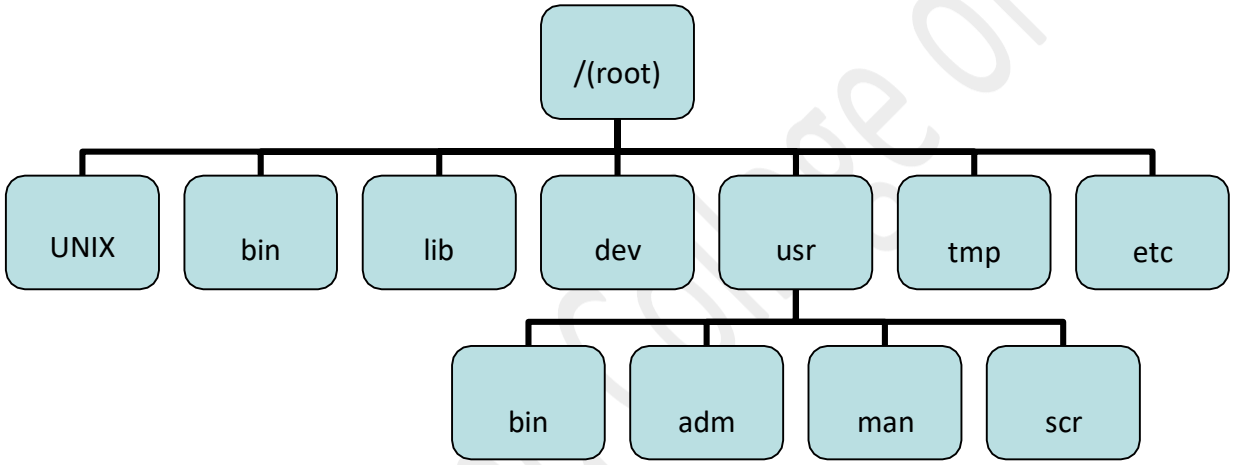
UNIX File System

All the Utilities, application, data in Unix is stored as Files. Even a directory is treated as a file which contains several other files. The UNIX file system resembles an upside down tree. Thus, the file system begins with a directory called *root*. The root directory is denoted as slash(/). Branching from the root, there are several other directories called *bin, lib, usr, etc, tmp, dev*. The root directory also contains a file UNIX, which is UNIX kernel itself. These directories are called sub-directories. Each sub-directory contains several files and directories called sub-sub-directories. The figure shows basic structure of Unix File system.



The main reason behind creation of directories is to keep related files together and separate them from other group. E.g. to keep all user related files in the *usr* directory, all device related files in the *dev* directory, all temporary files in *tmp* directory and so on.

1) The '/ ' directory (Root directory): The '/ ' directory is called as the root directory. It is also known as the super user's directory. The user who logs in as root will be placed in ' / ' initially.

2) The /bin directory: The /bin directory contains essential UNIX utilities. The word 'bin' stands for binary, hence this directory contains binary files as well as shell script. Binary files are executables files.

3) The /dev directory: The /dev or device directory is where all device files are kept. UNIX treats devices such as printer, disk storage devices, terminals and even areas of the computer's memory as a file. The files in /dev are termed as special files, since the file types are neither directory nor ordinary file. The /dev directory will contain a file for each device on the network. One important thing to note about the /dev directory is that there are no file sizes. Instead, there are ***major*** and ***minor*** device numbers.

The major number represents the type of device being referred to, while minor number distinguishes between possible instances of the same device or it indicates the special characteristic of the device.

$ ls –l /dev

```
total 52
brw-rw-rw-   1 root      sys     51,  0 Aug 28 07:28  cd0      CD-ROM
brw-rw-rw-   2 bin       bin      2, 64 Mar 30   1997  fd0      Default floppy drive
brw------    1 sysinfo   sysinfo  1,  0 Mar  8    1996  fd0      First hard disk
crw------    2 bin       bin      6,  1 Dec  5   1997  fd0      Default floppy drive
```

File types in /dev directory are not ordinary or directory instead the type of files is either ***Character Special*** or ***Block Special***. A character device is simply a device from which characters are read. These include terminals, printers and even raw memory. A block device is a device that is used only for the file system and can only accessed by users via standard input/output request such as for using the various UNIX utilities.

4) The /etc directory: The /etc directory is a directory that contains various administrative utilities together with other special system files that allow the host Unix System to start up properly at bootstrap time.

   Utilities for the handling of the system's terminal devices are stored in /etc, as are the lists of all the registered users of the system, including you.

5) The /tmp directory: The /tmp directory contains the temporary files created by UNIX or by the Users. These files get automatically deleted when system is shutdown and restarted.

6) The /usr directory: The /usr directory contains several directories, each associated with a particular user. The System administrator creates these directories when he creates accounts for different users. Each user is allowed to work with his directory called *home* directory.

   a) The /usr/bin directory: Within ***usr*** directory there is another **/bin** directory which contains additional Unix command files that are more important to the end user.

   b) The /usr/adm directory: This directory is a part of /usr directory. The 'adm' stands for administrative in this context, hence the /usr/adm directory contains administrative utilities for user by the systems administrator, although most user will not be able to access them.

Vivekanand College of BCA

c) The /usr/man directory: Unix has a few online help facilities of which man is one. Man is the Unix manual feature, a small Unix utility program, that enables access to information on all Unix utilities.

d) The /usr/scr directory: This is a directory that contains the source-code for many of the binary utilities.

7) The /lib directory: The /lib directory contains all library functions provided by Unix for programmers.

Features of UNIX File System

Following are the features of UNIX file system.

1) It is hierarchical file structure:

The Unix system organizes its files using upside-down hierarchical tree structure. All files will have a 'parent' file, apart from a directory called 'root' directory, which is the parent of all files on the system. This hierarchical component also adds to the dynamic flexibility of the file system.

2) Files can grow dynamically (or dynamic file expansion):

The file system structure is dynamic, that is to say, its size is not determined by any rules other than the amount of disk storage that it is available on the System. A file's size can be changed by user at any time.

3) Files have access permission:

Files are protected using file ownership mechanisms. These allow only a specific class of user to access certain files.

4) Structureless files :

It contains sequence of bytes in a file

5) All devices are implemented as files :

E.g. hardware devices such as printer, terminal, disk devices all are treated as files.

Physical file structure of Unix

A file system is a group of files and relevant information regarding them. Your whole hard disk may comprise a single file system of it may be partitioned into several file system.

Vivekanand College of BCA

The disk space allotted to a Unix file system is made up of 'blocks' each of which are typically of 512 bytes. Some file systems have blocks of 1024 or 2048 bytes as well.

All the blocks belonging to the file system are logically divided into four parts . 1) Boot Block, 2) Super Block, 3) I-node Block, 4) Data Block.

Boot Block	Super Block	I-node Block	Data Block

- ***Boot Block:***

  Boot Block or Block 0 is the first block of the file system. This block is normally reserved for booting procedures. It is also called Master Boot Record (MBR). The boot block contains the partition table and a small 'bootstrapping' program. When the system is booted, the system BIOS checks for the existence of the first block of hard disk and loads the entire segment (code) of the boot block into memory. It then hands over control to the bootstrapping program. This in turn loads the kernel (the file unix) into memory. However, the bootstrapping program is read in form the boot block of the root(main or track zero) file system. For other file system, this block is simply kept blank

- ***Super Block:***

  The Super Block or Block 1 is the 'balance sheet' of every Unix file System. It contains global file information about disk usage and availability of the data block and I-nodes. It contains

  - The Size of the file system.
  - The length of the file system's Logical block
  - Last time of Updating
  - The number of free data blocks available and a partial list of immediately allocable free data blocks.
  - Number of free I-nodes available and a partial list of immediately usable I-nodes.
  - The state of the file System (whenever 'clear' or 'dirty')

  As with I-nodes, the kernel also maintains a copy of the Super Block in memory.

- ***I-node Block:***

  We know that all entities in UNIX are treated as files. I-nodes are used to describe the file characteristic and the locations of the data blocks which store the contents of the system. The information related to all these files (not the contents) is stored in an I-node table on the disk. For each file, there is an I-node entry in the table. Each entry is made up of 64 bytes and contains the relevant details for that file. These details are:

Vivekanand College of BCA

- File Type (regular, directory, device etc)
- Number of links (the number of aliases the file has)
- The numeric UID of the owner.
- The numeric GUID of the owner.
- File mode(file access permission)
- Size of the file
- Date and Time of last modification of file data
- Date and Time of last access of file data
- Date and Time of last change of the I-node
- Addresses of blocks where the file is physically present

The I-node is accessed by number called *I-node Number.* This number is unique for every file in a single file system.

- ***Data Block:***

The remaining space of the file system is taken up by Data block or Storage block. For a directory each data block contains 16 bytes entries. Each such entry would have a file or subdirectory name up to 14 bytes and 2 bytes would be taken for the I-node. Thus, for directories, the table of filenames and their I-nodes is available.

e.g. (directory called /usr)

14 bytes	2 bytes
File name 1	I-node 1
File name 2	I-node 2
File name 3	I-node 3

The operation system's files, all data and programs created by users reside in this area.

Vivekanand College of BCA

Structrue of Inode

The inode is created, whenever a file is created. The inode is deleted whenever the related file is removed. The inode contains the following information:

1) **Owner's user id (uid) :** user id of owner
2) **Owner's group id (gid) :** the group id the user belongs to
3) **Protection bits:** UNIX divides all the users into three categories: owner, Group and the others. Each user can have read (r), write (w) and execute (x) rights. For each of these right, one bit is necessary. So 9 bits (3 bits for rights X 3 bits for 3 types of users) are required for each file in its inode.
4) **File type:** The file type specifies that a file is an ordinary file, a directory file, a FIFO file or a special file.
5) **File access times:** The file access times specify the time at which the file was created, last used and last modified.
6) **Number of links :** Number of links specify the number of symbolic name the file is known by. The kernel physically delete a file (free all the data blocks and the inode allocated to that file) only if this number is 0. when all the users delete this file from their directories than number of links becomes 0.
7) **File Size:** The size of the file.
8) **Addresses** of all the blocks allocated to this file, direct blocks (data blocks) or indirect blocks (index blocks that points to data blocks).

Each inode of particular file contain the above information. The inode reserves the memory to maintain block numbers of 13 blocks. Assume that a block is of 1024 bytes. 4 bytes are required to represent 1 block number. So each inode require 4 x 13 =52 bytes. Out of these 13 blocks, 10 block are pointed to actual data blocks allocated to the file. This 10 blocks are numbered from 0 to 9. This 10 blocks are referred to as "Direct Blocks".

Vivekanand College of BCA

As a new file is created, initially all the 10 direct entries in inode are emply. When something is to be written in a file, a block is required, the kernel goes to the list of free data blocks, choose a free data block, remove the selected free block number from the list of free data blocks and insert this block number in inode. This process continue upto first 10 block in a file.

Single Indirect Block

When all the first 10 blocks are filled, and new free block is required , the kernel allocates a data block from the available free blocks, and used create an index block in which it insert and entry for newly assigned block number. The 11^{th} entry in inode points to the index block. This index block also has 1024 bytes. So it contain 1024 / 4=256 entries of block number which points to actual data blocks. This index block is called a “Single indirect” block.

10+256 block=266 block=266 kb

Double Indirect Block

In double indirect the 12^{th} entry in the inode points to a double indirect index block. This index block has 256 entries again, but instead of pointing to 256 data blocks directly they point to 256 additional index blocks which in turn points to actual data blocks.

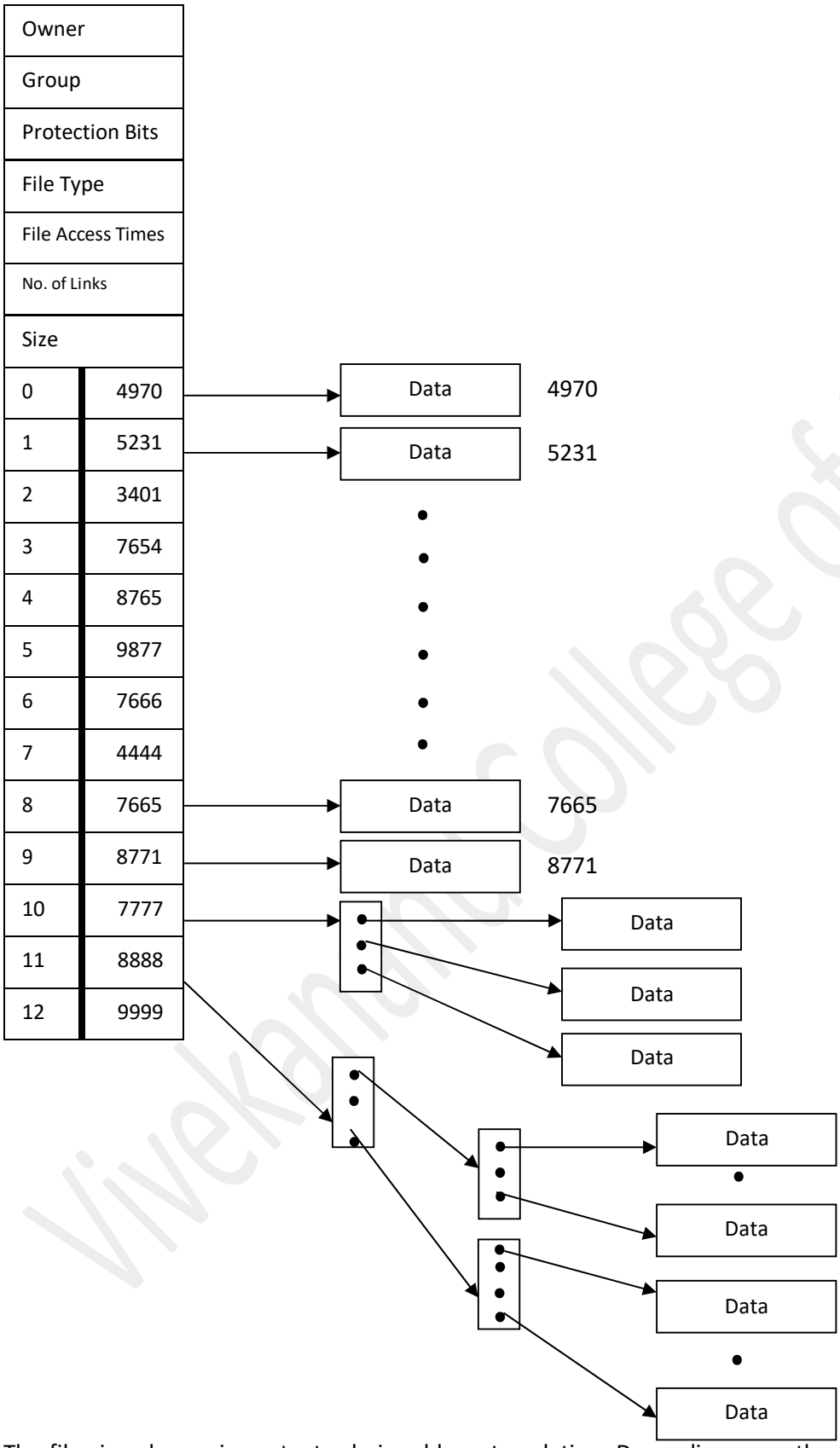
Total number of capabilities of addressing at double indirect index is

10+256+(256x 256) = 65802 blocks ir 65.8 MB.

Total number of capabilities of addressing at Triple indirect index is

$10+256^1+256^2+256^3$ = 65802 blocks ir 65.8 MB.

Vivekanand College of BCA



The file size play an important role in address translation. Depending upon the file size, the kernel knows whether to locate the block number in the inode itself (direct), or in an index at a specifil level of indirection.

Vivekanand College of BCA

Booting Sequence / Stages of System Startup

1) **ROM diagnostics are run:**
   Hardware and memory tests are performed
2) **Boot loader is loaded:**
   The boot loader is loaded into RAM by the ROM
3) **Kernel is loaded:**
   The kernel program is called by the user. Recall that the user needs to enter the name of the program. (E.g UNIX, HICIX etc at prompt during boot process which is optional. By default UNIX kernel is loaded).
4) **Kernel initialization takes place:**
   It involves performing memory tests, initializing the devices through device driver, swapper scheduler process and the init process.
5) **/etc/init or the init program** is invoked by the kernel. This program takes over the SystemControl.
6) **The init program:** This process has the PID1 (Process IDentification) which is the second process of the system. The init program reads the instruction of /etc/inittab (configuration file used by init program) file to carry out processes like identifying the run level i.e the mode in which system should run single user/multi-user, maintaining files to track activities etc..

7) The getty process : The init program also invokes the getty program which establishes a communication with the terminals of the UNIX system. The getty program uses the file called /etc/gettydefs for instructions to provide the login: prompt at each terminals connected to system and goes into suspended (sleep) mode and activated when user attempt to login.

8) The login program: Once the user types login name and password, getty transfer control to a login program, to verify the login-name and password entered by user. Thus if the login is successful the $ prompt appears on the screen. It is essential to know that it is the /etc/profile that enables the system administration to provide information like time and date, system name, number of users etc. This file can be modified by the system administrator to accommodate relevant messages for the users.

init -> getty -> login -> shell

Vivekanand College of BCA

(Fig: process of leading to shell)

Now init, is the only living ancestor of the shell which is goes off to sleep mode and waiting for the death of its children. When the user logs out, shell is killed and the death is intimated to init, init then wakes up and spawns another getty for that time to monitor the next login.

Structure of /etc/passwd

All user information except the encrypted password is stored in /etc/passwd. The encrypted password is stored in /etc/shadow. The /etc/passwd shows the entry of all registered users. Each entries consisting of seven field. Each field is separated by colon (:).

Username: password : UID : GID : Comment : home directory : login shell

(Fig: The structure of /etc/passwd file)

The following are the significance of all the seven fields:

1) **Username :** It is login name of the user.
2) **Password:** Encrypted password of the user, but it contains an x. The encrypted password is stored in /etc/shadow file.
3) **UID :** UID stand for User IDentification number. It is unique number, No two users have the same UID.
4) **GID :** GID stands for Group IDentification number. It is also unique number.
5) **Comment :** This field contains detailed information about the user such as full name of the user, address etc.
6) **Home directory :** It is the home directory of a user. When a user login to the unix system, initially he is placed in this directory.
7) **Login shell :** When the user logs in then default shell is allocated. Using this shell user communicate with the kernel.

Vivekanand College of BCA

The typical entries of /etc/passwd is as follows:

Kamal : x : 210 : 100 : kamal patel, surat : /home/kamal : /bin/ksh

Nirmal : x : 211 : 100 : nirmal patel, surat : /home/nirmal : /bin/ksh

Karan : x : 212 : 100 : karan patel, surat :/home/karan : /bin/ksh

Structure of /etc/shadow

The /etc/shadow stores actual password of the user in encrypted format. The general structure of /etc/shadow file is as follows.

username : pwd : Last pwd change : minimum : maximum : warn : Inactive : Expire : reserve

(Fig: The structure of /etc/shadow file)

This file contain 9 fields, each field is separated by colon. This file contain the encrypted password information for user's accounts and optional password aging information.

1. **Username :** It is login name of the user.
2. **Pwd :** It contains encrypted password. The password should be minimum 6-8 characters long including special characters / digits.
3. **Last password change :** Days since Jan 1, 1970 that password was last changed.
4. **Minimum :** The number of days left before the user is allowed to change passwordor minimum days before password may be changed.
5. **Maximum :** The maximum number of days the password is valid or days after which password must be changed.
6. **Warn :** The number of days before password is to be expire the user is warned that his / her password must be changed or days before password is to expire that user is warned.
7. **Inactive :** The number of days after password expires that account is disabled.

Vivekanand College of BCA

8. **Expire :** days since Jan 1, 1970 that account is disabled. i.e. an absolute date specifying when the login may no longer be used.
9. **Reserve :** it is a reserved field.

A typical entry of /etc/shadow is as follows:

Kamal :

2Cd3eesdf3$dfdfeefefefeffe23dfdf.:14433::88888::::

Nirmal :

3Ef3eeshj2$dfdjhhferefeggh23dfdf.:14422::88888::::

Vivekanand College of BCA