Linux Essentials

Lesson 00:

Document History

Date	Course Version No.	Software Version No.	Developer / SME	Reviewer(s)	Approver	Change Record Remarks
11-August– 2016	1.0	5.5	Amal Thambi		Amal Thambi	Crested as per the Infra BU requirement

Course Goals

Course Goals

- **This course is designed for entry level Infra resources to enable them with the skills required for provisioning and maintaining a Linux Server in production environment.**

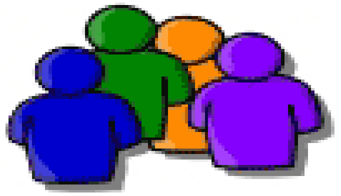
Pre-requisites

This course requires that you meet the following prerequisites:

- The audience should have completed IS Linux Fundamentals

Intended Audience

- For Entry level Infrastructure Services(IS) resources



Day Wise Schedule

- Day 1

Lesson 1: Basic of RHEL & Installation
Lesson 2: System Initialization
Lesson 3: Device Management
Lesson 4: Process Management

- Day 2

Lesson 5: System Services
Lesson 6: Kernel Management
Lesson 7: Bash Editing
Lesson 8: File System / Disk Management

- Day 3

Lesson 9: File Management
Lesson 10: Package Management
Lesson 11: User & Group Management
Lesson 12: File Permission

- Day 4

Lesson 13: Network Services
Lesson 14: Backup & Restore
Lesson 15: Troubleshooting
Lesson 16: Network Installation

Table of Contents

Table of Contents

Table of Contents

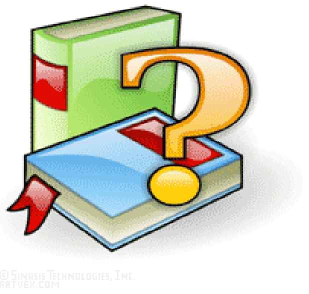
Table of Contents

Table of Contents

References

- **Books:**

RHCSA/RHCE Red Hat Linux Certification Study Guide (Exams EX300), 6th Edition - Certification Press



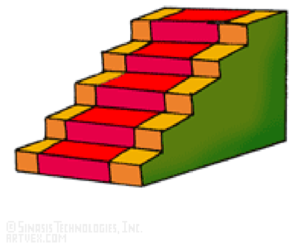
Red Hat Enterprise Linux 6 Administration: Real World Skills for Red Hat Administrators – Sybex Publications

- **Websites**

https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/5/html/Deployment_Guide/index.html

Next Step Courses

- **RHEL 7 Administration**



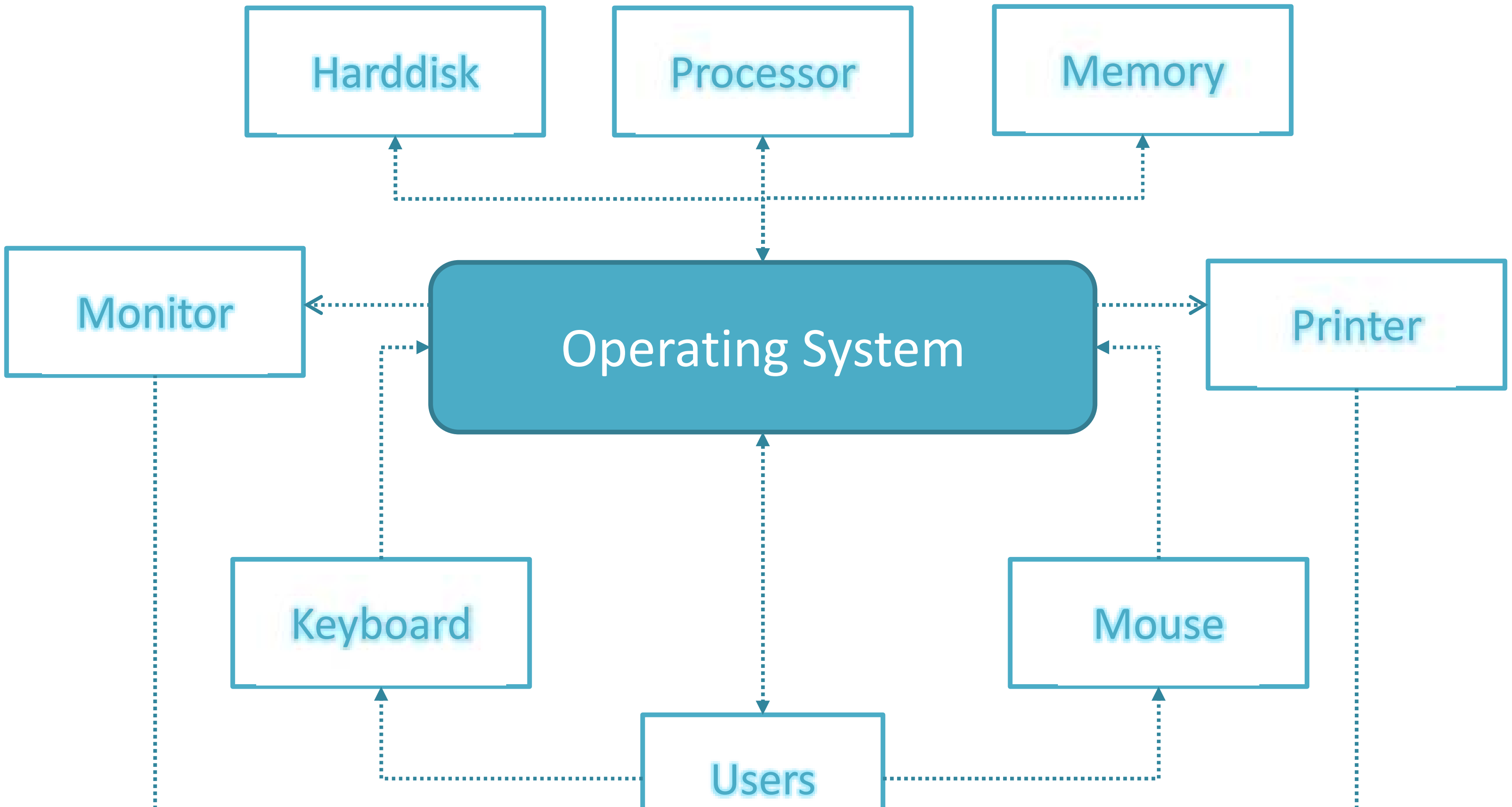
Other Parallel Technology Areas

- **Red Hat Linux Server**

Linux Fundamentals

Contents

Operating System - Recall



Software's and Licensing

Freeware	Shareware
Free with Limited Features	Free for limited time

Commercial	Open Source
Full / Partial features on cost	Free / Cost with source code

Open Source

1950’s and 60’s – Source was available

1960’s – Software’s became cost

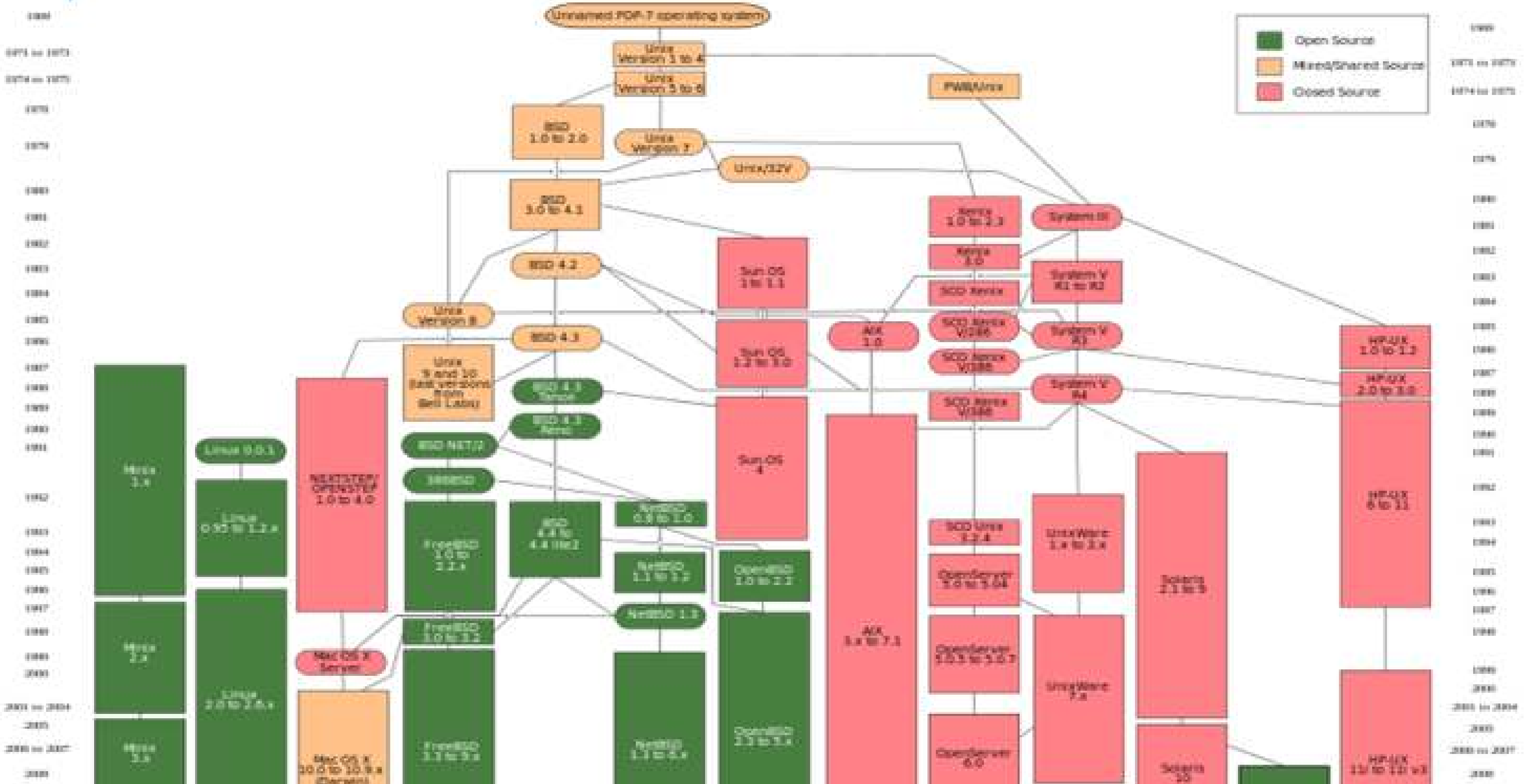
1983 – Free software movement, GNU Project

1990’s – Linux, Open source initiative (1997)

Open source licenses

License	License & Copyright Notice	Commercial Use	Disclosure of source	State changes	Distribution	Patent Use
ASL2.0	☑	☑	☒	☑	☑	☑
GNU GPL3.0	☑	☑	☑	☑	☑	☑
MIT	☑	☑	☒	☒	☑	☒

Unix



Open Source
Mixed/Shared Source
Closed Source
Unnamed PDP-7 operating system
Unix Version 1 to 4
Unix Version 5 to 6
PWB/Unix
BSD 1.0 to 2.0
Unix Version 7
Unix/32V
BSD 3.0 to 4.1
Xenix 1.0 to 2.3
System III
Xenix 3.0
SCO Xenix
System V R1 to R2
BSD 4.2
Sun OS 1 to 1.1
Unix Version 8
AIX 1.0
SCO Xenix V/286
System V R3
BSD 4.3
Sun OS 1.2 to 3.0
SCO Xenix V/386
HP-UX 1.0 to 1.2
Unix 9 and 10 (last versions from Bell Labs)
BSD 4.3 Tahoe
System V R4
HP-UX 2.0 to 3.0
BSD 4.3 Reno
SCO Xenix V/386
BSD NET/2
Linux 0.0.1
Minix 1.x
NEXTSTEP/OPENSTEP 1.0 to 4.0
386BSD
Sun OS 4
HP-UX 6 to 11
NetBSD 0.8 to 1.0
BSD 4.4 to 4.4 lite2
Linux 0.95 to 1.2.x
FreeBSD 1.0 to 2.2.x
SCO Unix 3.2.4
UnixWare 1.x to 2.x
NetBSD 1.1 to 1.2
OpenBSD 1.0 to 2.2
OpenServer 5.0 to 5.04
Solaris 2.1 to 9
AIX 3.x to 7.1
NetBSD 1.3
FreeBSD 3.0 to 3.2
Minix 2.x
Mac OS X Server
Linux 2.0 to 2.6.x
OpenServer 5.0.5 to 5.0.7
UnixWare 7.x
Minix 3.x
Mac OS X 10.0 to 10.9.x (Darwin)
FreeBSD 3.3 to 9.x
NetBSD 1.3 to 6.x
OpenBSD 2.3 to 5.x
OpenServer 6.0
Solaris 10
HP-UX 11i to 11i v3
1969
1971 to 1973
1974 to 1975
1976
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001 to 2004
2005
2006 to 2007
2008

Unix (contd.,)

Created in Bell Laboratories – 1960s

Re-Written in C Language - 1972

Multiple flavors evolved (HP-UX, Solaris, AIX) – 1980s

Linux, Open source BSD released – 1990s

Linux

MINIX was created – Andrew - 1987

Linus Torvalds developed a kernel – Linux - 1991

GNU Project

Free Software collaboration – 1983 – Richard Stallman

Initially – More software for Unix are developed

Shells, compilers, libraries are created for Linux

GNU GPL license evolved to 3.0

Linux Distributions

redhat

openSUSE

fedora

Live GNU/Linux System

KNOPPIX

CentOS

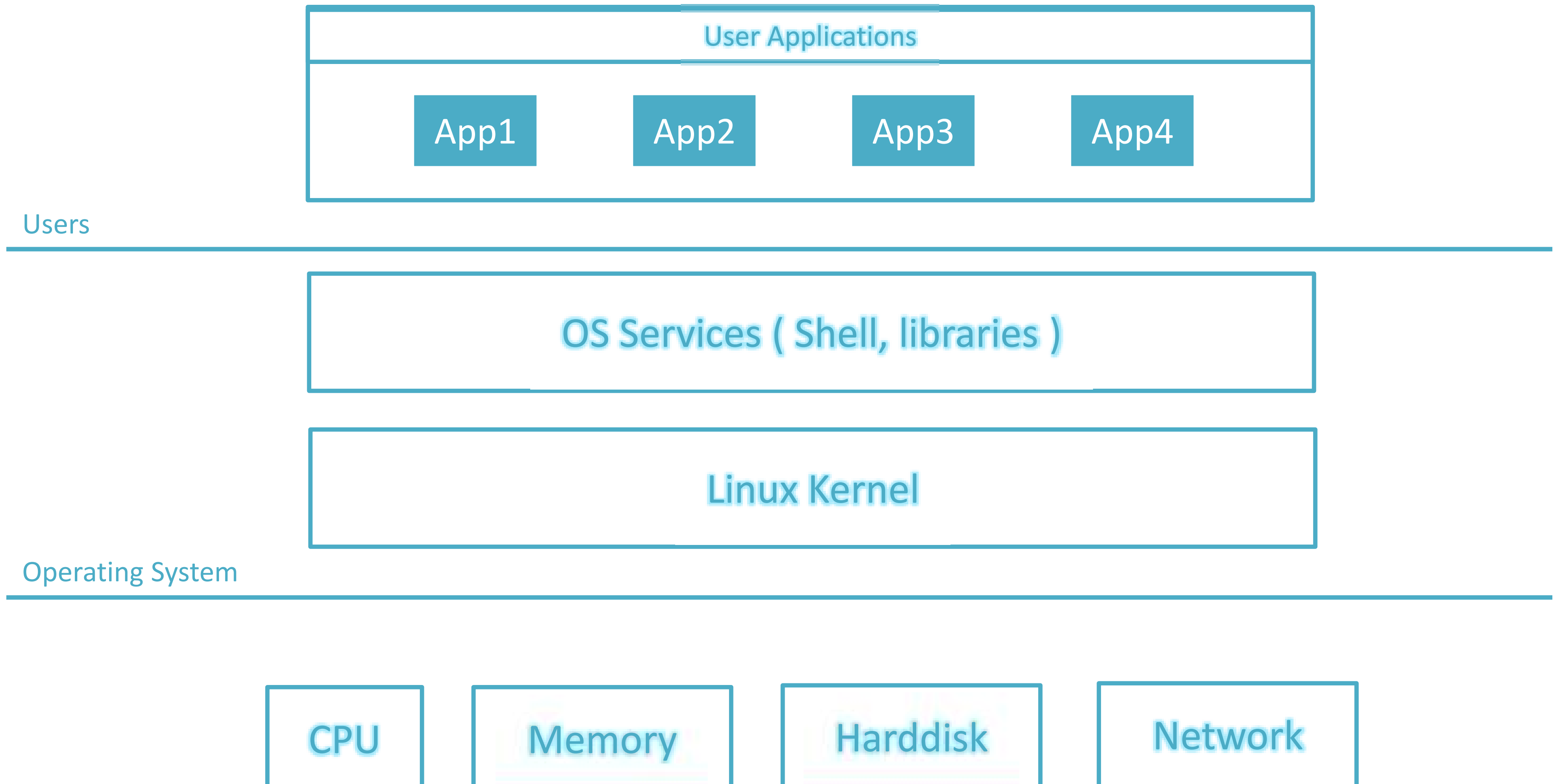
debian

GNU/Linux

ubuntu

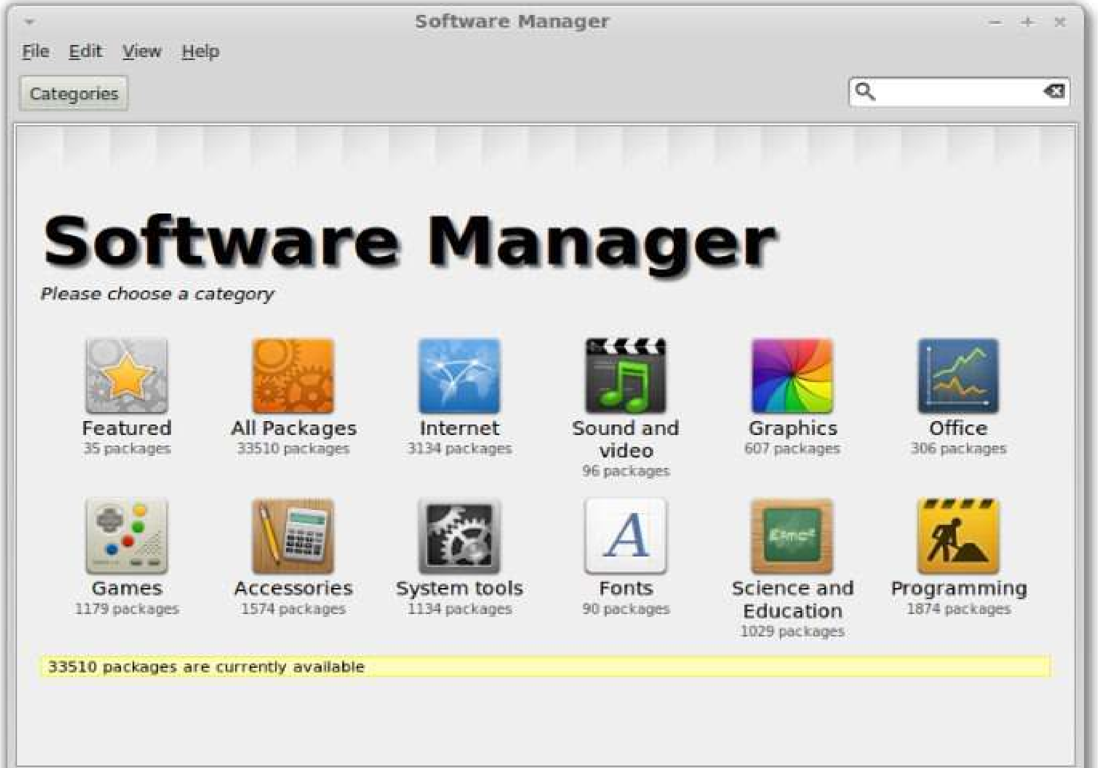
Linux Subsystems

Linux Subsystems



User Applications

Applications used by end users for processing



Software Manager

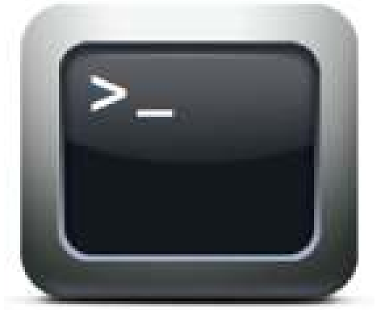
Please choose a category

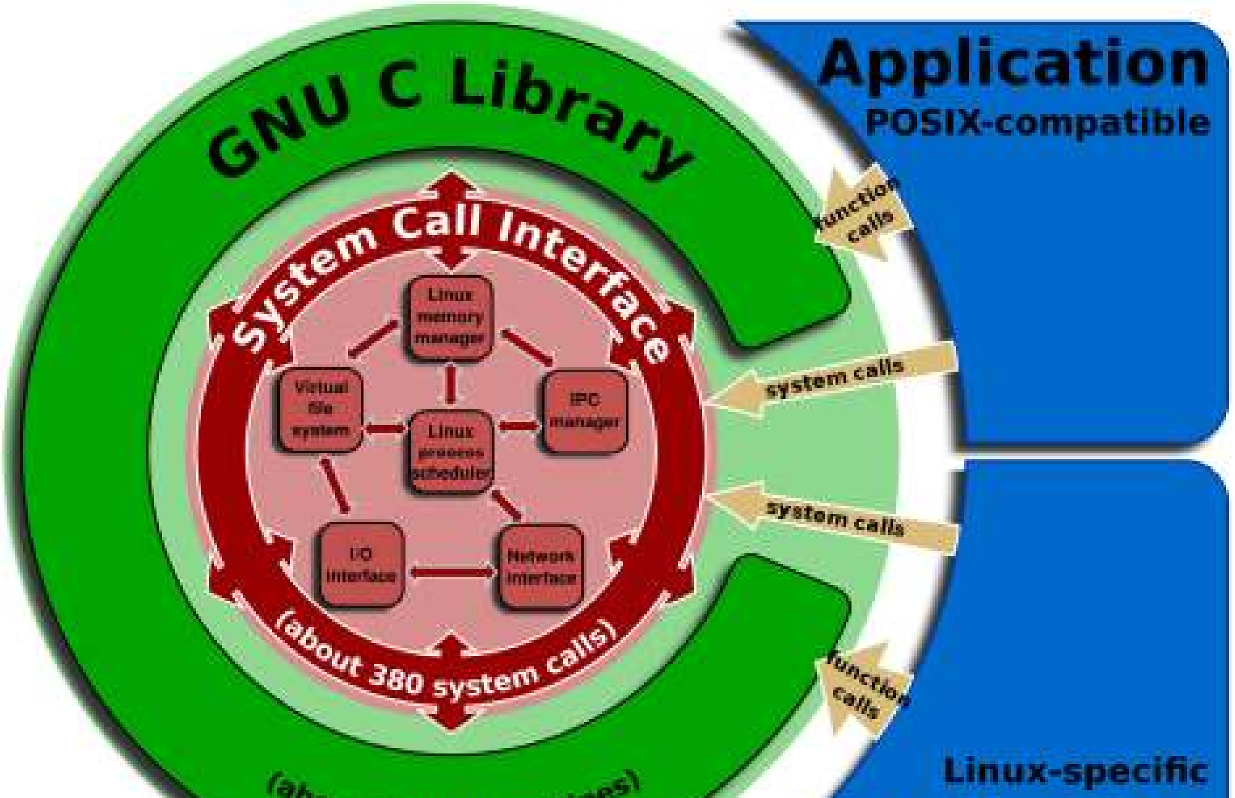
Featured 35 packages	All Packages 33510 packages	Internet 3134 packages	Sound and video 96 packages	Graphics 607 packages	Office 306 packages
Games 1179 packages	Accessories 1574 packages	System tools 1134 packages	Fonts 90 packages	Science and Education 1029 packages	Programming 1874 packages

33510 packages are currently available

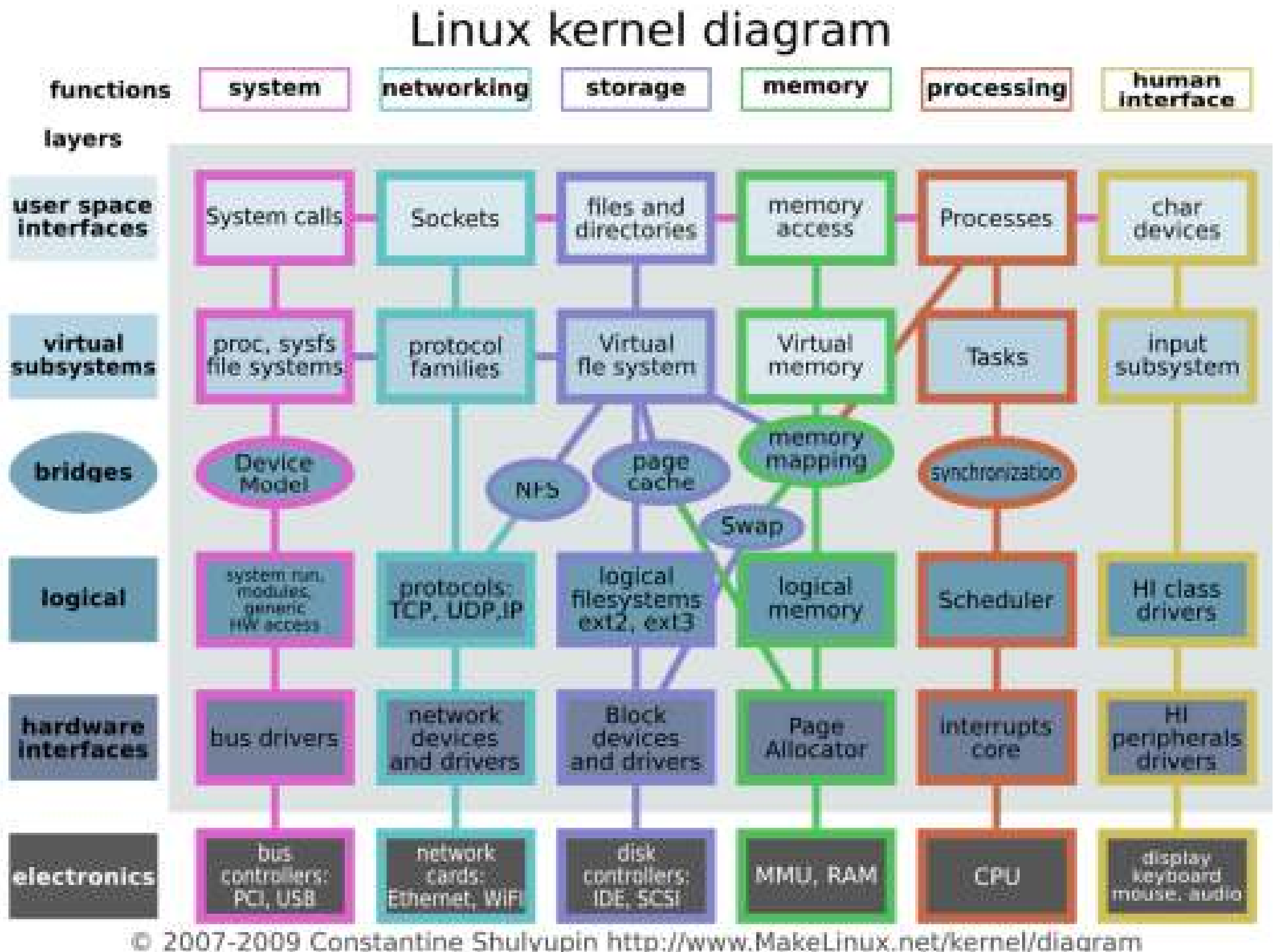
OS Services

Services which enables user applications to interact with kernel

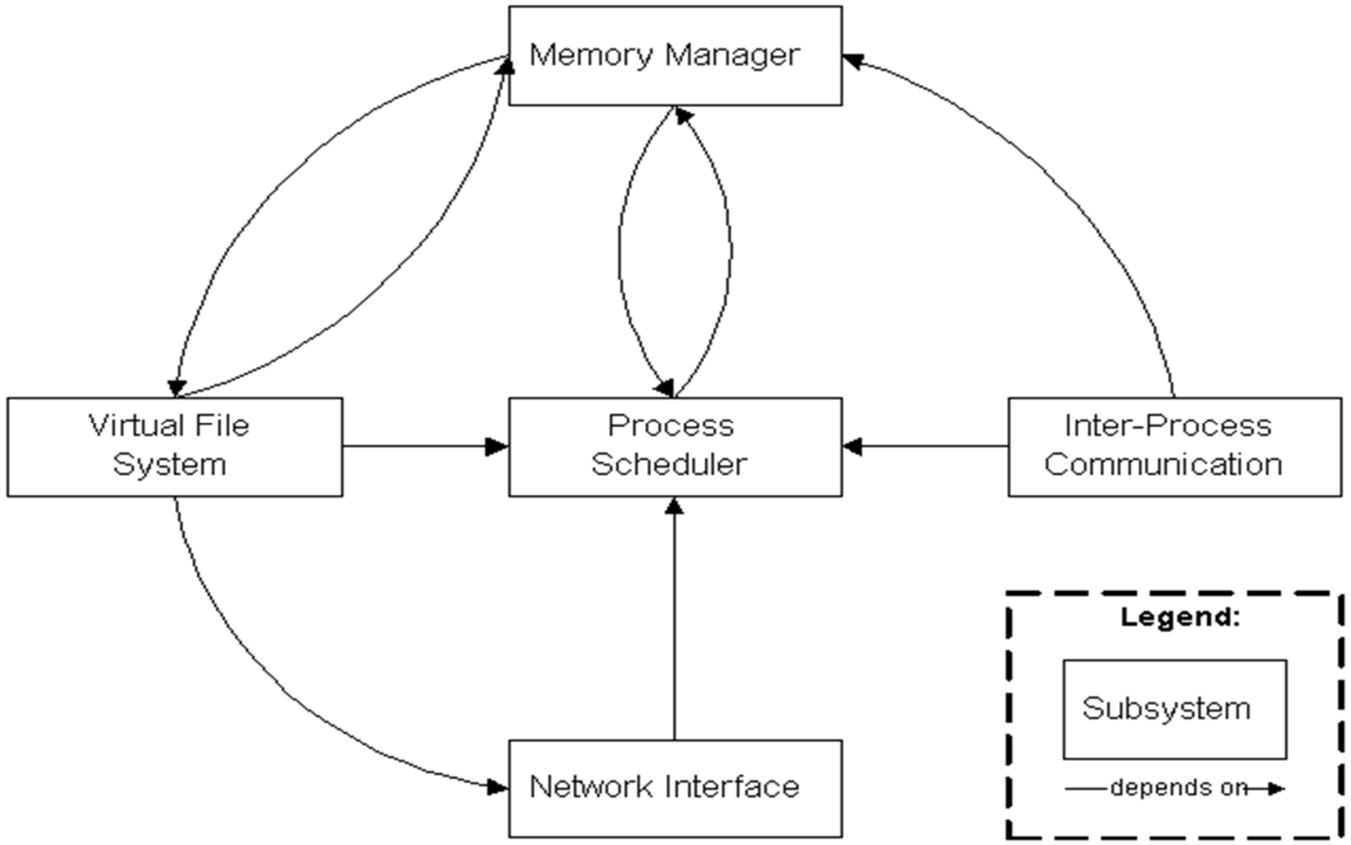




Linux Kernel



Linux Kernel (contd.,)



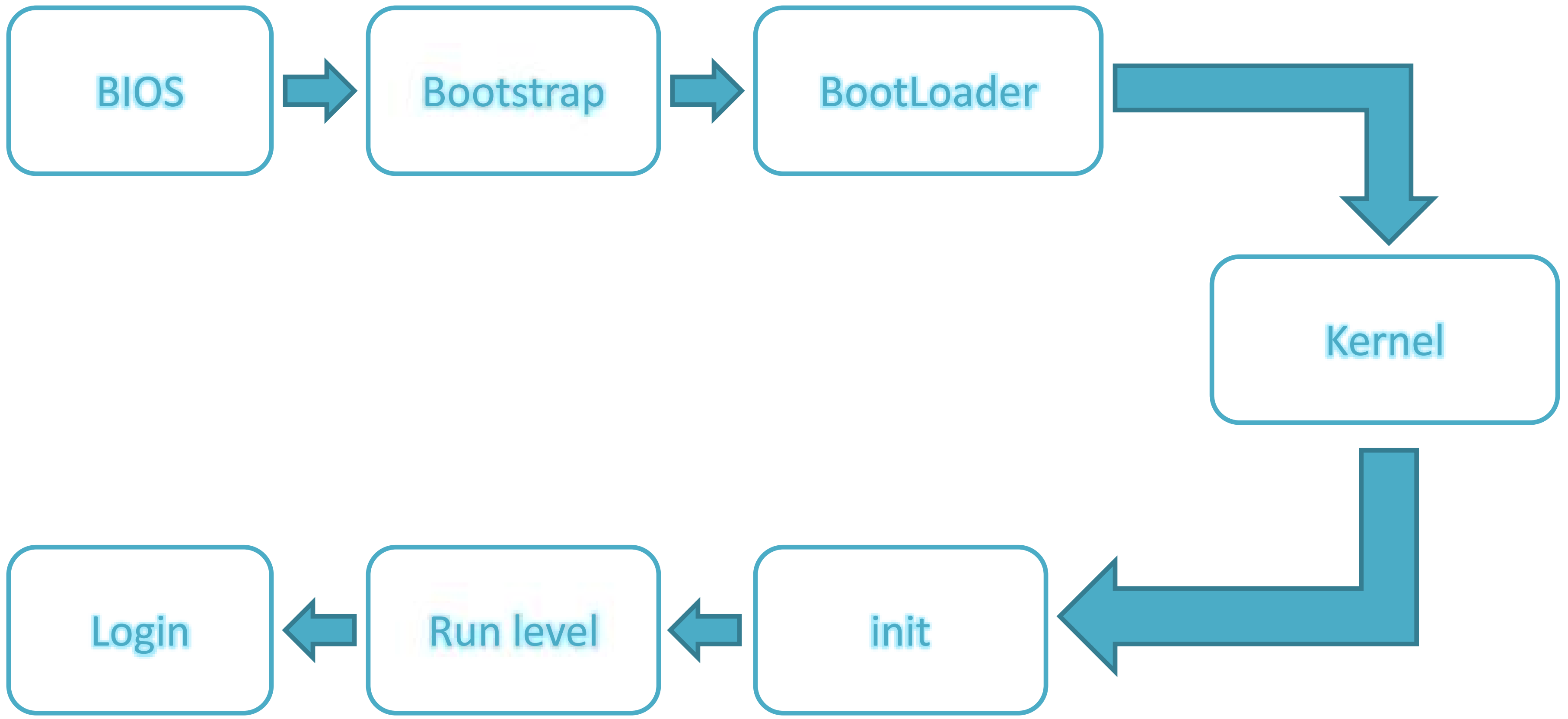
Hardware Controller

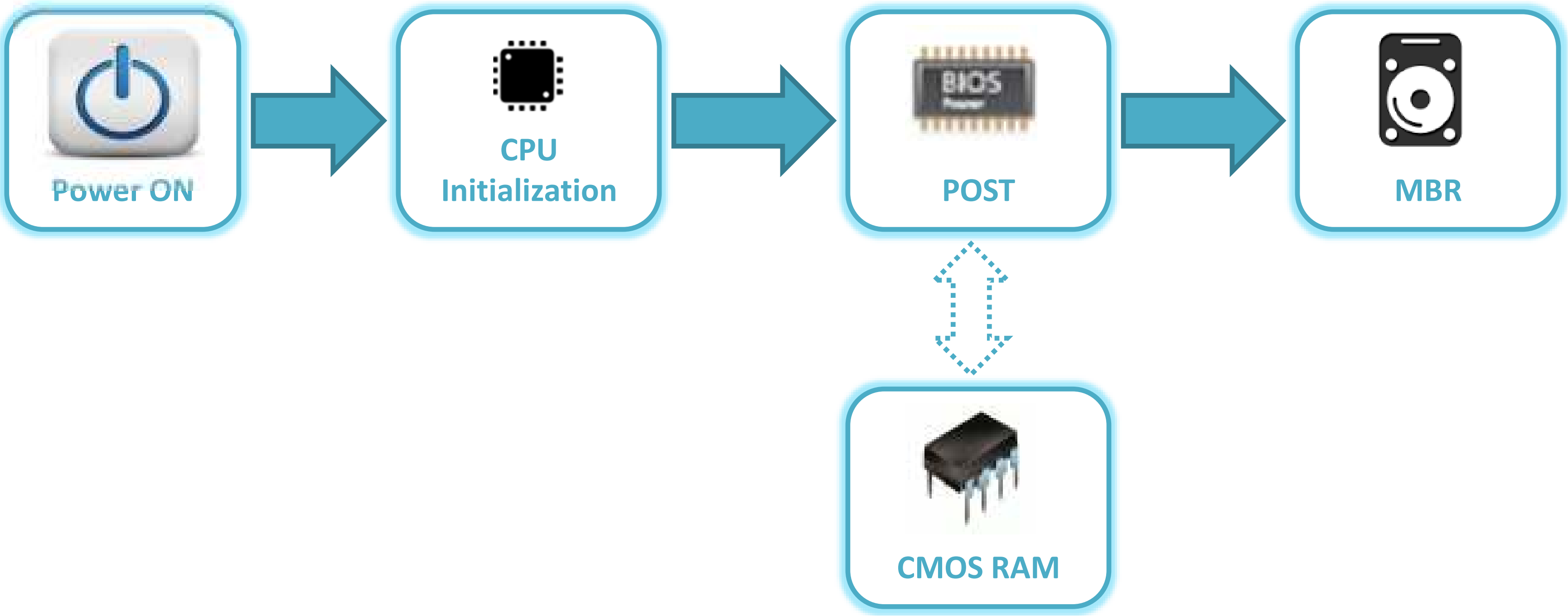
Devices – Controllers to take input from kernel

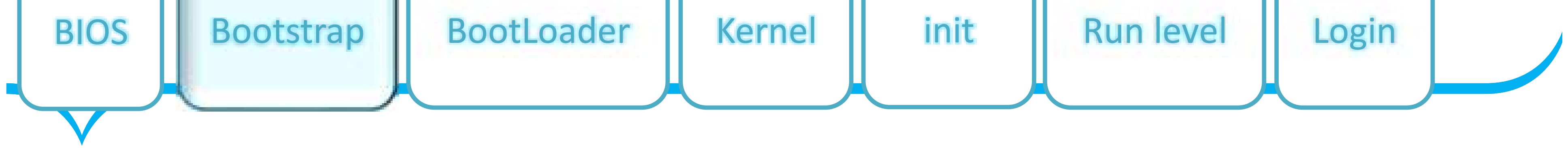
Kernel – Interfaces to connect and send commands

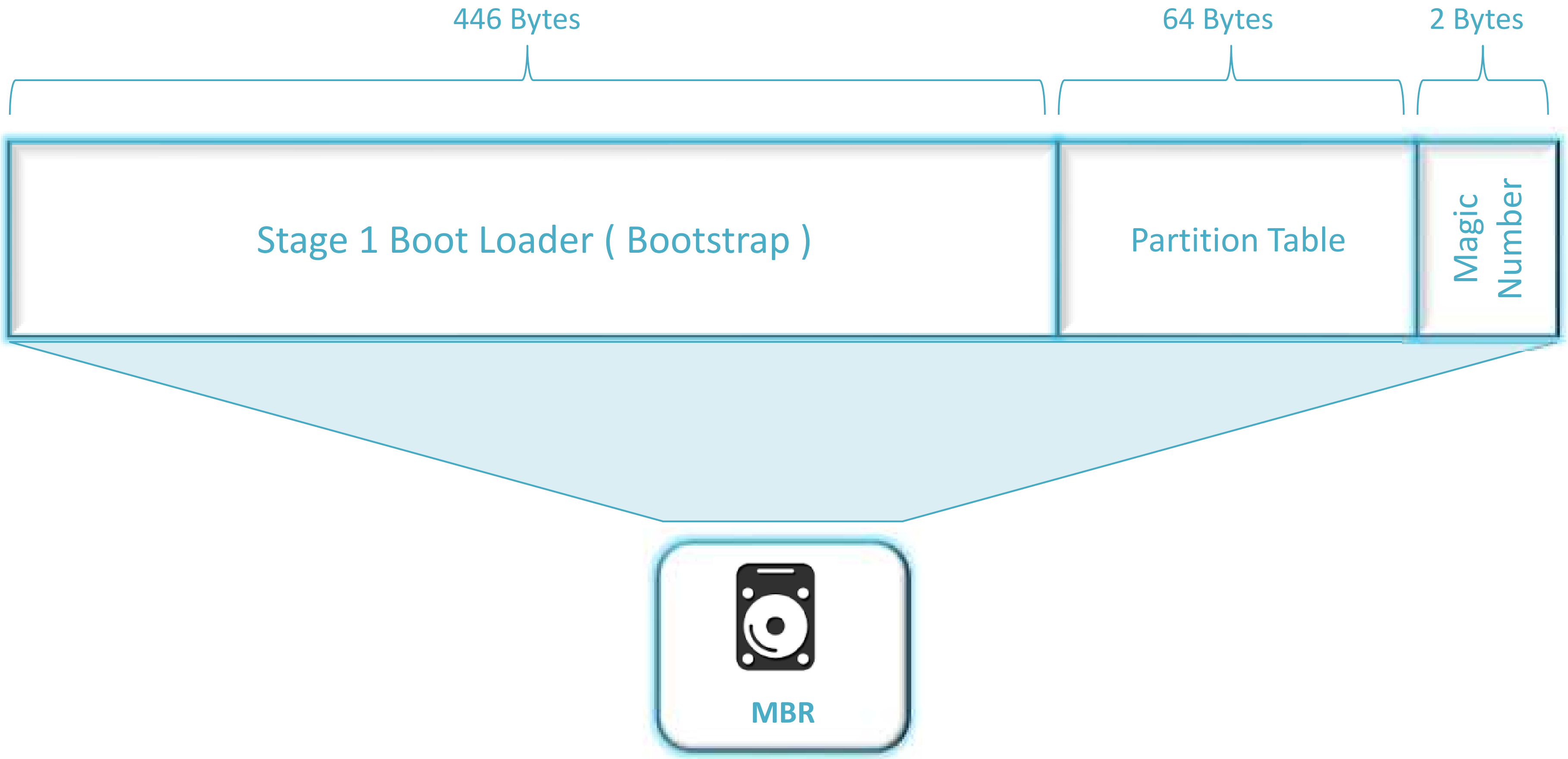
Booting Process

Booting Process

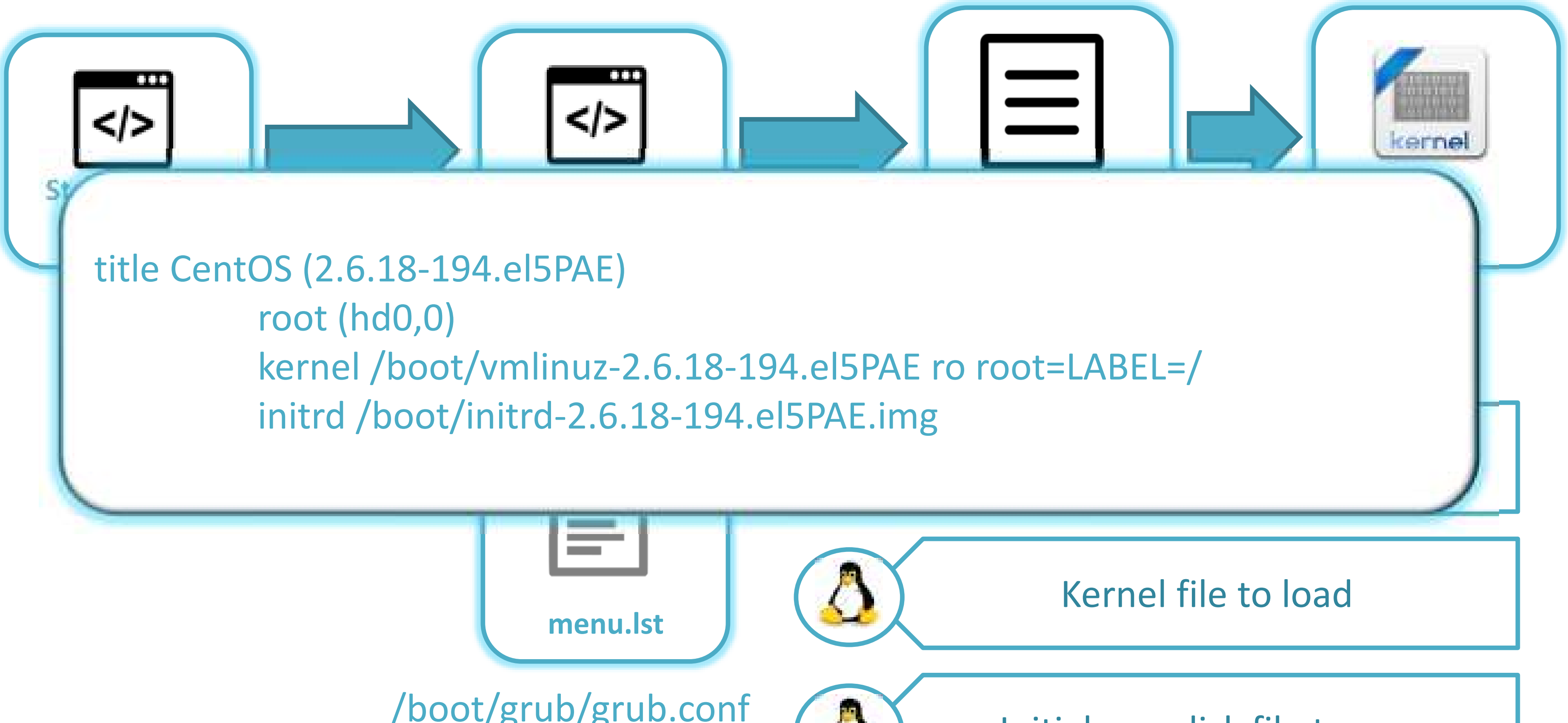



Power ON
CPU Initialization
BIOS
POST
MBR
CMOS RAM

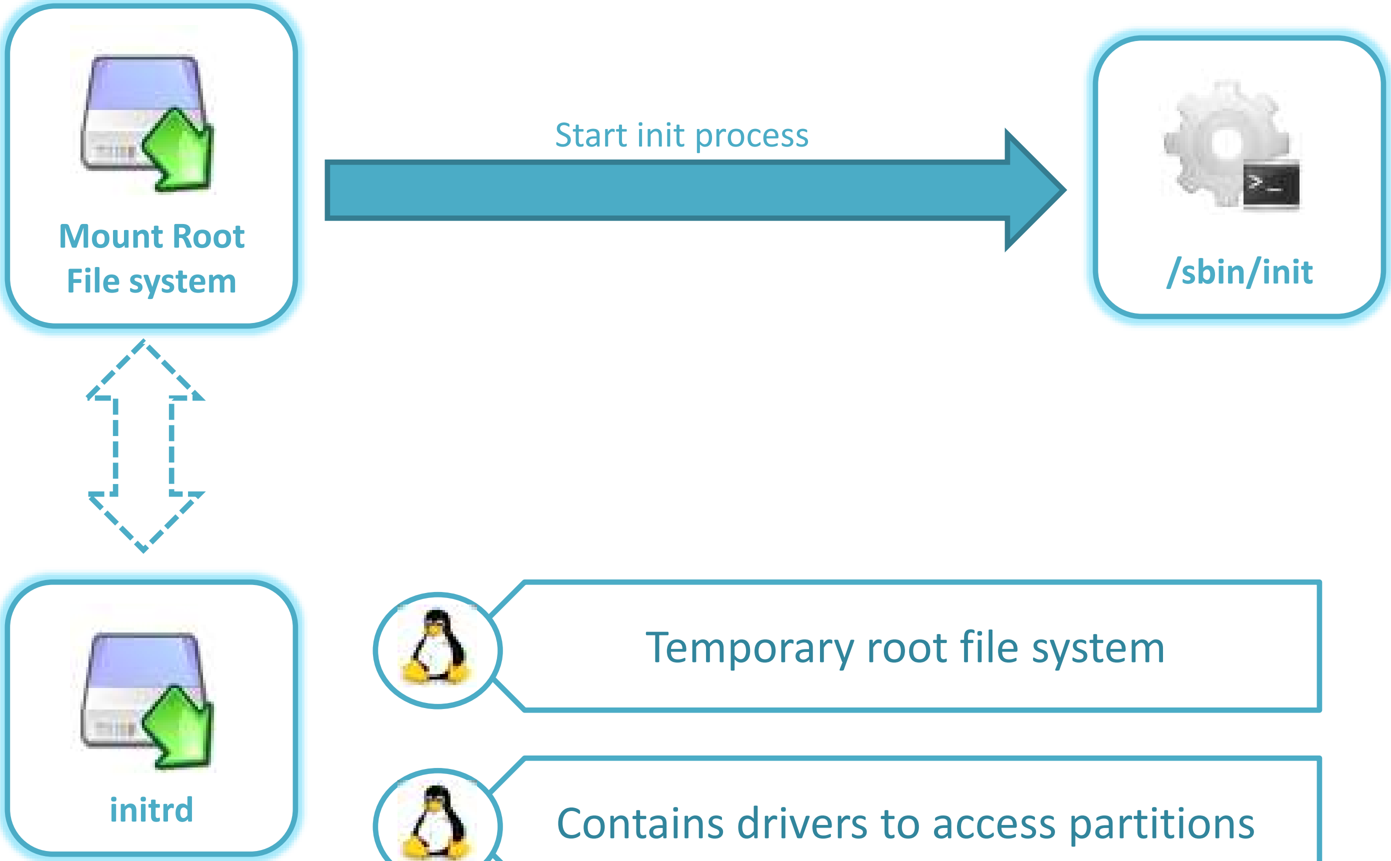

BIOS
Bootstrap
BootLoader
Kernel
init
Run level
Login


446 Bytes
64 Bytes
2 Bytes
Stage 1 Boot Loader (Bootstrap)
Partition Table
Magic Number
MBR

BIOS | Bootstrap | BootLoader | Kernel | init | Run level | Login

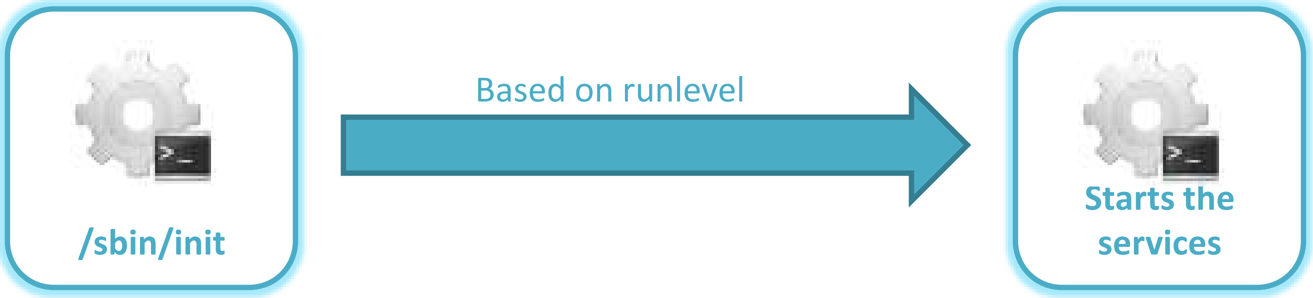


```
title CentOS (2.6.18-194.el5PAE)
          root (hd0,0)
          kernel /boot/vmlinuz-2.6.18-194.el5PAE ro root=LABEL=/
          initrd /boot/initrd-2.6.18-194.el5PAE.img
```



Temporary root file system

Contains drivers to access partitions



First process started with pid 1

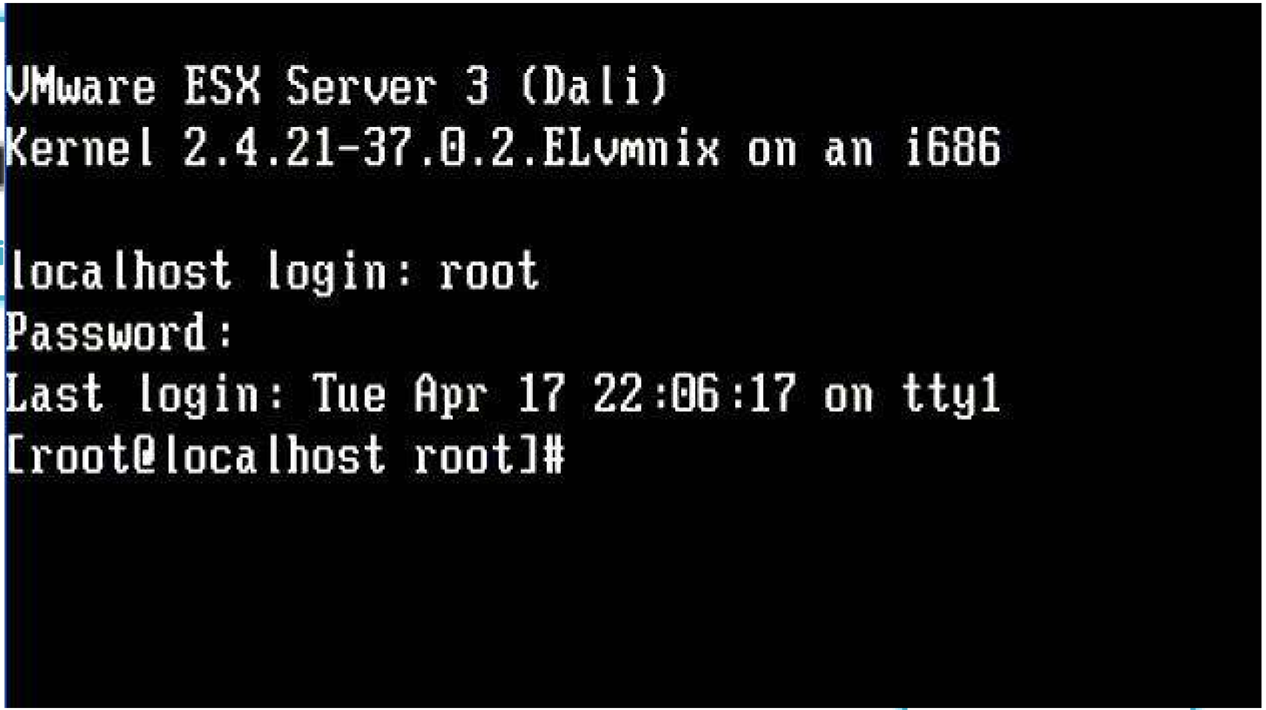
Uses /etc/inittab file to identify runlevel

BIOS | Bootstrap | BootLoader | Kernel | init | Run level | Login

- 0 – Halt (Shutdown)
- 1 – Single user mode
- 2 – Multiuser, without NFS
- 3 – Full multiuser mode (cmd line)
- 4 – Unused
- 5 – X11 (Gui mode)
- 6 – Reboot

- S – Start Scripts
- K – Kill Scripts

BIOS Bootstrap BootLoader Kernel init Run level Login



```
VMware ESX Server 3 (Dali)
Kernel 2.4.21-37.0.2.ELvmnix on an i686

localhost login: root
Password:
Last login: Tue Apr 17 22:06:17 on tty1
[root@localhost root]#
```

/etc/passwd
/etc/shadow

Shell

Linux Shell

- **Unix Shell :**

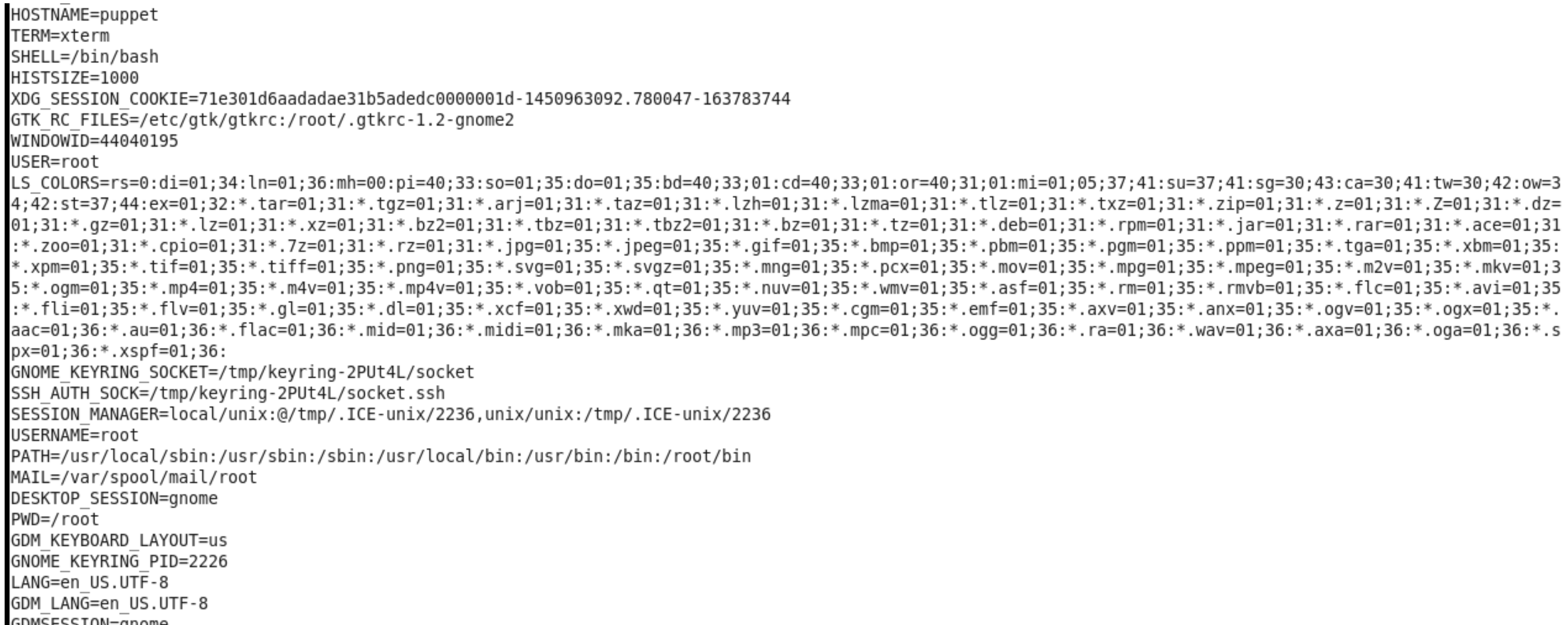
  - A Unix shell is a command-line interpreter .
  - It provides a traditional user interface for the Unix operating system and for Unix-like systems.
  - Whatever you type at the command line is understood and interpreted by a program and then that program gives you an output after executing your command. This program that understands what you type is called the shell.

- **Types of Shells :**

  - Ash
  - Bash
  - Corn
  - T - Shell
  - C- Shell

Linux Shell – Environment Variables

- **Environment variables are the built-in variables which contains a data that can be shared by 2 or more applications/processes.**



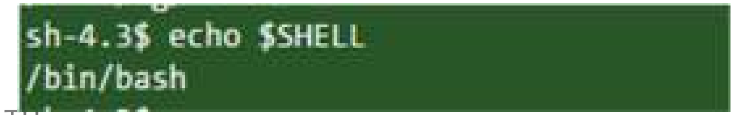
```
HOSTNAME=puppet
TERM=xterm
SHELL=/bin/bash
HISTSIZE=1000
XDG_SESSION_COOKIE=71e301d6aadadae31b5adedc0000001d-1450963092.780047-163783744
GTK_RC_FILES=/etc/gtk/gtkrc:/root/.gtkrc-1.2-gnome2
WINDOWID=44040195
USER=root
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=01;05;37;41:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=3
4;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arj=01;31:*.taz=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=
01;31:*.gz=01;31:*.lz=01;31:*.xz=01;31:*.bz2=01;31:*.tbz=01;31:*.tbz2=01;31:*.bz=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.rar=01;31:*.ace=01;31
:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:
*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;3
5:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35
:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.axv=01;35:*.anx=01;35:*.ogv=01;35:*.ogx=01;35:*.
aac=01;36:*.au=01;36:*.flac=01;36:*.mid=01;36:*.midi=01;36:*.mka=01;36:*.mp3=01;36:*.mpc=01;36:*.ogg=01;36:*.ra=01;36:*.wav=01;36:*.axa=01;36:*.oga=01;36:*.s
px=01;36:*.xspf=01;36:
GNOME_KEYRING_SOCKET=/tmp/keyring-2PUt4L/socket
SSH_AUTH_SOCK=/tmp/keyring-2PUt4L/socket.ssh
SESSION_MANAGER=local/unix:@/tmp/.ICE-unix/2236,unix/unix:/tmp/.ICE-unix/2236
USERNAME=root
PATH=/usr/local/sbin:/usr/sbin:/sbin:/usr/local/bin:/usr/bin:/bin:/root/bin
MAIL=/var/spool/mail/root
DESKTOP_SESSION=gnome
PWD=/root
GDM_KEYBOARD_LAYOUT=us
GNOME_KEYRING_PID=2226
LANG=en_US.UTF-8
GDM_LANG=en_US.UTF-8
GDMSESSION=gnome
```

Linux Shell – Environment Variables

- **Some of the important environment variables are:**
  - PWD

```
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/bin/X1
```

  - SHELL

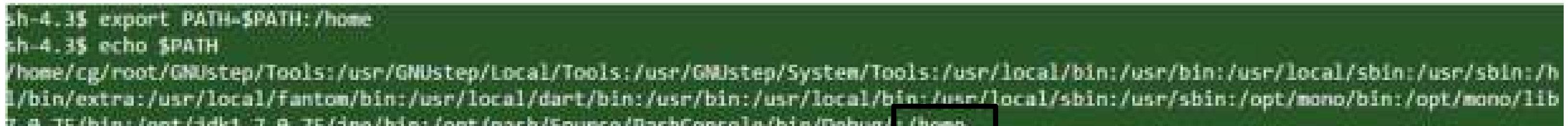


```
sh-4.3$ echo $SHELL
/bin/bash
```

  - PATH

```
root@tuxworld:~# echo $PWD
/root
```

- **SETTING ENVIRONMENT VARIABLES:**



```
sh-4.3$ export PATH=$PATH:/home
sh-4.3$ echo $PATH
/home/cg/root/GNUstep/Tools:/usr/GNUstep/Local/Tools:/usr/GNUstep/System/Tools:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/h
l/bin/extra:/usr/local/fantom/bin:/usr/local/dart/bin:/usr/bin:/usr/local/bin:/usr/local/sbin:/usr/sbin:/opt/mono/bin:/opt/mono/lib
```

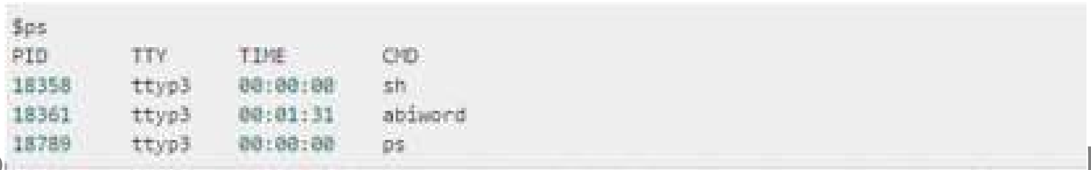
Commands for Practice

1 alias
2 aspell
3 at
4 atq
5 cal
6 cat
7 cd
8 cp
9 cut
10 date
11 dd
12 df
13 diff
14 dmesg
15 echo
16 eject
17 emacs
18 fdisk
19 find
21 grep
22 Halt
23 head
24 History
25 hostname
26 ifconfig
27 ll
28 ln
29 locate
30 login
31 logout
32 ls
33 mkdir
34 mke2fs
35 more
36 mount
37 mv
38 poweroff
39 put
41 reboot
42 rm
43 scp
44 sed
45 sh
46 shutdown
47 sort
48 su
49 tail
50 touch
51 umount
52 uname
53 uniq
54 uptime
55 vi
56 vim
57 wc
58 whereis
59 which

Process Management

Linux Process Management

- **PROCESS:**
  - An instance of a program is called a Process. In simple terms, any command that you give to your Linux machine starts a new process.



```
$ps
PID       TTY      TIME        CMD
18358     ttyp3    00:00:00    sh
18361     ttyp3    00:01:31    abiword
18789     ttyp3    00:00:00    ps
```

- **The operating system tracks processes through a five digit ID number known as the pid or process ID . Each process in the system has a unique pid.**

Linux Process Management (Contd…)

TYPES OF PROCESS:

- **Child Process:** A process that is created by some other process during run-time. Usually child processes are created to execute some binary from within an existing process.
- **Daemon Process:** These are special processes that run in background. They are system related process that have no associated terminal.
- **Orphan Process:** When parent process gets killed the child processes become orphan and then taken under by the init process. Though the init process takes the ownership of the orphan process but still these process are called as orphan as their original parents no longer exists.
- **Zombie Process :** A zombie process is one that should have closed, but is still active in the process table. This is usually caused when a parent process that spawned the process has not yet realized that it has completed, or wants to create another process of the same name without using the same process ID.
- **Interactive process :** These interact constantly with their users, and therefore spend a lot of time waiting for key presses and mouse operations.
- **Batch or Automatic process :** These do not need user interaction, and hence they often run in the background.
- **Real time process :** These have very strong scheduling requirements. They should have a short response time and, most important, such response time should have a minimum variance. Typical real-time programs are video and sound applications.

Linux Process Management

Process Scheduling :

- The scheduler is the component of the kernel that selects which process to run next.
- Scheduling refers to the way processes are assigned to run on the available CPUs

Functionality of Scheduler :

- **CPU utilization:** To keep the CPU as busy as possible.
- **Throughput** : Number of processes that complete their execution per time unit.
- **Turnaround** : Total time between submission of a process and its completion.
- **Waiting time** : Amount of time a process has been waiting in the ready queue.
- **Response time:** Amount of time it takes from when a request was submitted until the first response is produced.
- **Fairness** : Equal CPU time to each thread.

Linux Process Management (Contd…)

Types of Scheduling :

- **Normal : Referred to as other, this is the scheduling type set for normal programs**
- **FIFO : This is a real time scheduling priority. The FIFO term means the first started (first in) will be the first done (first out).**
- **RR : This is a round robin type of scheduling, where each task gets a certain amount of time then it must exit, yield control to the next task and get back into the task queue. This is a real time scheduling priority.**
- **Priority Based Scheduling : Assigns each process a priority, and scheduler always chooses process of higher priority over one of lower priority .**
- **Shortest Job First(SJF) : Itis a non-preemptive discipline in which waiting job (or process) with the smallest estimated run-time-to-completion is run next**

Linux Process Management (Contd…)

Inter Process Management :

- **It is a set of techniques for the exchange of data among multiple threads in one or more processes.**

Types of Inter Process Management :

- **Signals : Sent by other processes or the kernel to a specific process to indicate various conditions.**
- **Pipes : Unnamed pipes set up by the shell normally with the "|" character to route output from one program to the input of another.**
- **Sockets: A socket is one endpoint of a two-way communication link between two programs running on the network.**

Linux Process Management (Contd…)

- **Message queues : Message queues are a mechanism set up to allow one or more processes to write messages that can be read by one or more other processes.**
- **Semaphores : Counters that are used to control access to shared resources. These counters are used as a locking mechanism to prevent more than one process from using the resource at a time.**
- **Shared memory : The mapping of a memory area to be shared by multiple processes**

File System

Linux File System

- **What is a File?**

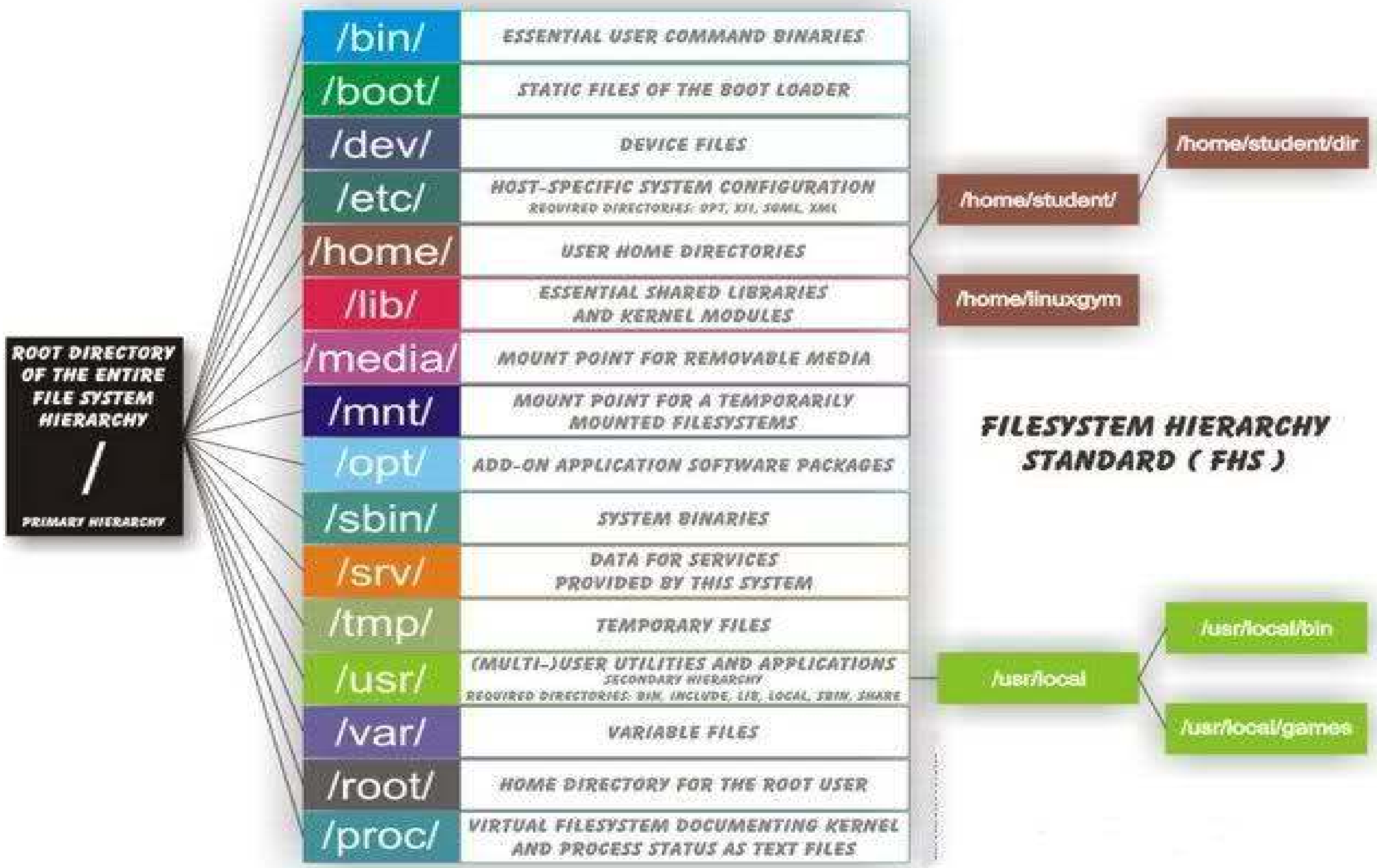
  File is a collection of data items stored on disk. Or it's device which can store the information, data, music (mp3), picture, movie, sound, book etc. In fact what ever you store in computer it must be inform of file. Files are always associated with devices like hard disk ,floppy disk etc. File is the last object in your file system tree.

- **What is a directory?**

  Directory is group of files

  - Root directory - It is root of your entire file system and can not be renamed or deleted which is denoted by / (forward slash)
  - Sub directory - Directory under root (/) directory is subdirectory which can be created, renamed by the user.

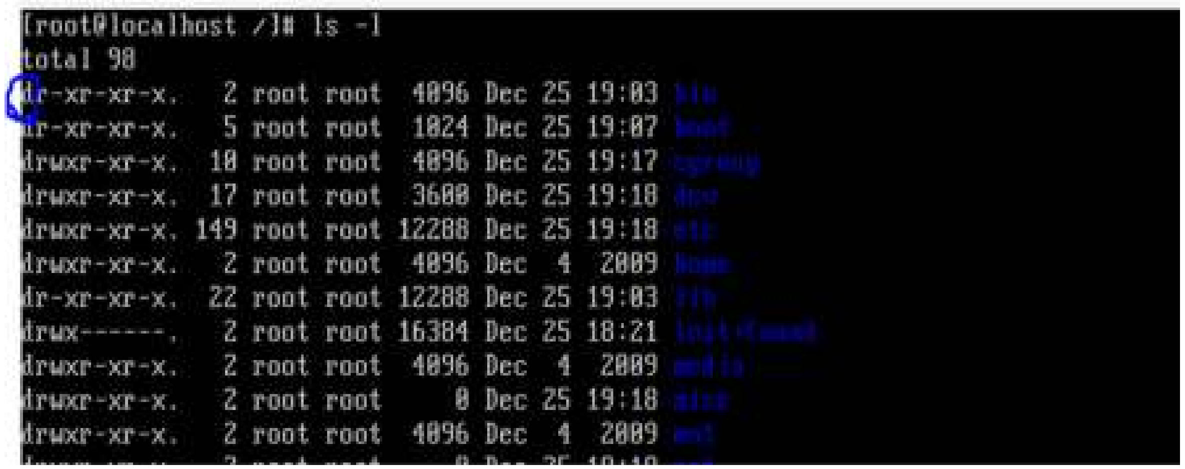
Linux Directory Structure



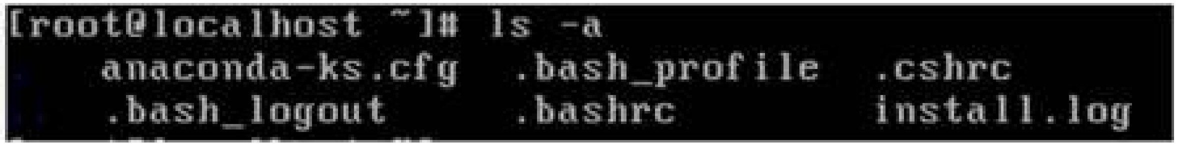
/bin/	ESSENTIAL USER COMMAND BINARIES
/boot/	STATIC FILES OF THE BOOT LOADER
/dev/	DEVICE FILES
/etc/	HOST-SPECIFIC SYSTEM CONFIGURATION REQUIRED DIRECTORIES: OPT, X11, SGML, XML
/home/	USER HOME DIRECTORIES
/lib/	ESSENTIAL SHARED LIBRARIES AND KERNEL MODULES
/media/	MOUNT POINT FOR REMOVABLE MEDIA
/mnt/	MOUNT POINT FOR A TEMPORARILY MOUNTED FILESYSTEMS
/opt/	ADD-ON APPLICATION SOFTWARE PACKAGES
/sbin/	SYSTEM BINARIES
/srv/	DATA FOR SERVICES PROVIDED BY THIS SYSTEM
/tmp/	TEMPORARY FILES
/usr/	(MULTI-)USER UTILITIES AND APPLICATIONS SECONDARY HIERARCHY REQUIRED DIRECTORIES: BIN, INCLUDE, LIB, LOCAL, SBIN, SHARE
/var/	VARIABLE FILES
/root/	HOME DIRECTORY FOR THE ROOT USER
/proc/	VIRTUAL FILESYSTEM DOCUMENTING KERNEL AND PROCESS STATUS AS TEXT FILES

Types of Files

- **Regular Files**
- **Directory Files**
- **Special Files**
  - Block file
  - Character Device File
  - Named Pipe File
  - Symbolic File
  - Socket File

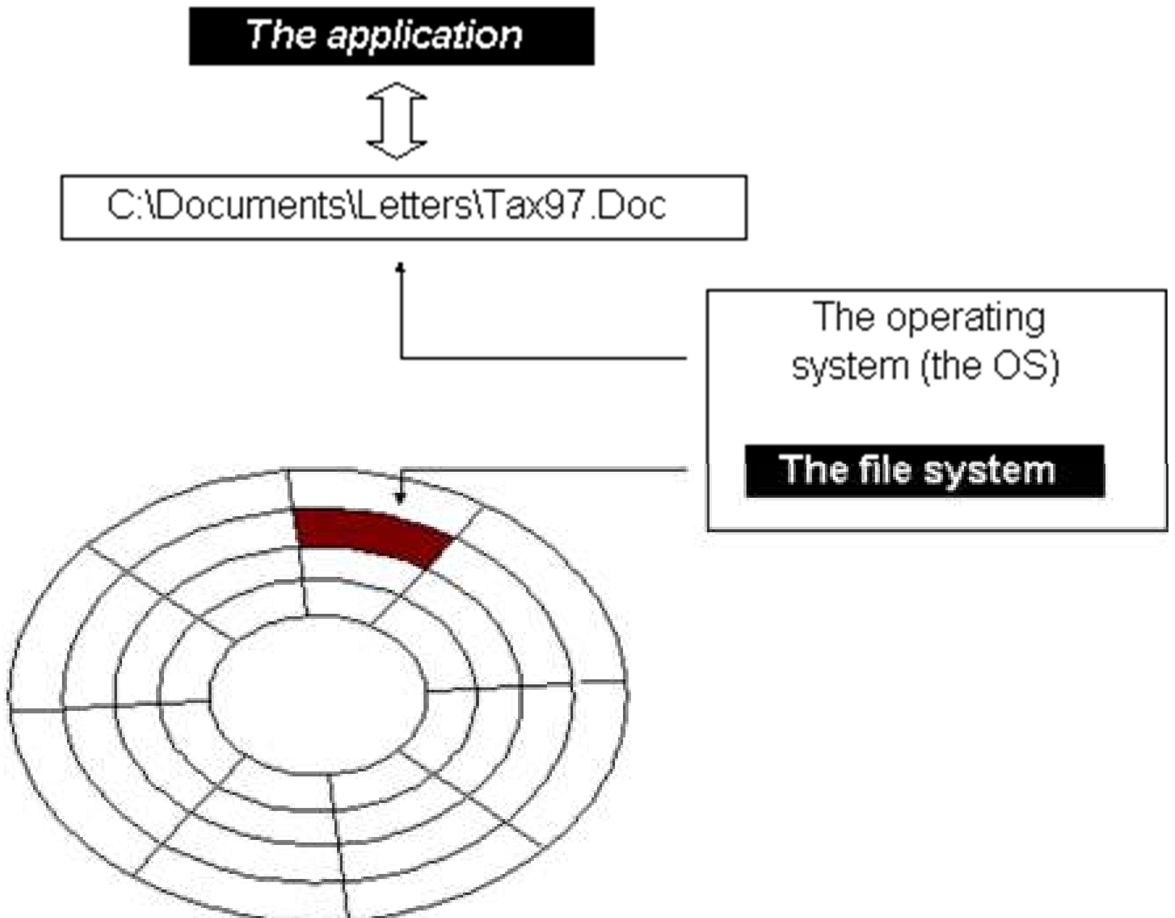


```
[root@localhost /]# ls -l
total 98
dr-xr-xr-x.   2 root root  4096 Dec 25 19:03 bin
dr-xr-xr-x.   5 root root  1024 Dec 25 19:07 boot
drwxr-xr-x.  10 root root  4096 Dec 25 19:17 cgroup
drwxr-xr-x.  17 root root  3600 Dec 25 19:18 dev
drwxr-xr-x. 149 root root 12288 Dec 25 19:18 etc
drwxr-xr-x.   2 root root  4096 Dec  4  2009 home
dr-xr-xr-x.  22 root root 12288 Dec 25 19:03 lib
drwx------.   2 root root 16384 Dec 25 18:21 lost+found
drwxr-xr-x.   2 root root  4096 Dec  4  2009 media
drwxr-xr-x.   2 root root     0 Dec 25 19:18 misc
drwxr-xr-x.   2 root root  4096 Dec  4  2009 mnt
```



```
[root@localhost ~]# ls -a
    anaconda-ks.cfg   .bash_profile   .cshrc
    .bash_logout      .bashrc         install.log
```

Basics of File System



Types of File System

- **EXT**
  - The Extended file system is used to on the storage media like hard disks and default file system in linux
- **CDFS :**
  - A file system that is used on compact disks to provide access to individual data and audio tracks
- **UFS**
  - The Unix File System is used various versions of unix like BSD and Solaris

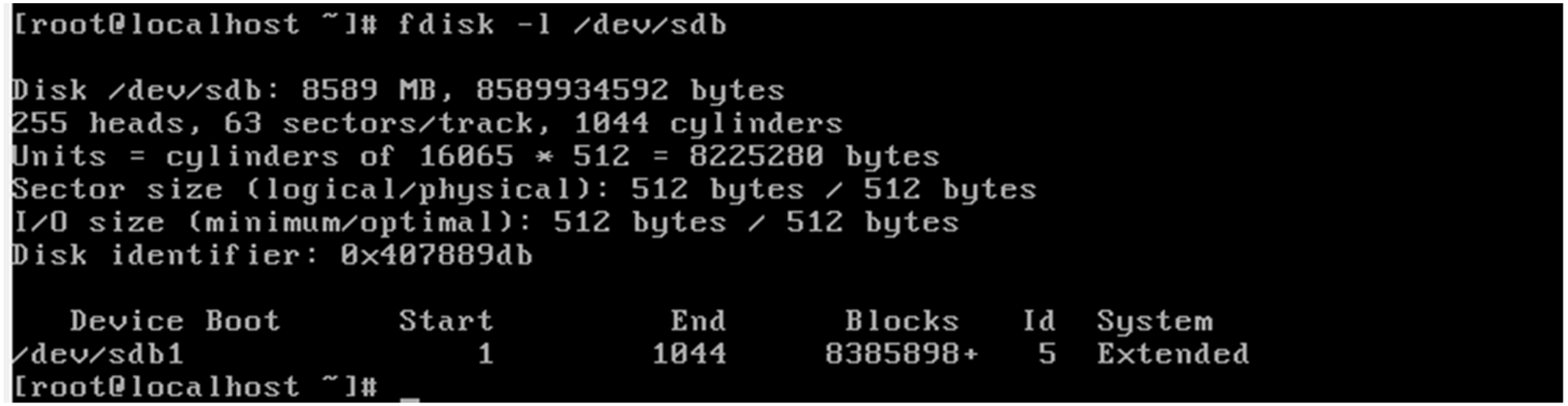
Disk Partitioning

- Disk partitioning is dividing the total storage of a drive into different small parts called partitions
- The partitions will be formatted with the specified file system so it can be used for storing the data

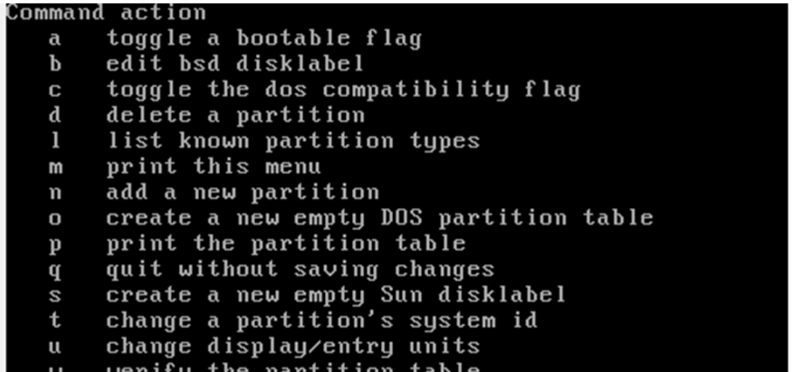
- Advantages:
  - Multiple File Systems
  - Partition Size
  - Multiple Operating Systems
  - Separate system files from users files

Disk Partitioning

- We can perform the disk partitioning by using the “fdisk” command



```
[root@localhost ~]# fdisk -l /dev/sdb

Disk /dev/sdb: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x407889db

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               1        1044     8385898+   5  Extended
[root@localhost ~]# _
```



```
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
```

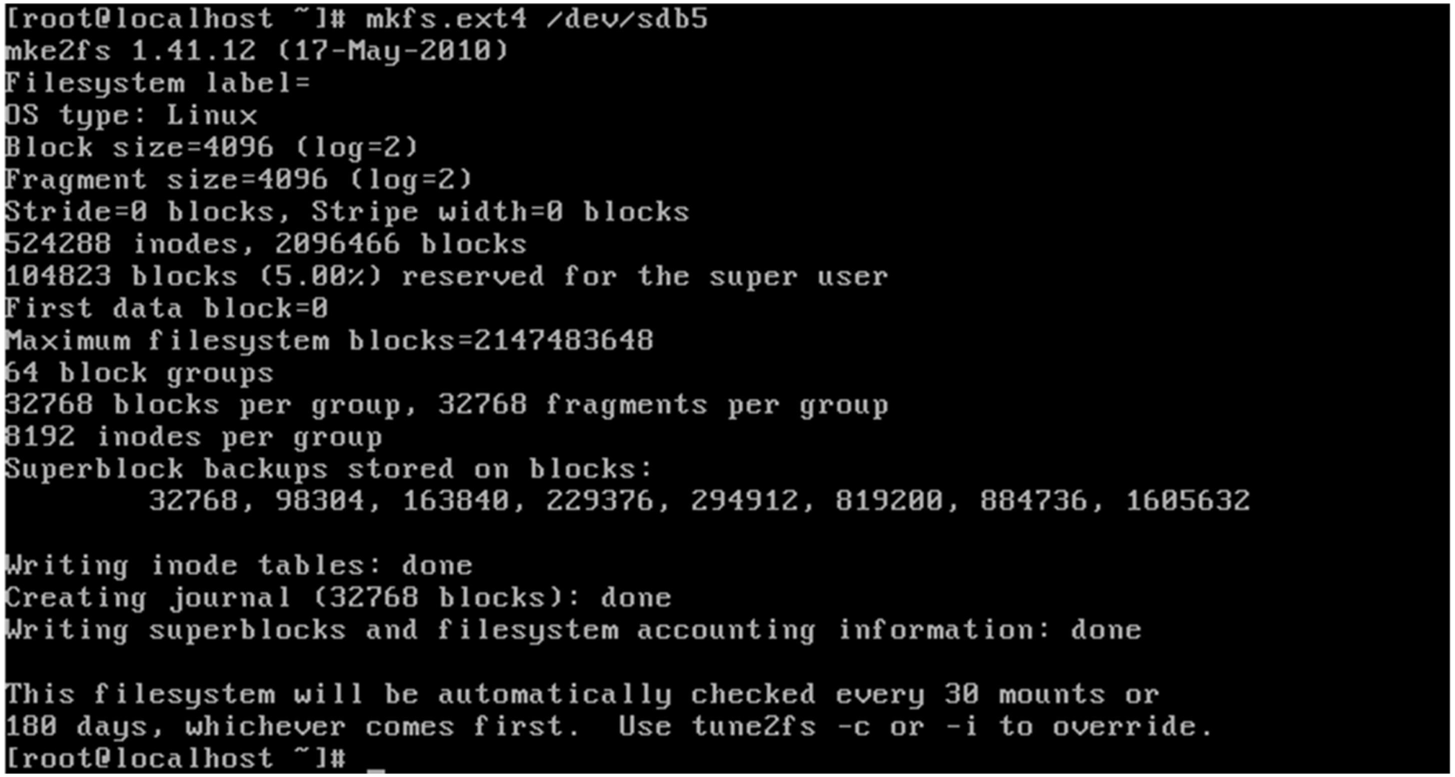
Formatting the Partition

- Formatting is the process of preparing a data storage device such as a hard disk drive, solid-state drive, floppy disk or USB flash drive for initial use

- Types of Formatting:

Hard Formatting	Soft Formatting
• Forming the tracks and sectors on the device • Done at manufacturer	• Creating the file system on the device to store the data • Done at OS level

Formatting a drive using EXT4 file system

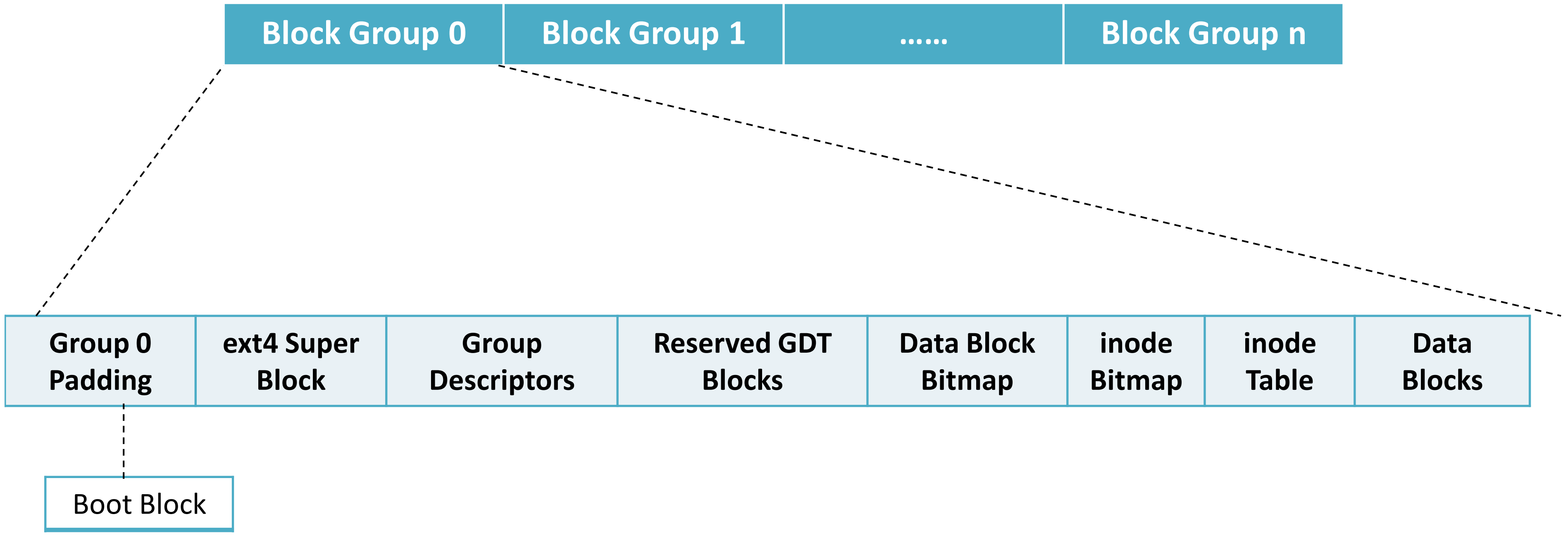


```
[root@localhost ~]# mkfs.ext4 /dev/sdb5
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
524288 inodes, 2096466 blocks
104823 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2147483648
64 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 30 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
[root@localhost ~]# _
```

EXT4 File System

- Fourth extended file system(EXT4) developed as successor of Ext3
- An Ext4 file system is split into a series of block groups
- Ext4 uses extents which improves performance when using large files and reduces metadata overhead for large files

Structure of an EXT4 File System

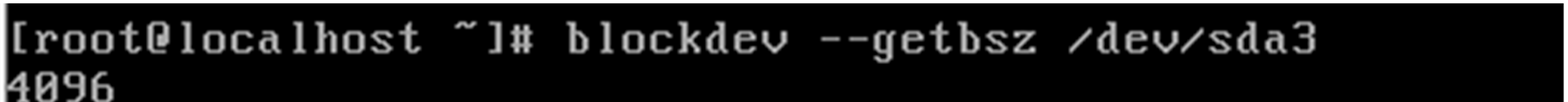


Blocks

- A block is a group of sectors between 1KiB and 64KiB
- The number of sectors must be an integral power of 2
- Default block size is 4096 bytes
- By default a file system can contain
  - » if 32-bit : 2^32 blocks
  - » if 64-bit : 2^64 blocks

- We can set the block size while creating the file system

```
$ mkfs.ext4 -b 4096 /dev/sdb1
```



```
[root@localhost ~]# blockdev --getbsz /dev/sda3
4096
```

Block Groups

- Blocks are grouped into larger units called block groups
- The block allocator tries very hard to keep each file's blocks within the same group, which reduces seek times

- Number of blocks per Block Group = 8 * block_size in bytes

- Block Group size = (Number of blocks) * block_size

- Ex: By considering the default block size 4096 bytes

  (8 * 4096) * 4096 = 128 MiB

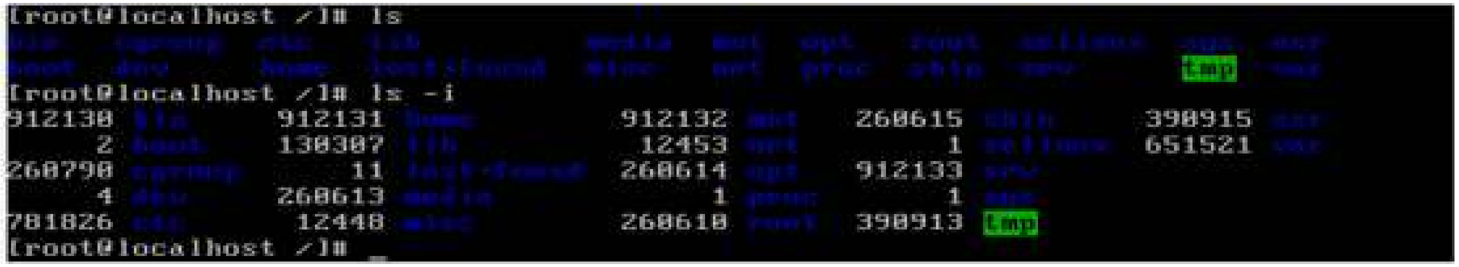
Block Groups

- **Boot Block:**
  - The boot block will maintain the boot sectors information
- **Super Block:**
  - The super block records various information about the enclosing file system, such as block counts, inode counts, supported features, maintenance information, and more
- **Group Descriptors:**
  - The standard configuration is for each block group to contain a full copy of the block group descriptor table
  - The group descriptor records the location of both bitmaps and the inode table
- **Data Block Bitmap:**
  - The data block bitmap tracks the usage of data blocks within the block group
  - One bit represents the usage status of one data block

Block Groups

- **Inode Bitmap:**
  - **inode** is a data structure used to represent a file system object
  - The inode bitmap records which entries in the inode table are in use
  - One bit represents the usage status of one inode table entry

- **Inode Table:**
  - Inode table will have a list of inodes. The inodes are placed in several tables, each of which contains the same number of inodes and is placed at a different blocks group
  - Each inode table is accessed from the group descriptor of the specific blocks group



```
[root@localhost /]# ls
bin   cgroup  etc   lib         media  mnt  opt   root  selinux  sys  usr
boot  dev     home  lost+found  misc   net  proc  sbin  srv      tmp  var
[root@localhost /]# ls -i
912130 bin      912131 home          912132 mnt    260615 sbin     390915 usr
     2 boot     130307 lib            12453 net         1 selinux  651521 var
260790 cgroup       11 lost+found    260614 opt    912133 srv
     4 dev      260613 media              1 proc        1 sys
781826 etc       12448 misc          260610 root   390913 tmp
[root@localhost /]# _
```

- **Data Blocks:**
  - The data blocks will contain the actual contents of files

Allocation of Blocks

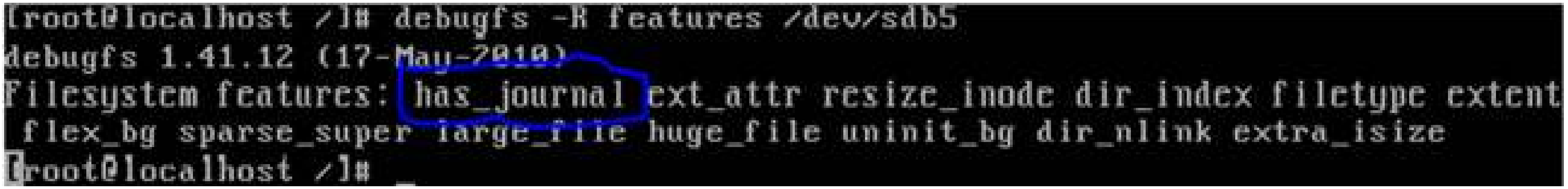
- **Ext4 first uses multi-block allocator**
  - When a file is first created, the block allocator allocates 8KiB of disk space to the file
  - When the file is closed, the unused allocations are freed, but if in case the space is fully used then the file data gets written out in a single multi-block extent
- **Ext4 uses Delayed allocation**
  - When a file needs more blocks for file writes, the file system defers deciding the exact location on the disk until all the dirty buffers are being written out to disk
  - Not committing to a particular location on disk until it's absolutely necessary is that the file system can make better location decisions
- **Ext4 tries to keep a file's data blocks in the same block group as its inode. This cuts down the I/O operations time**
- **All the inodes in a directory are placed in the same block group as the directory, when feasible**
- **The disk volume is cut up into 128MB block groups these mini-containers are used to try to maintain data locality**

Extents

- An extent is simply a set of blocks which are logically contiguous in a file system
- In Ext4 the file to logical block map has been replaced with an extent tree
- The inode must have the extents flag set for this feature to be used
- Storing the file structure as extents will result in significant compression of the file's metadata, since a single extent can replace a large number of block pointer
- The reduction in metadata will enable faster access

Journaling

- A journaling file system is a file system that maintains a special file called a journal that is used to repair any inconsistencies that occur during a system crash
- Journaling file systems store metadata or data or both based on the changes done, before writing the actual data to the hard disk



```
[root@localhost /]# debugfs -R features /dev/sdb5
debugfs 1.41.12 (17-May-2010)
Filesystem features: has_journal ext_attr resize_inode dir_index filetype extent
 flex_bg sparse_super large_file huge_file uninit_bg dir_nlink extra_isize
[root@localhost /]# _
```

Checking whether the journal is enabled

Types of Journaling

- **WriteBack**
  - Only the metadata is journaled and data is written to the file on the disk
  - File system recovery is the worst, but the performance is the best

- **Ordered**
  - This type is the reverse of writeback. The physical data is written first before the metadata is journaled
  - File system recovery is medial.

- **Journal**
  - The metadata and file contents are journaled.
  - System performance can be poorer than the other two modes, but the fault tolerance is much better.

Device Management

Linux Device Management

What is Device Files?

- Under Linux each and every hardware device treated as a file. A device file allows to accesses hardware devices so that end users do not need to get technical details about hardware.
- In short, a device file (also called as a special file) is an interface for a device driver that appears in a file system as if it were an ordinary file. This allows software to interact with the device driver using standard input/output system calls, which simplifies many tasks.

Types of Device Files:

- **Block Device :** A block device is one that is designed to operate in terms of the block. A block device has an associated block device driver that performs I/O by using file system block-sized buffers from a buffer cache supplied by the kernel

- **Character Device:** A character device is any device that can have streams of characters read from or written to it. A character device has a character device driver associated with it that can be used for a device such as a line printer that handles one character at a time.

Linux Device Management

In Linux all devices were considered as files

File	Device
/dev/fd0	Floppy disk
/dev/hda0	IDE Hard drive 1, partition 0
/dev/hdb3	IDE Hard drive 2, partition 3
/dev/sda	First SCSI hard drive
/dev/cdrom	CD ROM drive This device may be on the secondary controller as a master (/dev/hdc) or slave (/dev/hdd).
/dev/mouse	May be a pointer to /dev/psaux which is the ps2 device or /dev/cua which is a serial device or /dev/ttyS0
Disk Drives	
/dev/hda	primary IDE master
/dev/hdb	primary IDE slave

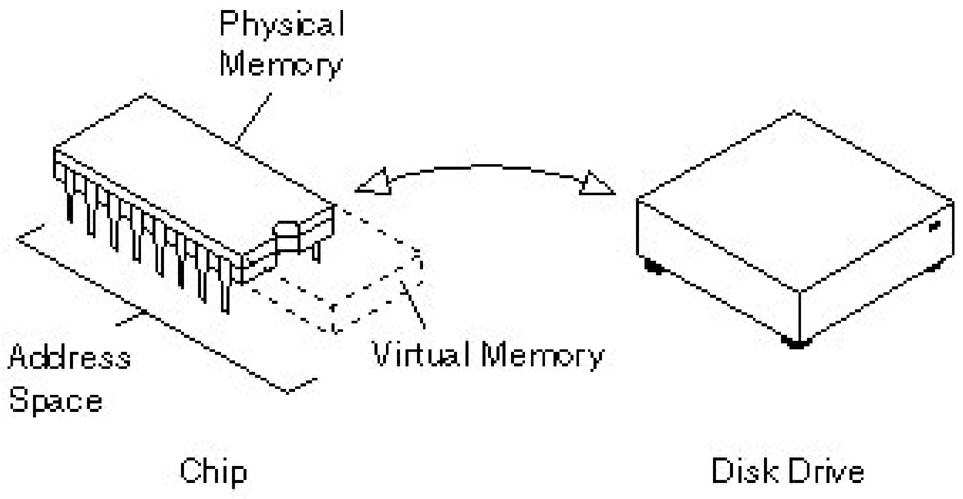
Linux Device Management

- **IRQ (Interrupt Service Request):**
  - IRQs are hardware lines over which devices can send interrupt signals to the microprocessor.
  - It is important to assign different IRQ addresses to different hardware devices is because the interrupt request signals run along single IRQ lines to a controller. This interrupt controller assigns priorities to incoming IRQs and sends them to the CPU.
  - The interrupt controller can control only one device per IRQ line, so if we assign the same IRQ address to multiple devices, we are likely to get an IRQ conflict

Memory Management

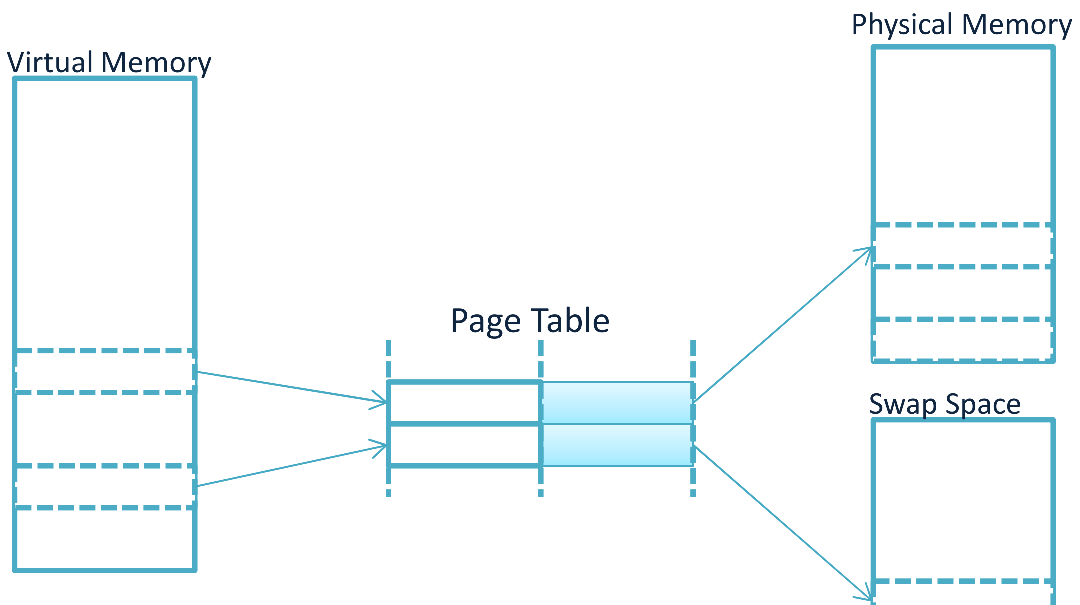
Memory Management

- **Physical memory refers to the RAM installed in the system**
- **Virtual Memory extends the available memory of the computer by storing the inactive parts of the content RAM on a disk**

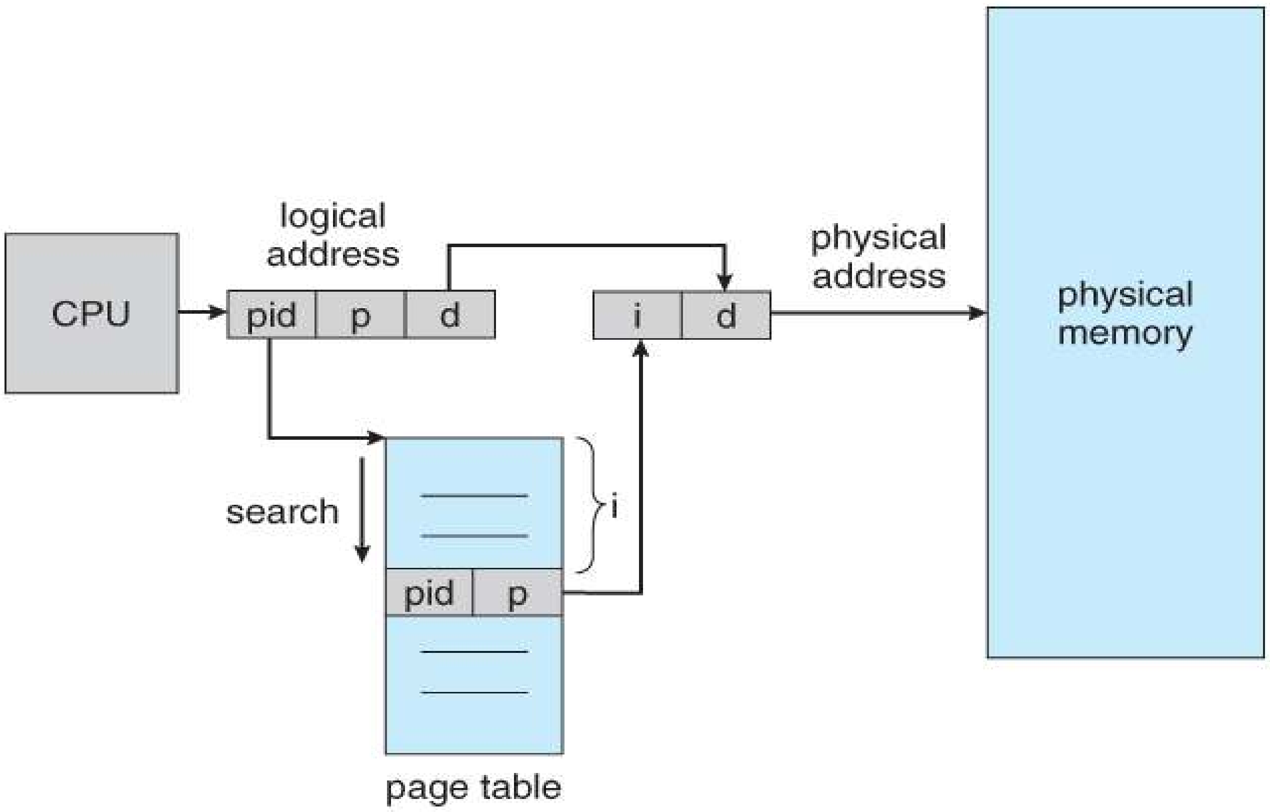


Memory Management

- **Page : Memory is divided into chunks of equal size called pages**
- **Page Table: Stores the mapping between virtual addresses and physical addresses**

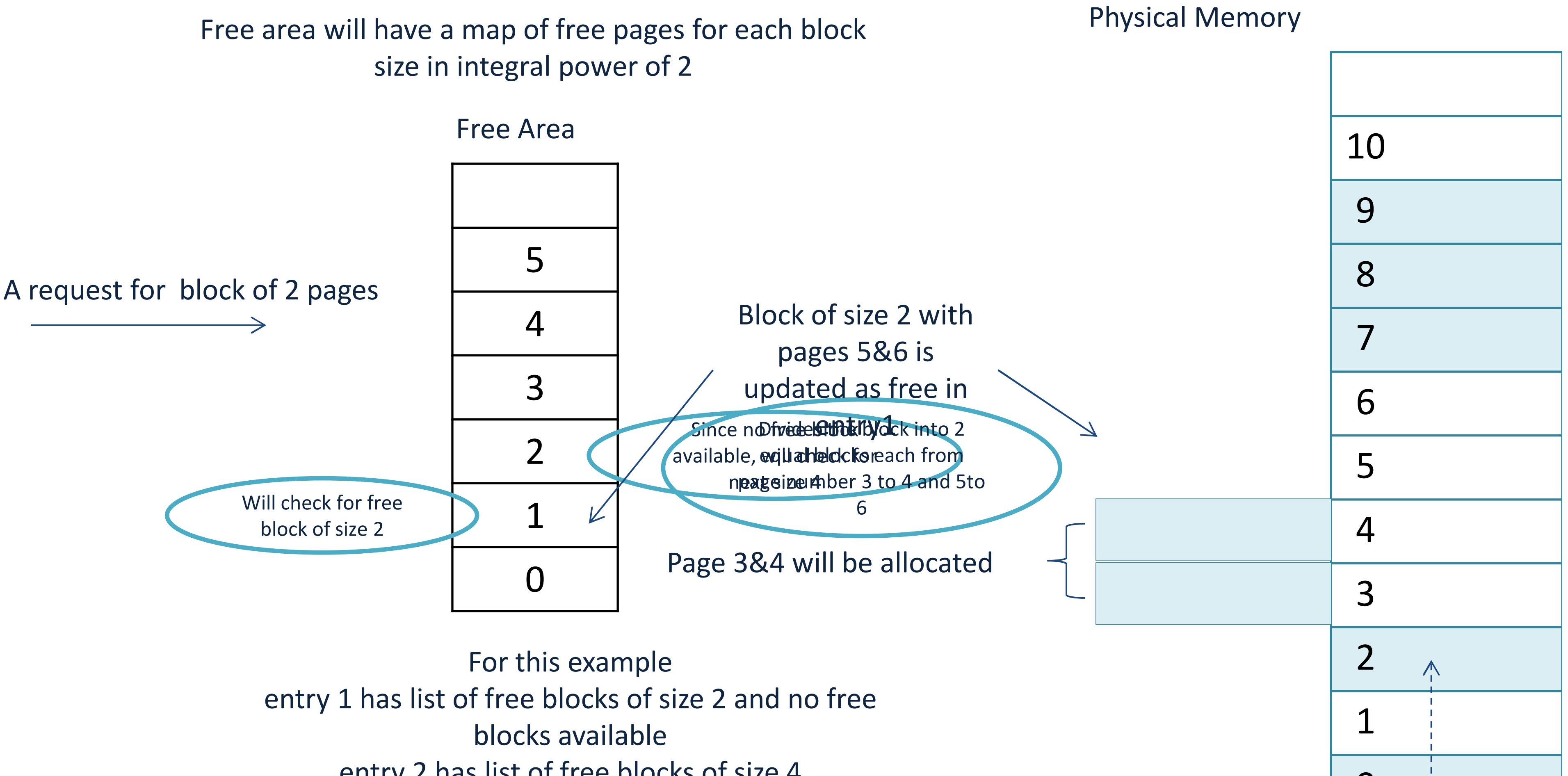


Mapping between physical and virtual address



Page Allocation

Free area will have a map of free pages for each block size in integral power of 2



A request for block of 2 pages

Will check for free block of size 2

Since no free block available, will check for next size 4

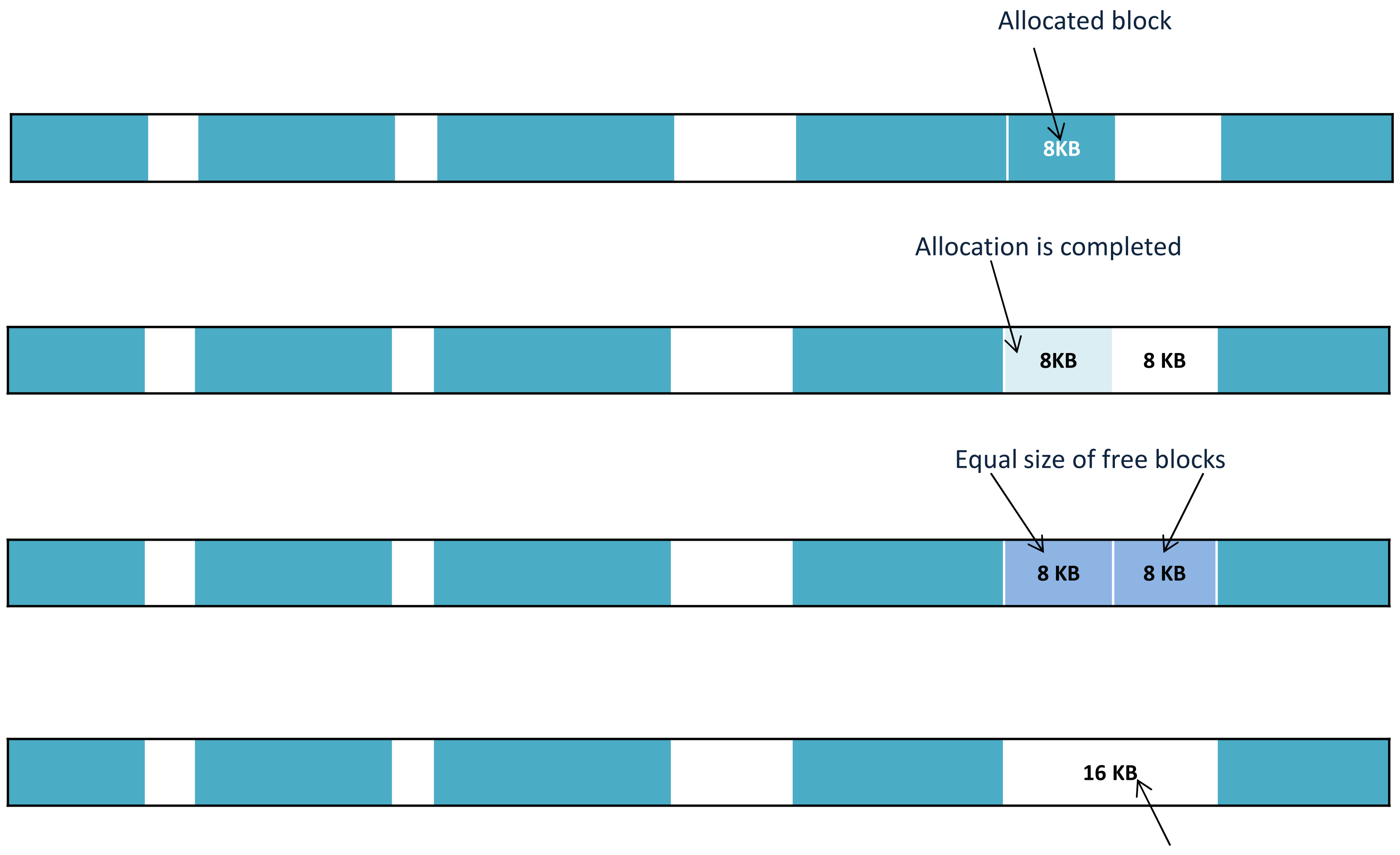
Divide the block into 2 equal blocks each from page number 3 to 4 and 5to 6

Block of size 2 with pages 5&6 is updated as free in entry1

Page 3&4 will be allocated

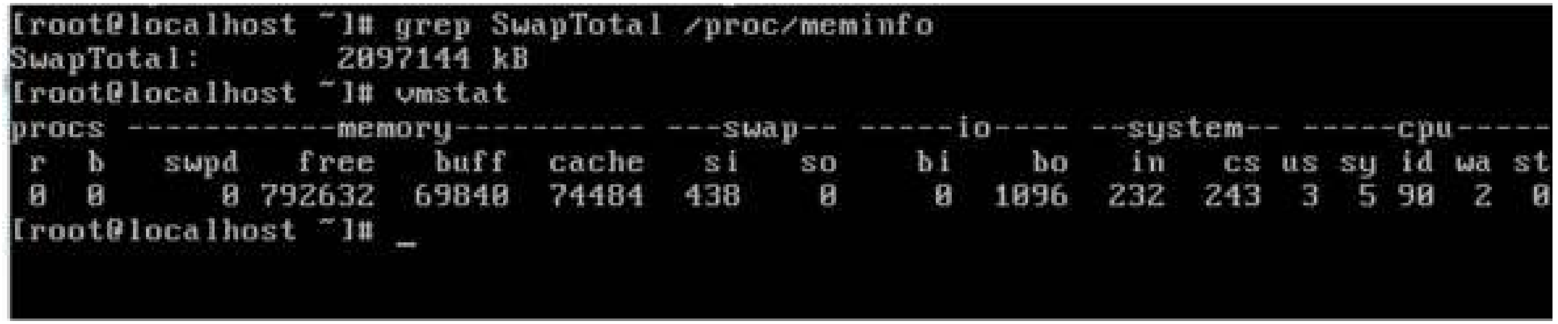
For this example
entry 1 has list of free blocks of size 2 and no free blocks available
entry 2 has list of free blocks of size 4

Page De-allocation



Swapping

- Interchanging the pages between physical memory and disk is called swapping
- When a process requires more memory than available then swapping occurs by moving the in-active pages into disk
- When a Page fault occurs then the required page will be swap-in from disk to physical memory



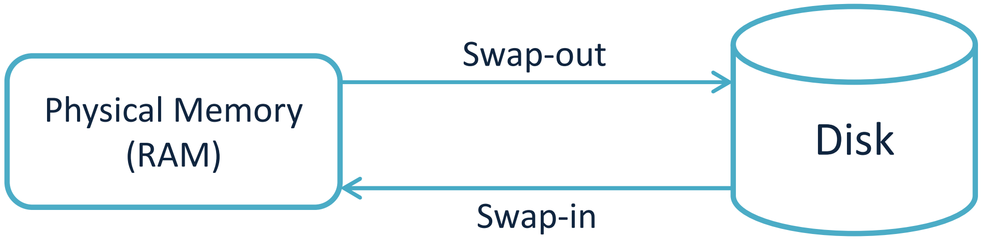
```
[root@localhost ~]# grep SwapTotal /proc/meminfo
SwapTotal:        2097144 kB
[root@localhost ~]# vmstat
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0      0 792632  69840  74484  438    0     0  1096  232  243  3  5 90  2  0
[root@localhost ~]# _
```

Check Swap Usage

Swapping

- **Swap-Out:**
  - The process of writing pages out from physical memory to disk to free memory is called swap-out

- **Swap-In:**
  - The process of loading pages in from disk to physical memory is called swap-in



Advantages of Virtual Memory

- More memory than physical RAM
- Multi Programming Environment
- More active processes
- Allocation of memory is cost-effective

Thank You

Linux Essentials

Lesson 1 Basic of RHEL & Installation

Module Overview

1.1 History

1.2 Versions

1.3 Pre-Req for Installation of RHEL 5.5

1.4 Installation of RHEL 5.5 on VM Machine

1.1. History

- **1984: The GNU Project and the Free Software Foundation.**
  - Creates open source version of UNIX utilities.
  - Creates the General Public License (GPL).
- **Software license enforcing open source principles**
  - 1991: Linus Torvalds Creates open source, UNIX-like kernel, released under the GPL Ports some GNU utilities, solicits assistance online.
  - Today: Linux kernel + GNU utilities = complete, open source, UNIX-like operating system.
- **Packaged for targeted audiences as distributions.**

1.2 Versions

- **Versions**
  - RHEL 2.1
  - RHEL 3
  - RHEL 4
  - RHEL 5
  - RHEL 6
  - RHEL 7
- **Editions**
  - Server
  - Workstation

1.3 Pre-Req for Installation of RHEL 5.5

- **CPU**
  - 2GHz or higher
- **Memory/RAM**
  - 1 GB minimum, upto the system limit
- **Hard Disk**
  - 4 GB minimum

1.4 Installation of RHEL 5.5 on VM Machine

1. Boot from DVD
2. Select Install or upgrade existing system options.
3. Anaconda is the Linux installer performs the installation.
4. Select language as English
5. Select Keyboard type as US English
6. Optionally test media if required
7. Select the storage device
8. Enter the hostname
9. Select the timezone
10. Enter the root password
11. Select the partition option
12. Select the type of installation(default:basic server)
13. Select additional software to be installed
14. Start the installation.
15. Once the installation is completed, perform the post installation tasks

Summary

- **In this Module you have learnt:**
  - What is the history of Linux?
  - What are the different version of Linux?
  - What are the Pre-Req for Installing of RHEL 5.5
  - Installation of RHEL 5.5 on VM Machine

Lab Exercise

- **INSTALLATION OF Linux**

Review Questions

1. **Who created Linux?**
2. **What is the latest version of RHEL?**

Linux Essentials

Lesson 1 Basic of RHEL & Installation

Module Overview

1.1. History

- **1984: The GNU Project and the Free Software Foundation.**
  - Creates open source version of UNIX utilities.
  - Creates the General Public License (GPL).
- **Software license enforcing open source principles**
  - 1991: Linus Torvalds Creates open source, UNIX-like kernel, released under the GPL Ports some GNU utilities, solicits assistance online.
  - Today: Linux kernel + GNU utilities = complete, open source, UNIX-like operating system.
- **Packaged for targeted audiences as distributions.**

1.2 Versions

- **Versions**
  - RHEL 2.1
  - RHEL 3
  - RHEL 4
  - RHEL 5
  - RHEL 6
  - RHEL 7

- **Editions**
  - Server
  - Workstation

1.3 Pre-Req for Installation of RHEL 5.5

- **CPU**
  - 2GHz or higher
- **Memory/RAM**
  - 1 GB minimum, upto the system limit
- **Hard Disk**
  - 4 GB minimum

1.4 Installation of RHEL 5.5 on VM Machine

1. Boot from DVD
2. Select Install or upgrade existing system options.
3. Anaconda is the Linux installer performs the installation.
4. Select language as English
5. Select Keyboard type as US English
6. Optionally test media if required
7. Select the storage device
8. Enter the hostname
9. Select the timezone
10. Enter the root password
11. Select the partition option
12. Select the type of installation(default:basic server)
13. Select additional software to be installed
14. Start the installation.
15. Once the installation is completed, perform the post installation tasks

Summary

- **In this Module you have learnt:**
  - What is the history of Linux?
  - What are the different version of Linux?
  - What are the Pre-Req for Installing of RHEL 5.5
  - Installation of RHEL 5.5 on VM Machine

Lab Exercise

- **INSTALLATION OF Linux**

Review Questions

1. **Who created Linux?**
2. **What is the latest version of RHEL?**

Linux Essentials

Lesson 2 System Initialization

Module Overview

2.1 BIOS Initialization

2.2 Boot Loader

2.3 Kernel initialization

2.1 BIOS Initialization

BIOS executes MBR

MBR executes GRUB

GRUB executes Kernel

Kernel executes /sbin/init

Init executes runlevel programs

Runlevel programs are executed from /etc/rc.d/rc*.d

2.2 Boot Loader

- **Bootloader loads the OS**

- **2 Major Bootloaders available for Linux**
  - GRUB
  - LILO

GRUB

- **GRUB- the Grand Unified Bootloader**
  - Command-line interface available at boot prompt.
  - Mostly Boots from ext2/ext3
  - Can be password protected
- **Grub configuration file is /boot/grub/grub.conf.**
- **Changes to grub.conf take effect immediately.**

2.3 Kernel initialization

- **Kernel boot time functions**
  - Device detection
  - Device driver initialization
  - Mounts root file system read only
  - Loads initial process (init)

- **Kernel Parameters are set in /etc/sysctl.conf , used to pass additional information to the kernel**

- **Messages from Kernel is stored in Kernel Ring Buffer, which can be seen by using dmesg command**

init initialization

- **init reads its config: /etc/config**
  - Initial run level
  - System initialization scripts
  - Run level specific scripts directories
  - Initialize X in run level 5

/etc/rc.d/rc.sysinit

- **Important tasks include:**
  - Activate udev & selinux
  - Sets kernel parameters in /etc/sysctl.conf
  - Sets the system clock
  - Loads keymaps
  - Enables swap partitions
  - Sets hostname
  - Root filesystem check & remount
  - Activate RAID & LVM devices
  - Enable disk quotas
  - Check & mount other file systems
  - Cleans up stale locks and PID files

Run Levels

- **init defines run levels 0-6, S.**
- **The run level is selected by either,**
  - The default in /etc/inittab at boot.
  - Passing an argument from the boot loader.
  - Running init x after boot (where x is the desired run level).
- **Show current & previous run levels**
  - /sbin/runlevel

/etc/rc.d/rc

- **Initializes the default run level as per the /etc/inittab initdefault line such as id:3:initdefault:**
  - /etc/rc.d/rc 0
  - /etc/rc.d/rc 1
  - /etc/rc.d/rc 2
  - /etc/rc.d/rc 3 ← (Run level 3)
  - /etc/rc.d/rc 4

Summary

- **In this Module you have learnt:**
  - BIOS Initialization
  - Boot Loader
  - Kernel initialization

Lab Exercise

- **Changing the runlevel**
- **Viewing the runlevel directories, default runlevel**
- **changing the default runlevel**

Review Questions

1. **What is a runlevel?**
2. **What is the default runlevel of your linux server?**
3. **What is initrd?**

Linux Essentials

Lesson 3 Device Management

Module Overview

3.1 Device Nodes

3.2 Char Device

3.3 Block Device

3.1 Device Nodes

- **In order for the operating system to recognize the hardware device, the device must have a software name, usually referred to as a device special file or device node.**
- **Device nodes correspond to resources that an operating system's kernel has already allocated.**
- **computer system accesses device nodes using standard system calls and treats them like regular computer files**

- **All devices are managed by udev**

3.2 Char Device

- **Character special files or character devices provide unbuffered, direct access to the hardware device.**
- **They do not necessarily allow programs to read or write single characters at a time; that is up to the device in question.**
- **Character devices are sometimes known as raw devices**

E,g) Keyboard

3.3 Block Device

- **Block special files or block devices provide buffered access to hardware devices, and provide some abstraction from their specifics.**

e.g) hard disk

Summary

- **In this Module you have learnt:**
  - Device Nodes
  - Char Device
  - Block Device

Lab Exercise

- **View the list of devices detected**

Review Questions

1. **How is harddisk accessed**
2. **How a 3rd IDE hdd will be detected in Linux**

Linux Essentials

Lesson 4 Process Management

Module Overview

4.1 Process and Job Control

ps

- ps command is used to display the process attributes of all active processes.
- Syntax:
  - ps [option [arguments] …]
- Options:
  - -f- full form
  - -e - system processes

nice-renice

Command /Utility	Description
nice	Changes the priority of a new process
renice	Changes the priority of a existing running process

- Higher the nice value , lower the priority
- Nice values may be altered.
  - When starting a process:

    $ nice -n 5 *command*
  - After starting:

    $ renice 5 *PID*
- Only root may decrease nice value

Top

- Interactive Process Management Tool
- Capabilities
  - Display real-time process information.
  - Allow sorting, killing and re-nicing.

nohup

- When using the command shell, prefixing a command with nohup prevents the command from being aborted if you log out or exit the shell.

- e.g) nohup gcalctool

kill - killall

- kill used to kill individual process
- killall is used to kill process all process that matches by name

Number	Signal	Description
9	SIGKILL	Forces the process to terminate unconditionally. This is the "sure kill" signal.
15	SIGTERM	Termination signal. Shuts down the process but gives the process a chance to terminate properly by cleaning up.

Example :

```
kill -9 1814
kill -15 1814
killall -9 gcalcctool
```

bg – fg - jobs

- **bg - bg is a job control command that resumes suspended jobs while keeping them running in the background.**

  e.g) bg %1

- **fg - bg is a job control command that resumes suspended jobs while keeping them running in the background.**

  e.g) fg command continues a stopped job by running it in the foreground

- **jobs - List the status of all running jobs.**
  - e.g) jobs

Summary

- **In this lesson you have learnt**
  - Process command and its options
  - Modifying the priority of process
  - Kill command
  - Foreground and background process

Lab Exercise

- **List processes**
- **Change process priority**
- **Kill foreground and background process**

Review Questions

- Question 1: ____ is appended to the command to send it at the background.

- Question 2: ____ command is used to know which all command are running at the background.

Linux Essentials

Lesson 5 System Services

Module Overview

5.1. NTP - Network Time Protocol

- Network Time Protocol daemon.
- Network Time Protocol is the most common method to synchronize the software clock of a GNU/Linux system with internet time servers.
- It is designed to mitigate the effects of variable network latency and can usually maintain time to within tens of milliseconds over the public Internet.

5.2. System Logging

- Linux system administrators often need to look at log files for troubleshooting purposes. In fact, this is the first thing any sysadmin would do.
- The default location for log files in Linux is /var/log

5.3. Log Server – Centralized Log Management

- Monitoring individual servers is difficult.
- We can designate one server as a centralized log server and make the clients to record their logs to the central server using rsyslog

5.4. VNC Service

- We can use VNC to take the control of a remote desktop

e.g) tigervnc

5.3. SSH Service

- sshd
- used for secure connectivity to server
- It uses port 22

5.3. Log Server – Centralized Log Management

- **We can configure all the Clients/Server Linux system to record the logs to a centralized Server, with the help of rsyslog**

5.4. SCP Service

- scp allows files to be copied to, from, or between different hosts.
- It uses ssh for data transfer and provides the same authentication and same level of security as ssh

5.5. CUPS Service

- CUPS (an acronym for Common Unix Printing System)
- Is a modular printing system for Unix-like computer operating systems which allows a computer to act as a print server.
- A computer running CUPS is a host that can accept print jobs from client computers, process them, and send them to the appropriate printer.
- One can access the browser interface by http://localhost:**631**

5.6. xinetd Service – Telnet

- Xinetd
  - It is a super daemon, that facilitates other daemons like telnet
  - starts programs that provide Internet services.
  - Instead of having such servers started at system initialization time, and be dormant until a connection request arrives, xinetd is the only daemon process started and it listens on all service ports for the services listed in its configuration file. When a request comes in, xinetd starts the appropriate server.
  - Because of the way it operates, xinetd (as well as inetd) is also referred to as a super-server.
- Telnet
  - Is a program that allows users to log into your server and get a command prompt
  - One of the disadvantages of Telnet is that the data is sent as clear text. This means that it is possible for someone to use a network analyzer to peek into your data packets and see your username and password. A more secure method for remote logins would be via Secure Shell (SSH) which uses varying degrees of encryption.

5.7. Service Management

- chkconfig can be used to activate/deactivate/list the services in your server
  - chkconfig httpd on
- service command an be used to start/stiop/status services in your system
  - service httpd start

5.8. Scheduling Service – crontab, anacron, at

- A system daemon which performs a specific task at regular intervals.

where the file contains the commands to execute

MIN	HOUR	DOM	MOY	DOW	COMMAND
(0-59)	(0-23)	(1-31)	(1-12)	(0-6)	---
$ 0	18	*	*	*	/home/gather

- Anacron was similar to cron, but it is duration/interval based instead of time based. It fires after a delay, recurring every some interval.

- At is another utility that we can use to schedule jobs that doesn’t recur(one time execution only). List of all jobs waiting to be executed can be seen by using atq command.

Summary

- **In this lesson you have learnt**
  - What is NTP?
  - How to see System Logging?
  - How to centralize logging?
  - What is VNC?
  - What is SSH Service?
  - What is SCP?
  - What is CUPS?
  - What is xinetd. How it helps other services?
  - How to manager Services?
  - How to schedule tasks using crontab?

Linux Essentials

Lesson 7 Bash Editing

Module Overview

7.1. **Bash profile & its components**

7.2. **Set Environment Variables**

7.3. **Create aliases**

7.1. Bash profile & its components

- The default shell is the Bash shell.
- Bash is a command language interpreter that executes commands read from any input(file/keyboard).
- The Bash Shell starts by reading the /etc/profile file , which usually contains the system variables, user environment and aliases.
- The login process continues with the files
  - .bash_profile
  - .bash_login
  - .profile
- If you want to modify the template of the profile files from which the profile files of new users will be created, you can modify /etc/skel/ files

User Environment initialization

```
execute /etc/profile

IF ~/.bash_profile exists THEN
    execute ~/.bash_profile
ELSE
    IF ~/.bash_login exist THEN
        execute ~/.bash_login
    ELSE
        IF ~/.profile exist THEN
            execute ~/.profile
        END IF
    END IF
END IF
```

7.2. Set Environment Variables

- Variables which are available in the users total environment are called as environment variables.
- Few common environment variables are,
  - HOME
    - describes the path to user's home directory.
  - PATH
    - specifies the path, in which, invoked commands needs to be searched.
  - SHELL
    - Sets the default shell that will be used by Tools.

export <envvariablename>=value

For persistent environment variables modify the appropriate profile files(system wide or user specific)

Commonly used Environment Variables

- PS1 – Can be used to customize your shell prompt.
- PATH – Display lists directories the shell searches, for the commands.
- EDITOR – The user's preferred text editor.(for sudo, git, subversion, …)
- HISTFILESIZE -The maximum number of lines contained in the history file
- HOME – User's home directory to store files.
- EUID - Effective User ID

7.3. Create aliases

- The alias command allows you to make new shortcuts and synonyms for commonly used comands. The basic usage is:

```
alias newcommand='yourcommand -arguments‘
```

```
alias ls='ll'
```

Summary

- **In this lesson you have learnt**
  - What is bash profile?
  - How to set environment variable?
  - How to create alias?

Linux Essentials

Lesson 6 Kernel Management

Module Overview

6.1 Kernel Images and Modules

6.2 Building Custom Kernel using CLI and GUI/Kernel Patching

6.1 Kernel Images and Modules

- Kernel Image
  - vmlinuz is the kernel image that will be used by the linux.
- Kernel Module
  - Kernel modules are pieces of code that can be loaded and unloaded into the kernel upon demand. This is to keep the kernel image as small as possible by not loading unnecessary modules before hand.

Working with Kernel Modules

- List Currently Loaded Modules => lsmod | less
- List Available Kernel Modules => modprobe -l | less
- To show information about a module => modinfo <modulename>
- To add a new module to kernel => insmod <modulename>
- Remove a Module from the Kernel => rmmod ts_fsm

6.2 Building Custom Kernel using CLI and GUI/Kernel Patching

- You can build custom kernel as per your requirements with the customized kernel sources
- You can patch the kernel with fixes/enhancements depending on your requirements.

Summary

- **In this lesson you have learnt**
  - What is Kernel Module
  - How to see the loaded/available Module

Lab Exercise

- Find the list of all loaded modules
- Find the list of all available modules
- Load a Module
- Unload/Remove a Module

Review Questions

- Question 1: What is a Kernel Module?
- Question 2: How can I load a Module into Kernel?

Linux Essentials

Lesson 8 File system and Disk Management

Module Overview

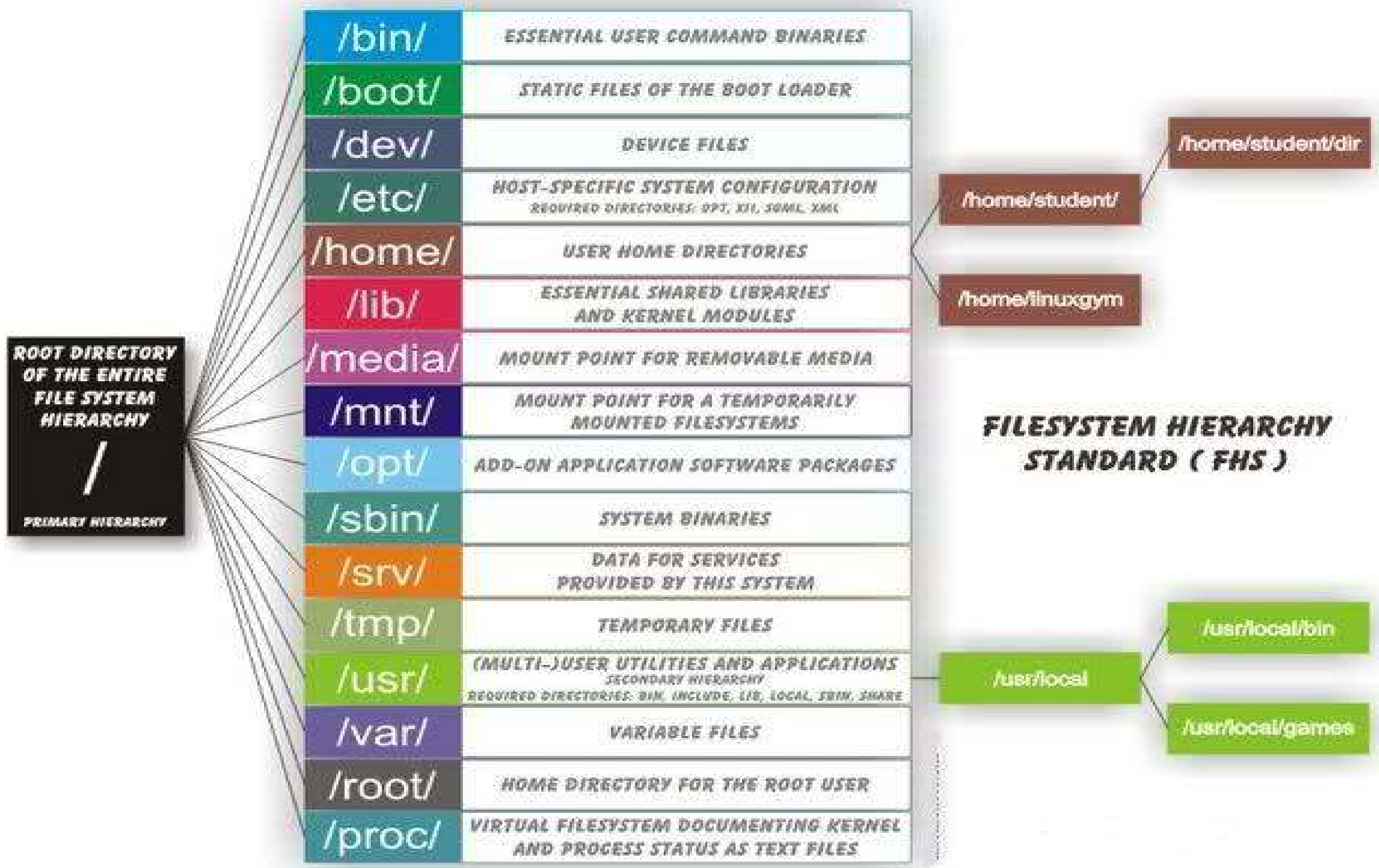
8.1. File system Management

8.2. Disk Management

8.1. File system Management

- Linux supports numerous file systems, but common choices for the system disk on a block device include the ext* family (ext2, ext3 and ext4), XFS, JFS, ReiserFS and btrfs.
- You need to format a partition with appropriate file system before storing data.
- mkfs is used to build a Linux file system on a device, usually a hard disk partition.
- df
- du

Linux Directory Structure(Recap)



/bin/	ESSENTIAL USER COMMAND BINARIES
/boot/	STATIC FILES OF THE BOOT LOADER
/dev/	DEVICE FILES
/etc/	HOST-SPECIFIC SYSTEM CONFIGURATION REQUIRED DIRECTORIES: OPT, X11, SGML, XML
/home/	USER HOME DIRECTORIES
/lib/	ESSENTIAL SHARED LIBRARIES AND KERNEL MODULES
/media/	MOUNT POINT FOR REMOVABLE MEDIA
/mnt/	MOUNT POINT FOR A TEMPORARILY MOUNTED FILESYSTEMS
/opt/	ADD-ON APPLICATION SOFTWARE PACKAGES
/sbin/	SYSTEM BINARIES
/srv/	DATA FOR SERVICES PROVIDED BY THIS SYSTEM
/tmp/	TEMPORARY FILES
/usr/	(MULTI-)USER UTILITIES AND APPLICATIONS SECONDARY HIERARCHY REQUIRED DIRECTORIES: BIN, INCLUDE, LIB, LOCAL, SBIN, SHARE
/var/	VARIABLE FILES
/root/	HOME DIRECTORY FOR THE ROOT USER
/proc/	VIRTUAL FILESYSTEM DOCUMENTING KERNEL AND PROCESS STATUS AS TEXT FILES

8.2. Disk Management

- Disks can be managed using GUI using Disk Utility tool
- fdisk can be used to create/delete/list partitions
- configuration file /etc/fstab contains the necessary information to automate the process of mounting partitions, for persistently mounting partitions, lvm and nfs shares.

Summary

- **In this lesson you have learnt**
  - What are the different file systems used in linux?
  - How to create a file system on a partition?
  - How to partition a hard disk?

Linux Essentials

Lesson 9 File Management

Module Overview

9.1. File Operation

Creating a File

- touch file.txt
- cat > file.txt
- vi file.txt

Copying a File

- cp example1.txt barney.txt

Renaming/Moving a File

- mv foo2.txt backups/foo3.txt

Deleting a File

- rm bar.txt

Directory Manipulation

Creating a Directory

- mkdir amal
- mkdir -p linuxtutorialwork/foo/bar
- mkdir -pv linuxtutorialwork/foo/bar

Copying a Directory

- cp example1 barney

Renaming/Moving a Directory

- mv barney backups

Deleting an Empty Directory

- rmdir linuxtutorialwork/foo/bar

Deleting a Non-Empty Directory

- rm -r backups

9.2. Disk Utilities

- df
  - df finds the disk free space or disk usage.
  - Ex: $df
  - Outputs a table consisting of six columns. Column names explains each column. Columns, size, used and avail use kilobyte as unit.
- du
  - du command displays the list of directories that exist in the current directory along with their sizes.
  - The last line of the output gives the total size of the current directory including its subdirectories.
  - Note that by default the sizes given are in kilobytes.

9.3. Links

- HardLinks
  - Hard links cannot link directories.
  - Cannot cross file system boundaries.

```
ln /full/path/of/original/file /full/path/of/hard/link/file
```

- SoftLinks
  - To create links between directories.
  - Can cross file system boundaries.
  - Removing the original file of a softlink will break the link

```
ln -s /full/path/of/original/file /full/path/of/soft/link/file
```

9.4 Compression / Decompression of files

- Red Hat Enterprise Linux provides the bzip2, gzip, and zip tools for compression from a shell prompt.
- The bzip2 compression tool is recommended because it provides the most compression and is found on most UNIX-like operating systems.
- The gzip compression tool can also be found on most UNIX-like operating systems.
- To transfer files between Linux and other operating system such as MS Windows, use zip because it is more compatible with the compression utilities available for Windows..
- bzip2 performs better compression(in terms of compressed size), but takes more time.

Compression Tool	File Extension	Decompression Tool
bzip2	.bz2	bunzip2
gzip	.gz	gunzip
zip	.zip	unzip

Summary

- **In this lesson you have learnt**
  - How to perform different file operations?
  - How to use different disk utilities?
  - What is the difference between hard and soft links?
  - How to compress/decompress in linux?

Linux Essentials

Lesson 10 Package Management

Module Overview

10.1. Package management

- RPM started as packaging format for Red Hat Linux.
- It is a program for installing, uninstalling and managing software packages in Linux.
- Advantages:
  - Straight forward program installation/uninstallation.
  - Ease of updating programs.
  - Availability of versions.
  - Software information stored in a local database.
- Packages are provided by Red Hat Network.
  - Centralized management of multiple systems.
  - Easy retrieval of errata packages.
  - Systems must be registered first.
  - Custom package repositories may also be used.

10.2. RPM

- Package installation is never interactive.
- Applies to all software(Core OS & Add-ons).

- Primary RPM options:

  Install : rpm – i

  Erase: rpm –e

  Verbose: -v

  Query all installed packages : -qa

  Query a installed package : -qp <rpmname>

  Query a package before installation(to know details of a package): -qip <rpmname>.rpm

10.3. YUM - Yellow Update Modifier

- Yum allows automatic updates, package and dependency management.

- Configuration in /etc/yum.conf and / etc/yum.repos.d/
- Used to install, remove and list software
  - yum *install packagename*
  - yum *remove packagename*
  - yum *update packagename*
  - yum *list available*
  - yum *list installed*
- Graphical Package Management
  - List and install software updates
  - View, install and un-install other packages

Summary

- **In this lesson you have learnt**
  - What is rpm format?
  - How to install/remove packages using rpm?
  - How to install/remove packages using yum?

Linux Essentials

Lesson 12 File Permissions

Module Overview

12.1	File permission
12.2	ACL - Access Control List for Files & Directory

12.1 File permission

- File/Directory Permissions can be modified with chmod by using 2 methods:
  - Symbolic
  - Numberic
- Default permissions of new file/directory is controlled by umask of the user who creates the file/directory

Changing File Permissions - chmod

- chmod command is used to change the file permissions.
- Syntax: chmod <category> <operation> <permission> <filenames>

Changing Permissions – Symbolic Method

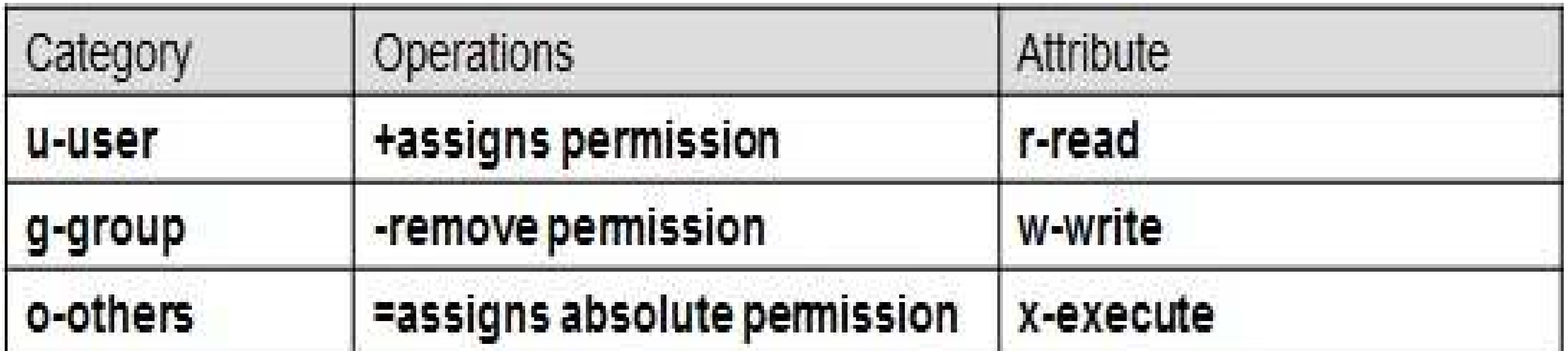
$chmod u+x note

- $ ls - l note
  - -rwx r-- r --1 …… note

$ chmod ugo+x note

- $ ls - l note
- -rwxr-xr-x …… note

- When we use + symbol, the previous permissions will be retained and new permissions will be added.
- When we use = symbol, previous permissions will be overwritten.



Category	Operations	Attribute
u-user	+assigns permission	r-read
g-group	-remove permission	w-write
o-others	=assigns absolute permission	x-execute

Changing Permissions – Numeric Method

- **Octal notation:**
  - It describes both category and permission.
  - It is similar to = operator (absolute assignment).
    - read permission: assigned value is 4
    - write permission: assigned value is 2
    - execute permission: assigned value is 1
  - Example 1:

```
$ chmod   666            note
```

- It will assign read and write permission to all.

Special Permissions

- **suid**
  - When set on a file, the file will execute with permissions of the owner of the command, and not as executor (default)of the command
- **sgid**
  - When set on a file, runs with group affiliation of the group of the command
  - When set on a directory all files/directories within it will have the same group membership
- **sticky bit**
  - files in directories with the sticky bit set , can only be removed by the owner and root, regardless of the write permissions of users on that directory

12.2 ACL - Access Control List for Files & Directory

- ACL provides an additional flexible permission mechanism for file system on a Linux system.

- It enhance the traditional UNIX file permissions for files & folder. With ACL, you can give permissions for any user or any group with fine-grained access rights.

- Set Permissions
  - setfacl -R -m u:rajesh:rwx /test/demoacl.txt
- Get(list ACL permissions) Permissions
  - getfacl /test/demoacl.txt

Summary

- **In this lesson you have learnt**
  - How to configure file/directory permissions?
  - How to configure fin-grained permissions using ACL?

Linux Essentials

Lesson 13 Network Services

Module Overview

13.1	Network configuration
13.2	NW Devices
13.3	Virtual NW Adapters
13.4	NW Diagnostics - ping , traceroute, host

13.1 Network configuration

- Network configuration can be modified using:
  - Using GUI
  - CLI using ifconfig
  - Using system-config-network
  - Modifying configuration files

13.2 NW Devices

- Network interfaces in Linux are enumerated as eth[0123…], but these names do not necessarily correspond to actual labels on the chassis.

e.g) eth0, eth1, eth2

13.3 Virtual NW Adapters

Multiple number of virtual network adapters can be added in a Virtual machine.

Virtual network adapters can be configured to directly communicate with the physical network or as private or as NAT or host only connection.

13.4 NW Diagnostics - ping , traceroute, host

- **ping**
  - Can be used to test the network connectivity to a remote host
- **traceroute**
  - Can be used to find the list of all intermediate hops a packet is travelling to reach the destination
- **host/dig command**
  - Can be used to test DNS name resolution

Summary

- **In this lesson you have learnt**
  - How to configure network using commandline?
  - How to add/remove/configure Virtual Network Adapters?
  - What are the network diagnostic tools and how to use them?

Linux Essentials

Lesson 14 Network Services

Module Overview

14.1 Backuping

- **tar**
  - Tape Archive. Combines multiple files/directories into a single tar file.
- **dd**
  - used for copying and converting data. It can also be used for backup/restore
- **dump**
  - more powerful tool for performing backup and recovery. Can perform incremental backup.
- **rsync**
  - Can be used for performing copying and synchronizing files across systems. Can be used in performing incremental backup.

14.2 Restoring using Backup

- restore command can be used to restore data from dump backups.
- It supports interactive and non-interactive restores.

Summary

- **In this lesson you have learnt**
  - How to backup?
  - How to restore?

Linux Essentials

Lesson 15 Troubleshooting

Module Overview

15.1 Root Password Recovery

- Root password might get forgotten or no more available
- In these type of cases, one can delete/reset the root password

1.Login as single user mode

2.Delete the root password as below:

```
passwd –d root
```

15.2 Linux Rescue Mode

- This is used to recover from some failures.
- One can boot from DVD and enter rescue mode, to access the files stored on your system's hard drive, even if you cannot actually run Red Hat Enterprise Linux from that hard drive.
- Rescue environment will find your Linux installation and mount it under the directory /mnt/sysimage
- You can mount your original Linux installation root from /mnt/sysimage to / by using chroot command

15.3 File System Recovery

- We can check the consistency of a file system by using fsck command
- It can be used to repair found errors.

Summary

- **In this lesson you have learnt**
  - How to recover from the loss of the root password?
  - How to use rescue mode to recover from failures?
  - How to check the consistency of filesystem?

Linux Essentials

Lesson 16 Network Installation

Module Overview

16.1 Introduction to Network Installation

16.1 Introduction to Network Installation

- We can install Linux over the network
- Additionally if you want to automate(without manually providing all options during installation) the installation, we can use a kickstart file(Similar to answer file in windows)
- We can use syslinux as the bootloader for implementing Network Installation

Summary

- **In this lesson you have learnt**
  - What is network installation?
  - How to perform network installation?