**Chapter 13**

**Administration File System and General System Administration**

**Device Files-**

The definition of a file has been broadened by unix by declaring all the physical devices as a file. It includes tapes, printer, floppy drive, hard disks and terminals.

Any output directed to it will be reflected onto the respective physical associative with the filename. Whenever you are issuing a command to print a file, actually you are directing the file’s output to the file associated with the printer. The kernel takes care of this mapping, these special files to respective devices.

$ls –l /dev

This listing will reveal two vital points:

1. Device files can be grouped into mainly two categories block and character devices depending on the first character of the permission field (b or c).
2. The fifth field–normally representing the size for other files–consist of a pair of numbers (major and minor numbers). A device file contains no data.

**Block and Character devices:**

Floppy drive, cdrom, hard disk comes into this section because to whenever a user performs a write operation firstly it is written into chunks or blocks. And when it performs the read operation, the buffer cache accessed the recently used data. So these devices are using buffer and writing there data into blocks for such devices prefix ‘b’ is used.

The terminal, tape drive and printer are character special devices because the read/write operation ignores the buffer cache and access the device directly.

**Major and Minor No.:**

The set of routines is requires to operate specific devices is known as device drivers. When a device is called the kernel passes some parameters for it to act properly. So the kernel must know not only the type of device but also certain details of it (like density, partition no of it).

The fifth column shows the two no. which are major and minor numbers, major no. shows actually the type of device all hard disk or same kind of hardware will have the same major numbers but the minor will be different.

Minor no. tells the kernel about the parameters to operate the particular device.

**Significance of the Device Names:**

Unix device files is that the same device can often be accessed with several different filename, this is sometimes required to provide backward compatibility and for associating a separate device with specific function.

**The UNIX File System-**

* etc : Contains all system configuration files and the files which maintain information about the user and group.
* Bin : Contains all binary executables
* usr : default directory provided by UNIX OS to create users home directories and contains manual pages
* tmp: System or users temporary files which will be removed when the system reboots.
* dev: Contains all device files i.e logical names to physical devices.
* devices : Contains all the device files. i.e. physical names to physical devices.
* home : default directory allocated for the home directories of normal users when the administrator don’t specify any other directory.
* var : contains all the system log files and message files.
* sbin : Contains all the system administrator executables files.

**Partitions:**

Partition divides hard disk into many logically independent disk and accessed by its own device files. And to use a partition, it is requires to create a file system on it. File system defines a directory structure. Some rules and the advantages of the partitions are:

1. Data is growing day by day but a partition should restrict it spill it over to other partition.
2. If one area is corrupted then other should be shielded for that harmful things which corrupt that area.
3. If the system have adequate number of partitions, each partition can be backed up separately into a single volume of tape. The administrator can have different backup schedules for the different partitions.

**File System:**

In unix is configured with multiple file system where each has its own directory headed by root . but usually we see one root that’s because all have the same file system and becomes single one at a time of use.

Each file system is organized in a sequence of blocks of 1024 bytes and generally has three four components.

1. The boot block – contains a small boot program.
2. The super block – it contains global information about the file system and it also maintains a free list of inodes and data blocks that can be immediately allocated be the kernel when creating a file.
3. The inode block – contains a table for every file system. All attributes of a file or directory stored in this area except the name of the file.
4. The data block – contains the files or programs of the user.

**File System Components**

**Inode Blocks**

every file has an inode – a 128 byte table that contains virtually everything you could possibly need to know about a file. All inodes are stored into a contiguous inode blocks. Each 128 byte inode contains –

1. File type (regular, directory, device etc…)
2. Number of links
3. Numeric user id of the user
4. Numeric group user id of the user.
5. File mode
6. Number of bytes in a file
7. Date and time of the last modification of the file data.
8. Date and time of last access of the file data.
9. Date and time of last change of the inode.
10. The array of 15 pointers to the file.

Whenever a file is opened its inode is copied to the memory inode table and then kept synchronizing periodically with the disk copy.

And 15 pointers are required to get the different-2 information about a file using ls command.

**The Superblock**

It contains global file information about disk usage and availability of data blocks and inodes. So its information should therefore be correct for healthy operation of the system. It mainly contains –

1. The size of the file system.
2. The length of the file systems logical block.
3. Last time of updating.
4. The number of free data blocks available and free partial list of immediately allocated free data blocks.
5. Number of free inodes available and free partial list of immediately usable inodes.
6. The state of the file system.(clean or dirty)

Kernel also maintains a copy of the superblock in memory. It reads and write this copy when controlling allocation of inodes and data block.

So the copy cannot be newer in disk as in memory. And till the superblock is corrupt unix will not boot for this kind of problem many systems have multiple superblocks written in different area of disk.

**Data Blocks**

The smallest data block that can be read or written by the disk controller is 512 bytes. The kernel reads and writes the data using different block size and this block is often referred to as a logical block.

Every block is identifiable by an address- a number that refers to the position of block in data block area. The block containing data is known as direct blocks.

When a file expands the kernel may not find the adjacent block free. So it has to locate the expanded part in some other part of disk. This slows down read/write operation and leads to the disk fragmentation.

Inode keeps track of all the direct block addresses, but inode can store only 12 of them. So the file system also contains indirect blocks. And the inodes keeps track of all the indirect blocks.

**Boot Block**

This is the master boot record (MBR). The boot block contains the partition table and the a small “bootstrapping” program.

When the system is booted the system BIOS looks for the first hard disk and loads the entire segment of the boot block into memory. It then hands over the control to the bootstrapping program. This in turn loads the kernel into memory.

**Mounting and Unmounting File System:**

Before mounting the file system to the we need to check the logical name of it by issuing command

**$ sudo fdisk –l**

**How to mount/unmount Windows partitions (FAT) manually, and allow all users to read/write -**

e.g. Assumed that /dev/hda1 is the location of Windows partition (FAT)

Local mount folder: /media/windows

**\* To mount Windows partition**

**$ sudo mkdir /media/windows**

**$ sudo mount /dev/hda1 /media/windows/ -t vfat -o iocharset=utf8,umask=000**

**\* To unmount Windows partition**

**$ sudo umount /media/windows/**

**How to mount/unmount Windows partitions (FAT) manually, and allow all users to read/write -**

e.g. Assumed that /dev/hda1 is the location of Windows partition (FAT)

Local mount folder: /media/windows

**\* To mount Windows partition**

**$ sudo mkdir /media/windows**

**$ sudo mount /dev/hda1 /media/windows/ -t vfat -o iocharset=utf8,umask=000**

**\* To unmount Windows partition**

**$ sudo umount /media/windows/**

**How to mount/unmount network folders manually, and allow all users to read/write -**

e.g. Assumed that network connections have been configured properly

Network computer's IP: 192.168.0.1

Network computer's Username: myusername

Network computer's Password: mypassword

Shared folder's name: linux

Local mount folder: /media/sharename

**\* To mount network folder**

**$ sudo mkdir /media/sharename**

**$sudo mount //192.168.0.1/linux /media/sharename/ -o username=myusername,password=mypassword,dmask=777,fmask=777.**

**\* To unmount network folder**

**$ sudo umount /media/sharename/**

**To find out the free disk space (df) and disk usage (du) :**

**Disk-free-**

Unix file system as one single tree structure with a single root directory. But there can be multiple disks and each will have its own root and file system.

So to find out the free space in all the disk ‘df’ is the command.

**$ df**

**$ df partition\_name**

**Disk-Usage-**

It will show you all the usages of disk and also can find out the disk usage of specific folder and all the files of the folder.

**$ du – will show the details for all files and directories.**

**$ du /home/oracle/ - will show all the details for all the files and directories in home/oracle.**

**$ du –s /home/oracle – will show summary.**

**Compression of Files using - Compress, gzip and zip:**

To save the space compression is used.

Compress and gzip work with multiple files and remove the original files after it but the zip compress the files and retains the original file. But can also group multiple files into a single file.

**$ compress filename**

**$ uncompress filename**

**$ gzip filename**

**$ gunzip filename**

**$ zip filename**

**$ unzip filename**

**Printing a file ls and cancel–**

**$ lp filename**

**$ lp –dlaser filename – providing the printer name also.**

**$ lp –t “First” filename – will show First in the first page title.**

**$ cancel laser- cancel current job on printer laser.**

**$lpq - to know the number of jobs.**

**$ lprm id– if you want to remove any job of the printer.**

**$ lprm - - removes all the jobs owned by the user.**

**File System Checking fsck-**

Delaying updating superblock and inode block from the memory copy of both, leaves lot of scope for the file system inconsistency. If the power goes off or system shutdown due to some failure in that case superblock are not updated may cause serious error or corruption. The common are –

1. Two or more inodes claiming the same disk block.
2. A block marked as free but not listed in superblock.
3. An inode neither marked free nor in use or having a bad block number that is out of range.
4. Mismatch between the file size specified in inode and the number of data blocks specified in the address array.

Fsck commands are used to check and repair a damaged file system. On many systems, file systems are marked as “dirty” and “clean”. Fsck then checks only the dirty file systems during the next startup.

**$ fsck**

**$ fsck /dev/sda5/**

It does following things –

1. Phase 1- Check Blocks and Sizes
2. Phase 2 – Check path names
3. Phase 3 – Check connectivity
4. Phase 4 – Check Reference counts
5. Phase 5 - Check free list.

**Useradd, usermod and userdel –**

**User-add:**

**#useradd –u 210 –g dba –d /home/oracle –s /bin/ksh –m oracle**

Username –oracle

User ID – 210

Group – dba

Home Path - /home/oracle

Shell - /bin/ksh (optional)

-m option- ensures that the home directory is created added.

Useradd command creates the user entry in /etc/passwd

**#passwd oracle**

Will set the new password for the user oracle.

**/etc/passwd**: all the information of user stored in here except the encryption technique of password. The encryption information is stored in **/etc/shadow**.

**oracle : x : 210 : 241: /home/oracle : /bin/ksh**

**user-mod:**

**# usermod –s /bin/bash oracle**

used to modify some parameters of user. Like sometimes need to change the login shell.

**user-del:**

**# userdel oracle**

this commands removes the user from the system. This removes all entries but the users home directory does not get deleted.

**Role of init in startup and shutdown:**

The startup and shutdown process are controlled by automated shell scripts which are changed quite infrequently.

The kernel (/stand/unix, kernel/genunix or vmliuz) is loaded into the memory and it starts spawning further process. The most important of these is /sbin/init with PID 1, which is responsible for the birth of all subsequent processes. Role of init –

1. It controls the run levels (the system states) and decides which process to run (and kill) for each run level.
2. It spawns a getty process at every terminal or modem port so that users can log in.
3. It also ensures that all the system daemons are running.

**init Run levels-**

0 – system shutdown

1 – System administration mode (local file system mounted)

2 – Multiuser Mode

3 – Full Multiuser Mode

6 – Shutdown and reboot mode

S or s – single user more (file system mounted)

**# init 0**

To check the current run level you are using ‘**who –r**’.

**/etc/inittab –**

The behavior of init controls by /etc/inittab. Init reads all the process from inittab. Its fields determine the process that should be spawned for each of the **init** run levels.

**Backing Up Files (tar, dump, cpio, dd)-**

**dd:** it was a very powerful copying command and mainly for administrator. Previously it was used to copy file system but now its use is restricted to only the floppy diskettes.

Insert first floppy drive and copied data temporarily.

**#dd if=/dev/sda4/ of=$$ bs=147456**

Insert second floppy drive and temporary data is copied

**#dd if=$$ of=/dev/sda4/ bs=147456**

if =input filename

of = output filename

bs =block size