to tools used for partitioning in Linux, fdisk and cfdisk (uses a curses, similar to aptitude, display and is not covered by LPI)
- To create and delete partitions it's needed to edit the partition table (with one of the programs above). Data is not touched at this point and root access is needed

```
FDISK

NAME
       fdisk - Partition table manipulator for Linux

SYNOPSIS
       fdisk [-u] [-b sectorsize] [-C cyls] [-H heads] [-S sects] device
      fdisk -l [-u] [device ...]
      fdisk -s partition ...
      fdisk -v

DESCRIPTION
       fdisk  (in the first form of invocation) is a menu driven program for creation and
manipulation of partition tables.  It understands DOS type partition tables and BSD or SUN type
disklabels.

       fdisk doesn't understand GUID Partition Table (GPT) and it is not designed for  large
partitions.  In  particular  case  use  more  advanced  GNU parted(8).

OPTIONS
       -l     List  the  partition tables for the specified devices and then exit.  If no
devices are given, those mentioned in /proc/partitions (if that
              exists) are used.
       -u     When listing partition tables, give sizes in sectors instead of cylinders.
      -s partition
              The size of the partition (in blocks) is printed on the standard output.
      -v     Print version number of fdisk program and exit.
```

```
OPTION 2
      d
            Deletes partition
      a
            Boot flag on/off
      m
            Displays help menu
      n
            Add a new partition. Prompt follows as primary or extended; partition number; first
cylinder; last cylinder. Swap partitions can be created with the t option.
      p
            Displays partition table in the memory (not yet applied)
      q
            Quit without saving
      t
            Changes partition system ID (a hex number that indicates filesystem. e.g.: 83 for
ext2)
      v
            Verifies setup
      w
            Write and exit
```

- Note that fdisk can use different value types when referring to partition size
  . In cylinders (+*n*) - +64
  . In human readable size (+*n*M) - +2048M
- Neither LILO or GRUB use the bootable flag. This is used by the BIOS to identify the 1st Stage boot loader

Filesystem Types

- Some filesystems provide journaling (ext3, reiserfs), which allows for a faster and better recovery after system crash
- Types:
  . ext2 - It's possible to convert it to ext3 (and then convert back to ext2)
  . ext3
  . ReiserFS - Has good performance, particularly for large number of small files. Scales well and provides journaling
  . XFS - High in transit data caching. Use of a UPS is strongly advised
  . vfat - Unzipping or untarring files in vfat will break links a loose all permissions
- Displaying supported filesystem types

```
$ ll /sbin/mk* | awk '{ print $8 $9 $10 }'
/sbin/mkdosfs*
/sbin/mke2fs*
/sbin/mkfs*
/sbin/mkfs.bfs*
/sbin/mkfs.cramfs*
/sbin/mkfs.ext2*
/sbin/mkfs.ext3*
/sbin/mkfs.minix*
/sbin/mkfs.msdos->mkdosfs*
/sbin/mkfs.ntfs->/usr/sbin/mkntfs*
/sbin/mkfs.reiserfs*
/sbin/mkfs.vfat->mkdosfs*
/sbin/mkreiserfs*
/sbin/mkswap*
```

Creating filesystems

- After partitioning, it's needed to add a filesystem to the partition to be able to use it. This can be achieved with mkfs, which is a front-end program for mkfs.*fstype* (like mkfs.ext2 and mkfs.msdos, which are linked to mke2fs and mkdosfs):
  . mkfs - mkfs.ext2 = mke2fs
  | \- mkfs.ext3 = mkfs.ext2 -j # j for journaling
  |----- mkfs.msdos = mkedosfs

```
MKFS

NAME
       mkfs - build a Linux file system

SYNOPSIS
       mkfs [ -V ] [ -t fstype ] [ fs-options ] filesys [ blocks ]

DESCRIPTION
       mkfs  is  used to build a Linux file system on a device, usually a hard disk partition.
filesys is either the device  name  (e.g.   /dev/hda1, /dev/sdb2).   blocks  is  the  number of
blocks to be used for the file system.

OPTIONS
       -V     Produce verbose output, including all file system-specific  com-
              mands  that are executed.  Specifying this option more than once
              inhibits execution of any file system-specific  commands.   This
              is really only useful for testing.

       -t fstype
              Specifies  the  type  of file system to be built.  If not speci-
              fied, the default file system type (currently ext2) is used.

       fs-options
              File system-specific options to be passed to the real file  sys-
              tem builder.  Although not guaranteed, the following options are
              supported by most file system builders.

       -c     Check the device for bad blocks before building the file system.

       -l filename
              Read the bad blocks list from filename

       -v     Produce verbose output.
```

- Example. Make a new ext2 filesystems labeled “tuxie” on hda3 checking for bad blocks and adding extra verbose

```
$ mkfs -t ext2 -L tuxie -cv /dev/hda3
```

- Some options are the same when using the mkfs command, like '-type'

Creating Ext[2\3} Filesystems
- Allows to add a label for easy identification of partition using the '-L' option
- Label can also be displayed using 'e2label' command
- Adding journaling to an existing ext2 filesystem is as easy as using the option '-j'

Creating ReiserFS Filesystem
- Labels can be added with the '-l' option or using 'reiserfstune' command

Creating XFS Filesystem
- Labels can be added with the '-L' option or using 'xfs_admin -L'

Creating Vfat Filesystem
- Labels can be added with the '-n' option and displayed with 'e2label'

Creating swap partitions (not required for LPI)
- Swap spaces are not mounted, but enabled instead (using the command 'swapon'

```
MKSWAP

NAME
       mkswap - set up a Linux swap area

SYNOPSIS
       mkswap [-c] [-vN] [-f] [-p PSZ] [-L label] device [size]

DESCRIPTION
       mkswap sets up a Linux swap area on a device or in a file.

OPTIONS
       -c     Check the device (if it is a block device) for bad blocks before creating the
swap area.  If any are found, the count is printed.
      -f     Force - go ahead even if the command is stupid.  This allows the creation of a
swap area larger than the file or partition it  resides  on.
       -p PSZ Specify the page size to use.
      -L label
              Specify a label, to allow swapon by label.  (Only for new style swap areas.)
```

Objective 2: Maintain The Integrity or Filesystems

- Many things can cause problems to a filesystem. Some common ones are:
  . Filesystem fills to capacity
  . Corruption (power failure or crash)
  . Runs out of inodes and cannot create new filesystem objects

Monitoring Free Disk Space And Inodes

- Every filesystem contain a finite number o inodes that are specified when the system is created
- It's mostly improbable that a system will run out of inodes, however if it happens, it's most likely to happen to a partition that contains small files
- If a filesystem runs out of capacity, deleting large files can resolve the problem. However, when it's the case of low number of inodes available, the filesystem either needs be recreated or a large amount of files need to to deleted

```
DF

NAME
       df - report file system disk space usage (Displays information on mounted filesystems)

SYNOPSIS
       df [OPTION]... [FILE]...

DESCRIPTION
       This  manual  page  documents  the  GNU version of df.  df displays the amount of disk
space available on the file system containing each  file name  argument.   If  no file name is
given, the space available on all currently mounted file systems is shown.

OPTIONS
      -a, --all
              include dummy file systems
      -h, --human-readable
              print sizes in human readable format (e.g., 1K 234M 2G)
      -i, --inodes
              list inode information instead of block usage
      -t, --type=TYPE
              limit listing to file systems of type TYPE
      -T, --print-type
              print file system type
      -x, --exclude-type=TYPE
              ignores specified filesystem type
```

Note: 'df' can also be used on files

Monitoring Disk Usage

- The built-in utility “du” displays information in regards to disk usage in the PWD or the given filename(s)

```
DU

NAME
       du - estimate file space usage

SYNOPSIS
       du [OPTION]... [FILE]...
       du [OPTION]... --files0-from=F

DESCRIPTION
       Summarize disk usage of each FILE, recursively for directories.

       -a, --all
              write counts for all files, not just directories
       -c, --total
              produce a grand total
      -h, --human-readable
              print sizes in human readable format (e.g., 1K 234M 2G)
      -S, --separate-dirs
              do not include size of subdirectories
      -s, --summarize
              display only a total for each argument
```

- Examples

```
# Include subdirectories and displays only the summary
$ du -s /etc

# Do not include subdirectories and display a summary
$ du -Ss /etc

# Displays a summary of all subdirectories with a human readable result
$ du -csh /etc

# Displays summary and sorts the result in order of largest to smallest
$ du -cs | sort -nr  # numeric and reversal. Sort reads the number and not the value
```

Modifying A Filesystem

- tune2fs is a utility used to modify existing ext2 and ext3 filesystem
- It can add journaling to an existing ext2 filesystem, display the amount of mounts before check is forced, assign label, etc...

```
TUNE2FS

NAME
       tune2fs - adjust tunable filesystem parameters on ext2/ext3 filesystems


DESCRIPTION
       tune2fs  allows  the  system  administrator  to  adjust various tunable
 filesystem parameters on Linux ext2/ext3 filesystems.

OPTIONS
       -c max-mount-counts
              Adjust the number of mounts after which the filesystem  will  be
 checked  by e2fsck(8).

       -C mount-count
              Set the number of times the filesystem has been mounted.
       -f     Force  the  tune2fs  operation  to  complete even in the face of
 errors.
       -i  interval-between-checks[d|m|w]
              Adjust the maximal time between two filesystem checks.  No postfix or d result in
```

```
days, m in months, and w in weeks.   A  value
of zero will disable the time-dependent checking.
       -j     Add an ext3 journal to the filesystem.
       -J journal-options
              Override the default ext3 journal  parameters.
       -l     List the contents of the filesystem superblock.

       -L volume-label
              Set  the volume label of the filesystem.
       -m reserved-blocks-percentage
              Set the percentage of the filesystem which may only be allocated
 by privileged processes.
       -r reserved-blocks-count
              Set the number of reserved filesystem blocks.
       -T time-last-checked
              Set the time the filesystem was last checked using e2fsck.
```

Checking And Repairing Filesystems

- Computers fails, and reasons could vary. If a system crashes while data is being transferred to it, the data being transferred is lost and the allocated space is marked as used. Filesystems writes are cached in memory, and a crash would result in data loss
- fsck is a front-end program for the many filesystem checking routines for the different filesystem types (fsck.ext2, which is linked to e2fsck):
  . fsck --- fsck.ext2 ---- e2fsck

```
$ ls -1i /sbin/*fsck* | sort -n
421420 /sbin/dosfsck
421420 /sbin/fsck.msdos
421420 /sbin/fsck.vfat
421494 /sbin/e2fsck
421494 /sbin/fsck.ext2
421494 /sbin/fsck.ext3
421537 /sbin/fsck.cramfs
421653 /sbin/fsck
```

- e2fsck can also check ext3 filesystems
- At boot the kernel runs fsck with the -A option (checks all filesystems in /etc/fstab*), as long as the entry does not contain the noauto option. If a significant error is found, the system goes into single-user mode so user can run fsck manually. In these cases there's not much you can do other than run fsck -y and hope for the best (unless you know the inner workings of the filesystem in details). Sometimes the problem can be beyond repair (very uncommon). Tools that can be used in cases like this are e2image, dumpe2fs and debugfs
- Part of the information that describes the filesystem is the superblock and it's saved on block 1 of the partitions. If this bock becomes corrupted the filesystem is unaccessible. By default filesystem makes copies of the superblock every 8192 blocks (1, 8193, 16385, ...) which can then be used by fsck to restore the main superblock
- Filesystems must be unmounted to be checked

** fsck runs at boot on the filesystems configured at /etc/fstab. The sixth field (dump) provides information of which filesystems should be checked and in what order. Two filesystems can have the same value and be checked simultaneously.*
. 0 - Do not check
. 1 - Check first
. 2 - Check second

```
$ cat /etc/fstab | head -n 2
LABEL=/             /           ext3    defaults        1 1
LABEL=/boot         /boot       ext3    defaults        1 2
```

```
FSCK

NAME
       fsck - check and repair a Linux file system

OPTIONS
       -t fslist
              Specifies  the  type(s)  of file system to be checked.
       -A     Walk through the /etc/fstab file and try to check all file systems in one run.
This option is typically used from the /etc/rc system  initialization file, instead of multiple
commands for checking a single file system.
       -C [  fd  ]
              Display completion/progress bars for those filesystem checkers (currently only
for ext2 and ext3) which support them.
       -N     Don't execute, just show what would be done.
```

```
E2FSCK

NAME
       e2fsck - check a Linux ext2/ext3 file system

DESCRIPTION
       e2fsck is used to check a Linux second extended file system (ext2fs).

OPTIONS
       -a     This option does the same thing as the -p option.  It is provided for backwards
compatibility only; it is  suggested  that  people  use  -p   option whenever possible.
      -b superblock
              Instead  of using the normal superblock, use an alternative superblock specified
by superblock.
       -c     This  option  causes  e2fsck  to use badblocks(8) program to do a read-only scan
of the device in order to find any bad blocks.
       -f     Force checking even if the file system seems clean.
       -l filename
              Add the block numbers listed in the file specified by filename to the list of bad
blocks.
       -n     Open the filesystem read-only, and assume an answer of 'no' to all questions.
       -p     Automatically  repair  ("preen")  the  file  system.
       -y     Assume an answer of 'yes' to all questions;
```

Why Journaling

- Before journaling, after a system crash a utility would need to check the whole disk and look for possible data errors. Journaling keeps a log of the changes, making it easier to identify what areas are prone to have data failures. This make a disk check utility work a lot faster than with non-journaling filesystems

Advanced Tools

ext2 and ext3

```
E2IMAGE

NAME
       e2image - Save critical ext2/ext3 filesystem metadata to a file

SYNOPSIS
       e2image [ -rsI ] device image-file

DESCRIPTION
       The  e2image program will save critical ext2 or ext3 filesystem metadata located on
device to a file specified by image-file.  The image file may be examined  by dumpe2fs and
debugfs, by using the -i option to those programs.  This can assist an expert in recovering
```

```
catastrophically corrupted filesystems.  In the  future, e2fsck will be enhanced to be able to
use the image file to help recover a badly damaged filesystem.
```

```
DUMPE2FS

NAME
       dumpe2fs - dump ext2/ext3 filesystem information

SYNOPSIS
       dumpe2fs [ -bfhixV ] [ -ob superblock ] [ -oB blocksize ] device

DESCRIPTION
       dumpe2fs  prints the super block and blocks group information for the filesystem present
on device.
```

```
DEBUGFS

NAME
       debugfs - ext2/ext3 file system debugger

SYNOPSIS
       debugfs  [  -Vwci  ]  [  -b  blocksize  ] [ -s superblock ] [ -f cmd_file ] [ -R
       request ] [ -d data_source_device ] [ device ]

DESCRIPTION
       The debugfs program is an interactive file system debugger. It can  be  used  to examine
and change the state of an ext2 file system. device  is the special file corresponding to the
device containing the ext2 file system (e.g /dev/hdXX).
```

Reiserfs Filesystems
- reiserfstune - Displays and adjusts filesystem parameters
- debugreiserfs - Similar to dumpe2fs and debugfs

XFS Filesystems
- xfs_info - Filesystem information
- xfs_growfs - Expands an xfs filesystem
- xfs_admin - Changes the parameters of an xfs filesystem
- xfs_repair - Repairs filesystem when mount checks are not sufficient
- xfs_db - Examines or debugs an xfs filesystem

Objective 3: Control Filesystem Mounting and Unmounting

Managing The Filesystem Table
- fstab is checked at every boot and mounts any filesystem listed there
- Entries are consulted when users wish to mount removable media
- Consists of six fields

```
$ cat /etc/fstab
# /etc/fstab: static file system information.
#
#<device>     <mount point>   <type>      <options>                  <dump>  <pass>
proc           /proc           proc        defaults                     0       0
/dev/sda1      /               ext3        defaults,errors=remount-ro   0       1
/dev/sda6      /home           ext3        defaults                     0       2
/dev/sda5      none            swap        sw                           0       0
/dev/scd0      /media/cdrom0   udf,iso9660 user,noauto,exec              0       0
```

. Device - Device file or partition holding the filesystem

. Mount Point - Directory on which the filesystem will be mounted
. Type - Type of filesystem (ext2, ext3, iso9660, msdos, nfs, smbfs, vfat, etc...)
. Options - Comma separated options
. Dump - Standard Unix backup utility. It will consult fstab to find out how often a system needs to be dumped. Linux native filesystems usually have a 1 and others have a 0
. Pass - Checked by “fsck -A” at boot. May contain a 0, 1 or 2

Note: A remote NFS filesystem would be place on fstab with the following format:

```
host:/share  /mount_point nfs    <options>    <dump>  <pass>
```

Mounting Filesystems

- Mount points must exist before you can mount anything over it
- Directories intended as mount point usually don't contain files or other directories. If a directory intended for mount has files and directories, those items become obscured when a mount is in effect
- The mount command can sometime detect the filesystem type (ext2, vfat), however most of the time it requires it to be specified via the (-t) flag
- It's also possibe to remount a command using the 'remount' flag for mount. If any process has an open file to that filesystem the remount will fail
- There are a few options to display mounted filesystems (mount, /proc/mounts, /etc/mtab):

```
$ cat /etc/mtab
/dev/sda3 / ext3 rw 0 0
proc /proc proc rw 0 0
sysfs /sys sysfs rw 0 0
devpts /dev/pts devpts rw,gid=5,mode=620 0 0
/dev/sda1 /boot ext3 rw 0 0
tmpfs /dev/shm tmpfs rw 0 0
none /proc/sys/fs/binfmt_misc binfmt_misc rw 0 0
sunrpc /var/lib/nfs/rpc_pipefs rpc_pipefs rw 0 0
none /proc/fs/vmblock/mountPoint vmblock rw 0 0
```

```
$ cat /proc/mounts
rootfs / rootfs rw 0 0
/dev/root / ext3 rw,data=ordered 0 0
/dev /dev tmpfs rw 0 0
/proc /proc proc rw 0 0
/sys /sys sysfs rw 0 0
none /selinux selinuxfs rw 0 0
/proc/bus/usb /proc/bus/usb usbfs rw 0 0
devpts /dev/pts devpts rw 0 0
/dev/sda1 /boot ext3 rw,data=ordered 0 0
tmpfs /dev/shm tmpfs rw 0 0
none /proc/sys/fs/binfmt_misc binfmt_misc rw 0 0
sunrpc /var/lib/nfs/rpc_pipefs rpc_pipefs rw 0 0
none /proc/fs/vmblock/mountPoint vmblock rw 0 0
/etc/auto.misc /misc autofs rw,fd=6,pgrp=5424,timeout=300,minproto=5,maxproto=5,indirect 0 0
-hosts /net autofs rw,fd=12,pgrp=5424,timeout=300,minproto=5,maxproto=5,indirect 0 0
```

```
$ mount
/dev/sda3 on / type ext3 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/sda1 on /boot type ext3 (rw)
tmpfs on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
none on /proc/fs/vmblock/mountPoint type vmblock (rw)
```

```
MOUNT

NAME
       mount - mount a file system

SYNOPSIS
       mount [-lhV]
      mount -a [-fFnrsvw] [-t vfstype] [-O optlist]
       mount [-fnrsvw] [-o options [,...]] device | dir
       mount [-fnrsvw] [-t vfstype] [-o options] device dir

DESCRIPTION
       All  files  accessible in a Unix system are arranged in one big tree, the file
hierarchy, rooted at /.  These files can be spread out over several devices. The mount command
serves to attach the file system found on some device to the big file tree.

OPTIONS
       -V     Output version.
      -h     Print a help message.
      -v     Verbose mode.
      -a     Mount all filesystems (of the given types) mentioned in fstab.
       -l     Add the ext2, ext3 and XFS labels in the mount output.
       -r     Mount the file system read-only. A synonym is -o ro.
      -w     Mount the file system read/write. This is the default. A synonym is -o rw.
      -L label
              Mount the partition that has the specified label.
       -t vfstype
              The argument following the -t is used to indicate the file system type.  The file
system types which are currently supported include: adfs, affs,  autofs,  cifs, coda, coherent,
cramfs, debugfs, devpts, efs, ext, ext2, ext3, hfs, hfsplus, hpfs, iso9660, jfs, minix, msdos,
ncpfs, nfs, nfs4, ntfs, proc, qnx4, ramfs, reiserfs, romfs, smbfs, sysv, tmpfs, udf, ufs,
umsdos, usbfs,  vfat,  xenix,  xfs,  xiafs

       -o     Options are specified with a -o flag followed by a comma separated string of
options.
              async
                     All I/O to the file system should be done asynchronously.
              auto
                     Can be mounted with the -a option.
              defaults
                     Use default options: rw, suid, dev, exec, auto, nouser, and async.
              dev
                     Interpret character or block special devices on the file system.
              exec
                     Permit execution of binaries.
              group
                     Allow an ordinary (i.e., non-root) user to mount the file system if one of
his groups matches the group of the device.
              encryption
                     Specifies an encryption algorithm to use. Used in conjunction with the
loop option.
              _netdev
                     The  filesystem resides on a device that requires network access (used to
prevent the system from attempting to mount these filesystems until the network has been
enabled on the system).
              noatime
                     Do not update inode access times on this file system (e.g, for faster
access on the news spool to speed up news servers).
              relatime
                     Update inode access times relative to modify or change time.
              noauto
                     Can only be mounted explicitly (i.e., the -a option will not cause the
file system to be mounted).
              nodev
                     Do not interpret character or block special devices on the file system.
              noexec
                     Do not allow direct execution of any binaries on the mounted file system.
```

```
(Until recently it was possible to  run  binaries  anyway using a command like /lib/ld*.so
/mnt/binary. This trick fails since Linux 2.4.25 / 2.6.0.)
              owner
                     Allow an ordinary (i.e., non-root) user to mount the file system if he is
the owner of the device.  This option implies the  options nosuid and nodev (unless overridden
by subsequent options, as in the option line owner,dev,suid).
              remount
                     Attempt to remount an already-mounted file system.  This is commonly used
to change the mount flags for a file system, especially to make a readonly file system
writeable. eg: rw,errors=remount-ro
               ro
                     Mount the file system read-only.
              rw
                     Mount the file system read-write.
              suid
                     Allow set-user-identifier or set-group-identifier bits to take effect.
              sync
                     All I/O to the file system should be done synchronously. In case of media
with limited number  of  write  cycles  (e.g.  some  flash drives) "sync" may cause life-cycle
shortening.
              dirsync
                     All  directory  updates  within the file system should be done
synchronously.  This affects the following system calls: creat, link, unlink, symlink, mkdir,
rmdir, mknod and rename.
              user
                     Allow an ordinary user to mount the file system.  The name of the mounting
user is written to mtab so that he can unmount  the  file system  again.  This option implies
the options noexec, nosuid, and nodev (unless overridden by subsequent options, as in the
option line user,exec,dev,suid).
               users
                     Allow every user to mount and unmount the file system.  This option
implies the options noexec, nosuid, and nodev (unless overridden by subsequent options, as in
the option line users,exec,dev,suid).
```

Unmount Filesystems

- Umount synchronizes the buffer of the filesystem and makes it unavailable
- Errors can be generated if filesystem contains a file open or has a working directory within the system
- A mount point to device can be given as an argument

```
UMOUNT

NAME
       umount - unmount file systems

USAGE
       -V     Print version and exit.
      -h     Print help message and exit.
      -v     Verbose mode.
      -n     Unmount without writing in /etc/mtab.
       -a     All of the file systems described in /etc/mtab are unmounted. (With umount
version 2.7 and later: the proc filesystem is not unmounted.)
       -t vfstype
              Indicate that the actions should only be taken on file systems of the specified
type.  More than one type may be specified in a comma separated list.  The list of file system
types can be prefixed with no to specify the file system types on which no action should be
taken.
      -O options
              Indicate that the actions should only be taken on file systems with the specified
options in /etc/fstab.  More than one option type may  be specified in a comma separated list.
Each option can be prefixed with no to specify options for which no action should be taken.
      -f     Force unmount (in case of an unreachable NFS system).  (Requires kernel 2.1.116 or
later.)
```

Swap Space

- Swap spaces cannot be mounted. Instead they are enabled or disabled with 'swapon' or 'swapoff'

```
$ swapon -s
Filename                                Type            Size    Used    Priority
/dev/sda2                               partition       530136  72      -1

$ cat /proc/swaps
Filename                                Type            Size    Used    Priority
/dev/sda2                               partition       530136  72      -1
```

Objective 4: Set and View Disk Quotas

Quotas

- Managing disk space can be a problem on a multi user system. Disk quotas allows an administrator to add a limitation on file storage.
- Disk quota can be assigned based on:
  . filesystems listed on /etc/fstab;
  . users listed on /etc/passwd;
  . groups listed on /etc/group
- Kernel older than 2.4 or 2.6 may not have full quota support requiring a new kernel build. Newer kernels support it via modules
- Disk quota is usually assigned to filesystems where user can store data (eg: /home/user/)
- Most filesystems store quota information on a file (aquota.user and aquota.group), while xfs it's part of the filesystem metada

Quota Limits

- Each filesystem as up to five quota limits, which are specified in disk blocks (usually 1,024 bytes each)
- Limits can be set using *'edquota'*
- Limits:
  . Per-user hard limit - Once limit is reached user cannot write data to the disk
  . Per-user soft limit - One limit is reached (but not the hard limit) user receives a warning message to clean up files on their system.
  . Per-group hard limit - Once reached, none of the group users will be able to write data (even if their personal hard limit has not been reached)
  . Per-group soft limit - Same as per-user soft limit, however based on group
  . Grace period - Once a soft limit has been reached, a user/group enters the grace period. After the grace period expires it becomes a hard limit. Grace period is defined in time (month, week, days, minutes, seconds) and is commonly set to 7 days.

Note: When user/group reaches a hard limit the program will fail and data may be lost. If shell is hidden user may not get the warning, resulting in a program error message that the disk is full or write protected.

Quota Commands

quota

- Displays quota for the invoking user and any specified filesystem
- The (-v) option displays quota on all filesystems
- Example:

```
# quota -uv test
Disk quotas for user test (uid 514):
Filesystem blocks quota limit grace files quota limit grace
/dev/hda3 100 5120 5120 10 1000 1000

# quota -gv test
Disk quotas for group test (gid 514):
Filesystem blocks quota limit grace files quota limit grace
```

```
/dev/hda3 100 0 0 10 0 0
```

```
QUOTA

NAME
       quota - display disk usage and limits

SYNOPSIS
       quota [ -F format-name ] [ -guvsil | q ]
       quota [ -F format-name ] [ -uvsil | q ] user...
       quota [ -F format-name ] [ -gvsil | q ] group...

DESCRIPTION
       quota  displays  users'  disk usage and limits.  By default only the user quotas
       are printed.

       quota reports the quotas of  all  the  filesystems  listed  in  /etc/mtab.   For
       filesystems that are NFS-mounted a call to the rpc.rquotad on the server machine
       is performed to get the information.


OPTIONS
       -g     Print group quotas for the group of which the  user  is  a  member.   The
              optional  group  argument(s)  restricts  the  display  to  the  specified
              group(s).

       -u     flag is equivalent to the default.

       -v     will display quotas on filesystems where no storage is allocated.

       -s     option will make quota(1) try to choose units for  showing  limits,  used
              space and used inodes.

       -i     ignore mountpoints mounted by automounter

       -l     report  quotas only on local filesystems (ie. ignore NFS mounted filesys-
              tems).
       -q     Print a more terse message, containing only  information  on  filesystems
              where usage is over quota.
```

quotaon - quotaoff

- Enables to disables quota checking
- This command (quotaon) is usually added to a runlevel script
- Example:

```
# quotaon -av
/dev/sda9: group quotas turned on
/dev/sda9: user quotas turned on
/dev/hda1: group quotas turned on
/dev/hda1: user quotas turned on
```

```
QUOTAON - QUOTAOFF

NAME
       quotaon, quotaoff - turn filesystem quotas on and off

SYNOPSIS
       /sbin/quotaon [ -vugfp ] [ -F format-name ] filesystem...
       /sbin/quotaon [ -avugfp ] [ -F format-name ]

       /sbin/quotaoff [ -vugp ] [ -x state ] filesystem...
       /sbin/quotaoff [ -avugp ]

DESCRIPTION
   quotaon
       quotaon  announces  to  the  system that disk quotas should be enabled on one or
       more filesystems. The filesystem quota files must be present in the root  direc-
```

```
       tory  of the specified filesystem and be named either aquota.user (for version 2
       user quota), quota.user (for version 1 user quota), aquota.group (for version  2
       group quota), or quota.group (for version 1 group quota).

OPTIONS
   quotaon
       -a     All  automatically  mounted  (no  noauto  option)  non-NFS filesystems in
              /etc/fstab with quotas will have their quotas turned on.   This  is  nor-
              mally used at boot time to enable quotas.

       -v     Display a message for each filesystem where quotas are turned on.

       -u     Manipulate user quotas. This is the default.

       -g     Manipulate group quotas.

       -p     Instead  of  turning  quotas  on just print state of quotas (ie. whether.
              quota is on or off)

   quotaoff
       -F format-name
              Report quota for specified format (ie. don't  perform  format  autodetec-
              tion).   Possible format names are: vfsold (version 1 quota), vfsv0 (ver-
              sion 2 quota), xfs (quota on XFS filesystem)

       -a     Force all filesystems in /etc/fstab to have their quotas disabled.

       -v     Display a message for each filesystem affected.

       -u     Manipulate user quotas. This is the default.

       -g     Manipulate group quotas.

       -p     Instead of turning quotas off just print state of  quotas  (ie.  whether.
              quota is on or off)
```

quotacheck

- Checks quotas on all filesystems and either creates quota files if they not exist or repairs them
- This command is usually added to a runlevel script or cron
- Example

```
# quotacheck -aguv
Scanning /dev/sda9 [/home] done
Checked 237 directories and 714 files
Using quotafile /home/quota.user
Using quotafile /home/quota.group
Scanning /dev/hda1 [/mnt/hd] done
Checked 3534 directories and 72673 files
Using quotafile /mnt/hd/quota.user
Using quotafile /mnt/hd/quota.group
```

```
QUOTACHECK  - This command is not specifically called out in the LPI Objectives for Exam 101

NAME
       quotacheck  -  scan  a filesystem for disk usage, create, check and repair quota
       files

SYNOPSIS
       quotacheck [ -gubcfinvdMmR ] [ -F quota-format ] -a | filesystem

DESCRIPTION
       quotacheck examines each filesystem, builds a table of current disk  usage,  and
       compares  this  table  against  that  recorded  in  the  disk quota file for the
       filesystem (this step is ommitted if option -c is specified). If  any  inconsis-
       tencies  are  detected,  both  the quota file and the current system copy of the
       incorrect quotas are updated (the latter only occurs if an active filesystem  is
       checked  which is not advised).  By default, only user quotas are checked.  quo-
       tacheck expects each  filesystem  to  be  checked  to  have  quota  files  named
```

```
       [a]quota.user  and [a]quota.group located at the root of the associated filesys-
       tem.  If a file is not present, quotacheck will create it.

OPTIONS
       -b     Forces  quotacheck  to  make backups of the quota file before writing the
              new data.

       -v     quotacheck reports its operation as it progresses.  Normally it  operates
              silently.   If  the option is specified twice, also the current directory
              is printed (note that printing can slow down the scan measurably).

       -d     Enable debugging mode.  It will result in a lot of information which  can
              be used in debugging the program. The output is very verbose and the scan
              will be slow.

       -u     Only user quotas listed in /etc/mtab or on the filesystems specified  are
              to be checked.  This is the default action.

       -g     Only group quotas listed in /etc/mtab or on the filesystems specified are
              to be checked.

       -c     Don't read existing quota files. Just perform a new scan and save  it  to
              disk.   quotacheck  also  skips scanning of old quota files when they are
              not found.

       -f     Forces checking and writing of new quota files on filesystems with quotas
              enabled. This is not recommended as the created quota files may be out of sync

       -a     Check all mounted non-NFS filesystems in /etc/mtab
```

edquota

- Extracts quota information from the user quota file, creates a temporary file and opens an editor so you can edit the quota.
Example (with vi):

```
# edquota -u jdoe
Quotas for user jdoe:
/dev/sda9: blocks in use: 87, limits (soft = 99900, hard = 100000)
        inodes in use: 84, limits (soft = 0, hard = 0)
/dev/hda1: blocks in use: 0, limits (soft = 0, hard = 0)
        inodes in use: 0, limits (soft = 0, hard = 0)
~
~
"/tmp/EdP.auHTZJ0" 5 lines, 241 characters
```

- Quotas can also be copied between users with the (-p) flag. The example below copies quota options from user1 to all other users

```
$ edquota -p user1 user2 user3 user4
```

```
EDQUOTA

NAME
       edquota - edit user quotas

SYNOPSIS
       edquota [ -p protoname ] [ -u | -g ] [ -r ] [ -F format-name ] [ -f filesystem ]
       username...

       edquota [ -u | -g ] [ -F format-name ] [ -f filesystem ] -t

       edquota [ -u | -g ] [ -F format-name ] [ -f filesystem ]  -T  username |  group-
       name...

DESCRIPTION
       edquota  is a quota editor.  One or more users or groups may be specified on the
       command line. If a number is given  in  the  place  of  user/group  name  it  is
       threated  as an UID/GID. For each user or group a temporary file is created with
```

```
       an ASCII representation of the current disk quotas for that user or group and an
       editor is then invoked on the file.  The quotas may then be modified, new quotas
       added, etc.  Setting a quota to zero indicates that no quota should be  imposed.

       The  editor  invoked is vi(1) unless either the EDITOR or the VISUAL environment
       variable specifies otherwise.

       Only the super-user may edit quotas.

OPTIONS
       -r     Edit also non-local quota use rpc.rquotad on remote server to set  quota.
              This  option  is available only if quota tools were compiled with enabled
              support for setting quotas over RPC.  The -n option is equivalent, and is
              maintained for backward compatibility.

       -u     Edit the user quota. This is the default.

       -g     Edit the group quota.

       -p protoname
              Duplicate  the  quotas  of  the prototypical user specified for each user
              specified.  This is the normal mechanism used to  initialize  quotas  for
              groups of users.

FILES
       aquota.user or aquota.group
                           quota  file at the filesystem root (version 2 quota, non-XFS
                           filesystems)
       quota.user or quota.group
                           quota file at the filesystem root (version 1 quota,  non-XFS
                           filesystems)
       /etc/mtab           mounted filesystems table
```

repquota

- Generates quota reports
- A (+) sign indicates a user that is over his quota limit
- Example:

```
# repquota -v /home
*** Report for user quotas on /dev/sda9 (/home)
                        Block limits     File limits
User         used soft    hard grace used soft hard grace
root    -- 418941      0       0        269    0    0
328     --    1411     0       0         20    0    0
jdean   --     9818 99900 100000        334    0    0
u1      --       44     0       0         43    0    0
u2      --       44     0       0         43    0    0
u3      --      127   155     300        124    0    0
jdoe    --       87 99900 100000         84    0    0
bsmith  --       42 1990    2000         41    0    0
```

```
REPQUOTA

NAME
       repquota - summarize quotas for a filesystem

SYNOPSIS
       /usr/sbin/repquota  [ -vsiug ] [ -c | -C ] [ -t | -n ] [ -F format-name
       ] filesystem...

       /usr/sbin/repquota [ -avtsiug ] [ -c | -C ] [ -t | -n ]  [  -F  format-
       name ]

DESCRIPTION
       repquota  prints  a summary of the disc usage and quotas for the speci-
       fied file systems.  For each user  the  current  number  of  files  and
       amount  of  space (in kilobytes) is printed, along with any quotas cre-
```

```
       ated with  edquota(8).   As  repquota  has  to  translate  ids  of  all
       users/groups  to  names  (unless option -n was specified) it may take a
       while to print all the information. To make translating as fast as pos-
       sible  repquota tries to detect (by reading /etc/nsswitch.conf) whether
       entries are stored in standard plain  text  file  or  in  database  and
       either  translates  chunks of 1024 names or each name individually. You
       can override this autodetection by -c or -C options.

OPTIONS
       -a     Report on all filesystems indicated in  /etc/mtab  to  be  read-
              write with quotas.

       -v     Report  all quotas, even if there is no usage. Be also more ver-
              bose about quotafile information.
```

warnquota
- Used to email warnings to users that are over quota
- Usually run under cron

Enabling Quotas
- Quota support must be enabled on kernel (there's a remote chance that it would not)
- Quota must be enabled per file system and can be done per user, per group or both

Step 1
- Add the option 'usrquota' and/or 'grpquota' to the filesystem in /etc/fstab
```
/dev/sda3 /home ext2 defaults,usrquota,grpquota 1 2
```

Step 2
- Create the proper configuration files (quota.user and/or quota.group) in the root of the file system.
```
# touch /home/quota.user /home/quota.group
```

- Add the proper permissions (set quota.group 644 if you want users to be able to examine quotas on groups they belong to)
```
# chmod 600 /home/quota.user /home/quota.group
```

Step 3
- Initialize the database with 'quotacheck'
```
# quotacheck -avug
Scanning /dev/sda9 [/home] done
Checked 236 directories and 695 files
Using quotafile /home/quota.user
Using quotafile /home/quota.group
```

Step 4
- Verify that database was created
```
# ls -l /home/quota.*
-rw------- 1 root root 16192 Dec 27 19:53 /home/quota.group
-rw------- 1 root root 16192 Dec 27 19:53 /home/quota.user
```

Step 5
- Enable quota
```
# quotaon -a
```

Step 6
- Add quota to the desired runlevel
```
if [ -x /sbin/quotacheck ]; then
    echo "Checking quotas."
    /sbin/quotacheck -avug
    echo " Done."
fi
if [ -x /sbin/quotaon ]; then
```

```
    echo "Turning on quotas."
   /sbin/quotaon -avug
fi
```

Step 7

- Add quotacheck to a script on /etc/crontab.*/ (preferably weekly) or root's crontab

```
#!/bin/bash
exec /sbin/quotacheck -avug
```

or

```
# run quotacheck weekly
0 3 * * 0   /sbin/quotacheck -avug
```

Objective 5: Use File Permissions to Control Access to Files

Linux Access Control

- A set or properties maintained separately for each file (also known as access mode)
- Access mode is part of file's
- Controls access via:
  . User
  . Group
  . Other
- New files inherit user and default group from creator

Permission Table

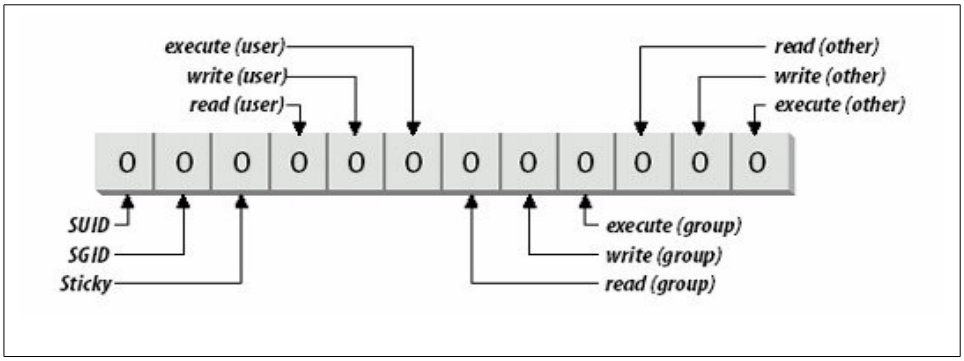
Permission	Mnemonic	File Permission	Directory Permission
Read	r	Read	List contents
Write	w	Write or change	Create and remove files within
Execute	x	Run as a program	cd into

Note: Without read permission to a directory it's contents cannot be listed, however a file can still be accessed if you know it's path

```
$ ll dir1
ls: dir1: Permission denied

$ ll dir1/file1
-rw-r--r-- 1 root root 3 Mar  7 17:35 dir1/file1
```

Access Mode Bits



Special Permissions
- SUID - Enables a user to run a program with another user owning the result (usually the user who runs the program owns the result). This allows the use of root only program by normal users
- SGID - Same as SUID however for folder settings. Another useful property is that files created within a folder that has SGID enabled will belong to the group owner of that folder allowing a group shared folder.
- Sticky - When set on a directory, only file owner, directory owner and root can delete the files. Older Unix system used sticky bit on files to keep loaded executable in memory. Newer Linux kernels ignore it if set on files

The mode String
- A file with special permission set will add a capital “S” or “T” representing the permission. If executable permission is also set for that bit the indicating letter will change to lower case.

```
# ls -l saran
-rw-r----- 1 root root 0 Feb  2 16:16 saran

# chmod 4640 saran  #rw-
# ls -l saran
-rwSr----- 1 root root 0 Feb  2 16:16 saran

# chmod 4740 saran  #rwx
# ls -l saran
-rwsr----- 1 root root 0 Feb  2 16:16 saran
```

Identifying User And Groups

Effective Id Vs Current Id
- Two commands can be used to identify who you are logged in right now and who you initially logger in as

```
[user1@centos ~]$ whoami  # effective user
user1

[user1@centos ~]$ who am i  # original user
root     pts/1        2009-03-08 17:26 (localhost:10.0)
```

- Another command that can be used with either user id or name is 'id'

```
$ id
uid=1000(victor) gid=1000(victor)
groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse)
,109(lpadmin),115(admin),125(vboxusers),126(sambashare),1000(victor),1001(usbusers)
```

Displaying Groups
- Two commands can help with this, 'id' and 'groups'

```
$ groups
victor adm dialout cdrom floppy audio dip video plugdev fuse lpadmin admin vboxusers sambashare
usbusers
```

File Attributes
- There are additional attributes available on filesystems (including 'immutable', which avoids file deletion)
“The letters ‘acdijsuADST’ select the new attributes for the files: append only (a), compressed (c), no dump (d), immutable (i), data journalling (j), secure deletion (s), no tail-merging (t), undeletable (u), no atime updates (A), synchronous directory updates (D), synchronous updates (S), and top of directory hierarchy (T). ”

- Two commands can be used to modify and list these special attributes, 'chattr' and 'lsattr'.

```
$ ll file1
-rw-r--r-- 1 root root 16 Feb 23 16:04 file1
```

```
root@~ # chattr +i file1

root@~ # rm file1
rm: remove write-protected regular file `file1'? y
rm: cannot remove `file1': Operation not permitted

root@~ # lsattr file1
----i--------- file1
```

Setting Access Mode

New Files (umask)
- New files are create according to user's default settings (established by 'umask')
- Debian
```
$ umask
0022
```

- AIX
```
$ umask
077
```

- umask takes either three digit octal strings or integer

```
# umask
0022
# umask 2
# umask
0002
# umask 0
# umask
0000
# umask 22
# umask
0022
```

- The umask value is handed different by files and folders.
  . File - (umask -6) = permission ; 022 = 644
  . Folder - (umask -7) = permission ; 022 = 755
- For more security a umask of 077 can be used
- Symbolic mode (u, g, o) can also be used on umask (077 is the same as u=rwx,g=,o=)
- To change your umask add the umask command with a value in a startup script (like .bash_profile)
- Programs started by user will inherit his/her permission. This can cause the program not be able to access filesystem object that the user does not have access (see SUID)

Changing Access Mode
- Can be done via symbolic or octal mode
- Special permissions
  . X - Adds execute permission to folders or files with special execute permission
  . s - SUID or SGID
  . t - Sticky bit
- Examples
```
$ chmod -v u=rw,go=r afile
mode of afile retained as 0644 (rw-r--r--)

$ chmod -R -v o-rwx adir
mode of adir retained as 0770 (rwxrwx---)
mode of adir/file1 changed to 0660 (rw-rw----)
mode of adir/file2 changed to 0660 (rw-rw----)
mode of adir/file3 changed to 0660 (rw-rw----)
mode of adir/file4 changed to 0660 (rw-rw----)
```

```
mode of adir/dir1 changed to 0770 (rwxrwx---)
mode of adir/dir1/file6 changed to 0660 (rw-rw----)
mode of adir/dir1/file5 changed to 0660 (rw-rw----)
mode of adir/dir2 changed to 0770 (rwxrwx---)

$ ll file4
-rw-rw-r-- 1 user1 user1 0 Mar  7 18:35 file4
$ chmod -v o= file4
mode of `file4' changed to 0660 (rw-rw----)
```

```
CHMOD

NAME
       chmod - change file access permissions

SYNOPSIS
       chmod [OPTION]... MODE[,MODE]... FILE...
       chmod [OPTION]... OCTAL-MODE FILE...
       chmod [OPTION]... --reference=RFILE FILE...

DESCRIPTION
       This  manual  page  documents  the GNU version of chmod. chmod changes the permissions
of each given file according to mode, which can be either a symbolic representation of changes
to make, or an octal number representing the bit pattern for the new permissions.

OPTIONS
       Change the mode of each FILE to MODE.

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

       --no-preserve-root
              do not treat '/' specially (the default)

       --preserve-root
              fail to operate recursively on '/'

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

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

       --reference=RFILE
              use RFILE's mode instead of MODE values

       -R, --recursive
              change files and directories recursively
```

Objective 6: Manage File Ownership

- Common commands used to change file ownership are 'chown' and 'chgroup'

chown

- A group number or id can be given as parameter

```
CHOWN

NAME
       chown - change file owner and group

SYNOPSIS
       chown [OPTION]... [OWNER][:[GROUP]] FILE...
       chown [OPTION]... --reference=RFILE FILE...
```

```
DESCRIPTION
       This  manual page documents the GNU version of chown.  chown changes the user and/or
group ownership of each given file.  If only an owner (a user name or numeric user ID)  is
given,  that user  is  made the owner of each given file, and the files' group is not changed.
If the owner is followed by a colon and a group name (or numeric group ID), with no spaces
between them, the group ownership of the files is changed as well.  If a colon but no group
name follows the user name, that user is made the owner of the files and the group of the files
is  changed  to  that user's login group.  If the colon and group are given, but the owner is
omitted, only the group of the files is changed; in this case, chown performs the same function
as chgrp.   If  only  a colon  is given, or if the entire operand is empty, neither the owner
nor the group is changed.

OPTIONS
       Change the owner and/or group of each FILE to OWNER and/or GROUP.  With --reference,
change the
       owner and group of each FILE to those of RFILE.

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

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

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

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

Note: BSD style uses a "." instead of ":", which comes from System V

chgrp

- A group number or id can be given as parameter

```
CHGRP

NAME
       chgrp - change group ownership

SYNOPSIS
       chgrp [OPTION]... GROUP FILE...
       chgrp [OPTION]... --reference=RFILE FILE...

DESCRIPTION
       Change  the  group  of  each FILE to GROUP.  With --reference, change the group of each
FILE to
       that of RFILE.

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

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

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

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

newgrp

- Can be used to open a new shell using another of the user's group as the default. Once the shell is closed user's group priorities groes back to normal

Objective 7: Create and Change Hard and Symbolic Links

Different Types Of Links

Symbolic Links
- Points to another file
- Filesystem independent
- Has it's own inode
- Can be broken (or stale)
- Paths needed to be watched for. A symlink created using relative path is not the same as one created with absolute path

Hard Links
- Not a link but a reference to an existing inode
- All permissions and data are save on the inode, thus changes that are kept within the inode made to one file will affect the other (not rename and delete)
- Data will only be deleted one all links have been deleted
- Directories cannot have a hard link
- The command 'ls -l' will also provide a total number of pointers to that file (or inode)

```
$ find . -inum 421422 -exec ls -li {} \;
421422 -rwxr-xr-x 3 root root 47288 May 24  2008 ./mkfs.ext3
421422 -rwxr-xr-x 3 root root 47288 May 24  2008 ./mkfs.ext2
421422 -rwxr-xr-x 3 root root 47288 May 24  2008 ./mke2fs
```

Why Use Links
- Make changes to original files (instead of multiple) and all links will have the same (eg: /etc/init.d/)
- Avoid wasting disk space having multiple copies of the same file
- File name reference (eg: when upgrading to a newer kernel)

Notes:
- Creating a symbolic link of another symbolic link will create a third symbolic link - link-s2 ---> link-s1 ---> file
- Creating a hard link of a symbolic link will create a symbolic link of the previous symbolic link or file - link-h1 ---> file

```
LN

NAME
       ln - make links between files

SYNOPSIS
       ln [OPTION]... [-T] TARGET LINK_NAME   (1st form)
       ln [OPTION]... TARGET                  (2nd form)
       ln [OPTION]... TARGET... DIRECTORY     (3rd form)
       ln [OPTION]... -t DIRECTORY TARGET...  (4th form)

DESCRIPTION
       In  the  1st  form, create a link to TARGET with the name LINK_NAME.  In the 2nd form,
create a link to TARGET in the current directory.  In the 3rd and 4th forms, create links to
each TARGET in  DIRECTORY.   Create  hard  links by default, symbolic links with --symbolic.
When creating hard links, each TARGET must exist.

       -b     like --backup but does not accept an argument

       -d, -F, --directory
              allow the superuser to attempt to hard link directories (note: will probably fail
due to
              system restrictions, even for the superuser)

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

```
      -n, --no-dereference
             treat destination that is a symlink to a directory as if it were a normal file

      -i, --interactive
             prompt whether to remove destinations

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

      -S, --suffix=SUFFIX
             override the usual backup suffix

      -t, --target-directory=DIRECTORY
             specify the DIRECTORY in which to create the links

      -T, --no-target-directory
             treat LINK_NAME as a normal file

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

Preserving Links

- Commands like cp and tar have options to copy the links instead of files to preserve disk space. By default they are not enabled, which will result in the file itself being copied.
- Example

```
# ls -l dir1
total 13
lrwxrwxrwx 1 root root      19 Jan 4 02:43 file1 -> /file1
-rw-r--r-- 1 root root   10240 Dec 12 17:12 file2

# cp -r dir1 dir2
# ls -l dir2
total 3117
-rw-r--r-- 1 root root 3164160 Jan 4 02:43 file1
-rw-r--r-- 1 root root   10240 Jan 4 02:43 file2

# cp -rd dir1 dir3
# ls -l dir3
total 13
lrwxrwxrwx 1 root root      19 Jan 4 02:43 file1 -> /file1
-rw-r--r-- 1 root root   10240 Jan 4 02:43 file2
```

Finding Links

=> Finding symbolic links

1- Finding file pointed to a symbolic link

```
$ ls -l
lrwxrwxrwx 1 root root    5 Feb  7 15:36 saran-l -> saran
```

2- Finding links to a file

```
$ find . -lname saran
./saran-l
```

=> Finding hard links

- This is a bit easier as hard links are bound to the same filesystem

```
$ ls -li mkfs.ext* mke2fs
421422 -rwxr-xr-x 3 root root 47288 May 24  2008 mke2fs
421422 -rwxr-xr-x 3 root root 47288 May 24  2008 mkfs.ext2
421422 -rwxr-xr-x 3 root root 47288 May 24  2008 mkfs.ext3
```

. Method 1

```
$ ls -li
2269229 -rw-r--r-- 2 root root    0 Feb  7 15:13 saran
2269229 -rw-r--r-- 2 root root    0 Feb  7 15:13 saran-h
$ find . -inum 421422
./mkfs.ext3
./mkfs.ext2
./mke2fs
```

. Method 2

```
$ find . -samefile mke2fs
./mkfs.ext3
./mkfs.ext2
./mke2fs
```

Objective 8: Find Systems Files and Place Files in the Correct Location

- In 1993 a Linux community formed a project to provide a standardized filesystem layout for all standardized Linux distributions. The project was called FSSTAND and was released in 1994.
- In the following year the group started to include Unix and Unix-like operating systems. This had a great feedback from many people and was renamed to FHS - Filesystem Hierarchy Standard.
- FHS is not a requirement for Linux developers, however most of the big distros abide to it as they have an understanding of the importance to standards.
http://www.pathname.com/fhs/

Data Types

- Divided in two categories

=> Data Sharing
- Sharable - Data can be shared over a network and used by other users and operating system. This includes data files, executables, etc...
- Non-sharable data - Not shared. Eg: configuration files

=> Data Modification
- Variable - Data changes by natural frequent processes. Eg: /var/log/messages
- Static - Data rarely changes (days, years). Eg: /bin/bash, /bin/ls

	Sharable	Non-sharable
Static	/usr/ /usr/local/	/etc/ /boot/
Variable	/var/mail/ /home/	/var/log/ /proc/

The Root Filesystem

- It should contain utilities and files sufficient to boot the operating system and mount file systems
- Contain utilities needed by the system admin to repair or restore damaged systems
- Should be relatively small as smaller systems are less pron o be corrupted due to a system crash
- Software should create files or directories in the root directory

Essential Directories
- /bin/ - Contains essentials executable commands that may be needed in case of a system problem
- /dev/ - Device files
- /etc/ - Configuration files unique to the system needed at boot
- /lib/ - Shared libraries

- /mnt/ - Provides a centralized mount point to system administrators
- /root/ - Recommended root's home
- /sbin/ - Essential system administration utilities (fsck, fdisk, mkfs)

Non-essential Directories
- /boot/ - Contain files for the boot loader. Most of the times also contains the kernel.
- /home/ - User personal directory
- /opt/ - Software that is not packaged with system (3rd party software)
- /tmp/ - Temporary files (recommended to be deleted after system reboot)
- /usr/ - Executable programs that are not needed on boot (or not needed to recover systems). Usually mounted as read only via NFS from a network location
- /var/ - Data that varies over time (web sites, ftp, logs, mail, spool)

The /usr/ Filesystem
- Contains user and system administration commands and daemons that are only used under normal operation of the OS
- No host-specific or variable data are stores in /usr/
- Large software packages cannot place a directory directly under /usr/ (except X11)

Directories
- /usr/X11R6/ - Contains directories for XFree86
- /usr/bin/ - User commands that are not necessary for emergency system maintenance
- /usr/include/ - Standard location for header files used for C and C++
- /usr/lib/ - Shared libraries for various programs. Creation of sub-folders is also allowed (eg: /usr/lib/vlc/)
- /usr/local/ - A top level of a second hierarchy. Contains subdirectories with same name as found in /usr/

```
$ ls -l /usr/local/
total 32
drwxr-xr-x  2 root root 4096 2008-10-18 19:55 bin
drwxr-xr-x  2 root root 4096 2008-07-02 06:16 etc
drwxr-xr-x  2 root root 4096 2008-07-02 06:16 games
drwxr-xr-x  2 root root 4096 2008-07-02 06:16 include
drwxr-xr-x  6 root root 4096 2009-01-15 23:48 lib
lrwxrwxrwx  1 root root    9 2008-07-13 02:39 man -> share/man
drwxr-xr-x  2 root root 4096 2008-07-02 06:16 sbin
drwxr-xr-x 14 root root 4096 2009-01-16 00:01 share
drwxr-xr-x  2 root root 4096 2008-07-02 06:16 src
```

- /usr/sbin/ - System administration commands that are not essential for filesystem emergency
- /usr/share/ - Datafiles that are independent of hardware architecture and operating system versions
- /usr/src/ - Optional directory on newer gibc-based systems. Older libc4 and libc5 systems required it to have a copy of the kernel source or include/asm and include/linux for kernel header files

The /var/ Filesystem
- Contains data that varies over time
- Because data keeps growing and changing, this directory is usually not included with / to prevent the / partition from filling

Directories
- /var/account/ - Process accounting data
- /var/cache/ - Used by programs to store intermediate date. Programs should be able to regenerate the date so the system admin can delete files as needed. This folder does not need to be backed up
- /var/crash/ - Crash dumps
- /var/games/ - Saves game data like state information, scores, etc..
- /var/lock/ - Used by applications to signal their existence to other processes. File are usually empty

```
$ file /var/lock/subsys/vmware
/var/lock/subsys/vmware: empty
```

- /var/log/ - System log files
- /var/mail/ - System mailbox. Replacement for /var/spool/mail/

```
$ ls -l /var/spool/mail
lrwxrwxrwx 1 root root 7 2008-07-13 02:46 /var/spool/mail -> ../mail
```

- /var/opt/ - Location for temporary files created by programs in /opt/
- /var /run/ - Contains various files describing present state of the system (like pid files)

```
$ cat /var/run/sshd.pid
5060
$ ps aux | grep sshd | grep -v grep
root      5060  0.0  0.0   5316   992 ?        Ss   12:33   0:00 /usr/sbin/sshd
```

- /var/spool/ - Queued information (like cron, printer, etc...)
- /var/state/ - Information that helps applications maintain state across multiple instances
- /var/tmp/ - Same as /tmp/, owever data here is more persistent and may not be deleted after system boot
- /var/yp/ - Database files for NIS - Network Information Service

Where's That Binary

- Binary files locations for /bin/ and /sbin/

Binary file locations

Type of file	User Commands	System Admin Commands
Vendor-supplied, essential (/)	/bin/	/sbin/
Vendor-supplied, non-essential (/usr/)	/usr/bin/	/usr/sbin/
Locally-supplied, non-essential (/usr/)	/usr/local/bin/	/usr/local/sbin/

Locating Files

- There are a set of tools that can be used for help in finding file sin the system:
  . find
  . which
  . type
  . whereis
  . locate
  . whatis
  . apropos

Utility	Usage	Database
find	Searches specified areas in the filesystem	no
which	Uses PATH to locate executables	no
type	Tells how a command will be evaluated	no
whereis	Finds location, source and man page for a command	
locate (or slocate)	Locates files that might be related to name given	yes - updatedb
whatis	Displays short description of system commands	yes
apropos	Searches word on manual pages name and description	yes

find

```
FIND

NAME
       find - search for files in a directory hierarchy

SYNOPSIS
       find [-H] [-L] [-P] [path...] [expression]

DESCRIPTION

       GNU find searches the directory tree rooted at each given file name by evaluating the
```

```
given expression from left to right, according to  the  rules of precedence (see section
OPERATORS), until the outcome is known (the left hand side is false for and operations, true
for or), at which point find moves on to the  next  file name.
```

which

- Searches user path and displays the first occurrence of a command
- The option (-a) can be used to display multiple occurrences

```
$ which -a awk
/bin/awk
/usr/bin/awk
```

```
WHICH

NAME
       which - shows the full path of (shell) commands.

SYNOPSIS
       which [options] [--] programname [...]

DESCRIPTION
       Which takes one or more arguments. For each of its arguments it prints to stdout the
full path of the executables that would have been executed when this argument had been entered
at the shell prompt. It does this by searching for an executable or script in the directories
listed in the environment variable PATH using the same algorithm as bash(1).

OPTIONS
       --all, -a
           Print all matching executables in PATH, not just the first.
```

type

- Tells how a command will be evaluated
- Displays shell built-ins as well

```
$ which ulimit
/usr/bin/which: no ulimit in
(/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/b
in)
$ type ulimit
ulimit is a shell builtin
```

whereis

- Finds location, source and man page for a command

```
$ whereis fdisk
fdisk: /sbin/fdisk /usr/share/man/man8/fdisk.8.gz
```

```
WHEREIS(1)                                                              WHEREIS(1)

NAME
       whereis - locate the binary, source, and manual page files for a command

SYNOPSIS
       whereis [ -bmsu ] [ -BMS directory...  -f ] filename ...

DESCRIPTION
       whereis  locates  source/binary  and  manuals sections for specified files.  The
supplied names are first
       stripped of leading pathname components and any (single) trailing extension of the
form .ext,  for  exam-
       ple,  .c.   Prefixes  of s.  resulting from use of source code control are also dealt
with.  whereis then
       attempts to locate the desired program in a list of standard Linux places.
```

locate

- Uses a database of stored path
- Searches for file paths on a database that is usually update by cron
- Usually a link to the 'slocate' command

```
$ ls -l `which locate`
-rwx--s--x 1 root slocate 23856 Mar 14  2007 /usr/bin/locate
```

- Database can be created and updated via the 'updatedb' command
- Database location can vary with systems
  . /var/lib/mlocate/mlocate.db
  . /var/lib/slocate/slocate.db

```
LOCATE

NAME
       locate - find files by name

SYNOPSIS
       locate [OPTION]... PATTERN...

DESCRIPTION
       locate  reads  one  or more databases prepared by updatedb(8) and writes file names
matching at least one of the PATTERNs to standard output, one per line.

       PATTERNs can contains globbing characters.  If any PATTERN  contains  no  globbing
characters, locate behaves as if the pattern were *PATTERN*.
```

whatis

```
WHATIS

NAME
       whatis - search the whatis database for complete words.

SYNOPSIS
       whatis keyword ...

DESCRIPTION
       whatis  searches  a  set of database files containing short descriptions of system
commands for keywords and displays the result on the standard output.  Only complete word
matches  are  displayed.

       The whatis database is created using the command /usr/sbin/makewhatis.
```

apropos

```
APROPOS

NAME
       apropos - search the whatis database for strings

SYNOPSIS
       apropos keyword ...

DESCRIPTION
       apropos  searches  a set of database files containing short descriptions of system
commands for
       keywords and displays the result on the standard output.
```