

## File System



### File System(Files and directories)

#### Directory commands

- mkdir, cd
- rmdir ls

#### File commands

- touch , cat, vi editor
- rm ,cp, mv
- link

## UNIX File System



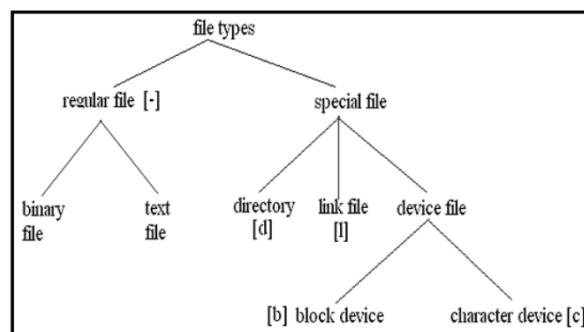
#### •File System:

- It is the collection of files and directories organized in a hierarchical structure(inverted tree structure)

#### •File:

- Container for storing information.
- UNIX imposes no internal structure on a file's content. The user is free to structure and interpret the file content
  - Example: The employee data can be stored as
    - 1:amit:25000:developer (or)
    - 1 amit 25000 developer
- UNIX treats directories (folder for storing filenames and other directory names) and devices(hard disk, memory, printer etc.) as files only

## File types



## File types

- These files can contain text data or binary data
- The contents of the text file are readable by the user
- The contents of the binary file are unreadable by the user and readable by the system or an application
- The binary files may be executed
- The files are denoted by (-)

### Ordinary or regular files

- Contains the information about the files and the subdirectories
- It has no data
- The directory has 2 entries for each file or subdirectory it contains
  - filename
  - a unique number to identify the file in the hard disk (inode number)

### Directory files

## File types

### Device files

- Represents hardware devices and are present in the /dev directory
- *Character special files* or *character devices* relate to devices through which the system transmits data, one character at a time. Denoted by (c)
- *Block special files* or *block devices* correspond to devices through which the system moves data in the form of blocks. These devices often represent addressable devices such as hard disk, drives. Denoted by (b)

## File System Elements

### Rules for a file name

- Filename can consists up to 255 characters
- UNIX has no concept called file name extension
- It can have any ASCII character except the / and the NULL character
- The recommended characters are
  - alphabetic characters and numerals
  - period (.) , hyphen (-) and underscore ( \_ )
  - can't have a file name having 2 consequent periods
- File names starting with ( . ) are hidden from the user
- The file names are case sensitive

## Summary on File Types:



### 1. Regular file

- binary file
- text file

### 2. Special file

- Directory file
- Link file
- Device file
  - Block device file
  - Character device file

### Symbols used to identify the file type

- ( - ) Regular file
- ( d ) Directory file
- ( l ) Link file
- ( b ) Block device file
- ( c ) Character device file

## File System Single Rooted Hierarchy



↓  
Boot lib etc bin sbin usr home export opt mnt var dev tmp

/boot	-	Boot image of the kernel
/lib	-	Library files needed for the operating system and applications
/etc	-	Configuration files storage area of system and applications
/bin	-	Executables that can be executed by normal users
/sbin	-	Executables for super user (root)
/usr	-	Application's storage area
/home	-	Area of mount the remote home directories of users
/export	-	System's normal users home directory
/opt	-	optional (provides storage for large, static application software packages)
/mnt	-	Common mounting area
/var	-	Variables area ( www, mail, log, etc... )
/dev	-	Device files directory
/tmp	-	System temporary area, Globally accessible

## Special Files: DIRECTORY



- NAME INODE NUMBER
- NAME is the file name or subdirectory name
- INODE NUMBER is the unique number given to the file or subdirectory in a given file system

### Directory contents:

- An *inode* is a *structure* in a file system on UNIX operating systems that stores all the information about a file except its name and its actual data
- The inode table is a data structure that contains a list of inodes
- Inode will not contain inode number. It is just an index of the entry in the inode table

### INODE:

## INODE Entries

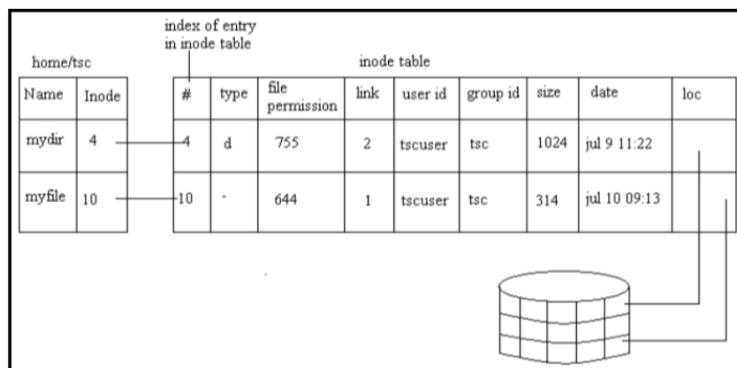


Type	Type of the file (- d b l)
Permission	File Access Permissions
Ref. count	Hard link count
User	Owner of the file
Group	Primary group info
Size	size of the file
Disk Address	Location of a file in the hard disk
Time	created/modified/accessible time

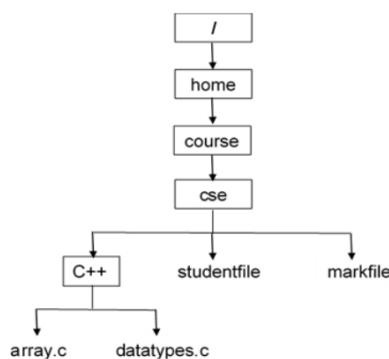
The part of the inode information can be viewed using the command

```
$ stat <filename>
```

## Contents of inode



## Sample Directory structure:



## Path names

### Path:

- The sequence of file names, separated by / describes the path
- The path is used to locate the file in the system

### Types of path:

- **Absolute/Full path:**
  - Starts from the root always ( / )
  - e.g /home/course/cse/C++
- **Relative path:**
  - Starts from the current directory where the user is located
  - The path is said to be relative unless it starts with /
  - e.g: ./C++

### Where Am I

`pwd`

The Print Working Directory is used to print the current working directory of the user

## Directory commands

### Creating a directory:

`mkdir`

- make directory
- **syntax:**
  - `mkdir <directory name>`
- **example:**
  - `$mkdir sample`

## Directory commands

### Deleting a directory:

`rmdir`

- remove directory
- **syntax:**
  - `rmdir <directory name>`
- **example:**
  - `$rmdir sample`

### constraints on removing a directory:

- The directory should be empty
- The directory name must not be the current working directory, and it is any parent directories name



## Directory commands

Traversing from one directory to other:

cd

change directory

It changes the current directory to the directory specified as argument for the cd command

**syntax:**

cd [<directory name>]

**example:**

\$ cd sample (relative path) now the current working directory is sample

\$ cd /home/tech/cap/capuser/sample (full or absolute path)

## Directory commands



Traversing from one directory to other:

The other ways of navigating across the directory :

- cd - takes the user to their home directory
- cd ~ - same as cd
- cd . - remains in the current directory - No change
- cd .. - takes the user one level above the current directory
- cd - - takes the user to the previous visited directory

## Directory commands



**Listing directory contents:**

The path of the directory name can either be relative or absolute

**ls**

To obtain list of all file names and subdirectory names in the current directory

- syntax:
- ls [directory name]

If no directory name is given, the current directory will be used as the arguments

## Directory commands



### Listing directory contents:

#### example:

```
1) $ ls  
    o/p: Cdir markfile studentfile  
2) $ ls Cdir/  
    o/p: array.c datatype.c
```

#### options in ls command:

- a - list all the files including the hidden files
- l - long listing of the directory contents. Gives information about the file or directory like name, permission, size, date etc
- i - displays the inode number of the files

#### options to explore:

```
-d, -R, -r, -h
```

## File commands



### creation of files:

#### cat:

used to create file and display the content of the file

##### – creation:

- syntax:
  - cat > filename
  - when the command is given, the system waits for the user to enter the contents of the file
  - To save and exit from the command, ctrl+d is given

##### – example:

```
$ cat > samplefile  
hi this is a test file  
[ctrl+d]  
$
```

## File commands



cat command is also used for concatenating files and printing the result on the standard output

#### syntax:

```
cat [<filename[s]>]
```

displays the content of the file, if the file already exists

#### Example:

```
$ cat samplefile  
hi this is a test file  
$ cat samplefile testfile  
hi this is a test file  
this is the content  
(it prints the content of samplefile and then the testfile)
```

## File commands

### deleting files

#### rm:

- removes files or directories
- syntax:  
• rm <filename[s]>

#### examples:

```
$ rm testfile      -deletes the file  
$ rm -r mydir     -recursive delete (Deletes the entire subtrees along with  
the specified directory)  
$ rm -i mydir     -interactive deletion  
$ rm -rf testfile  -forceful deletion (Deletes the directory forcefully even if  
the directory is not empty)
```

## File commands

### copying file:

#### cp:

- copies a file or a group of files
- It creates an exact image of the file on the disk
- It can have different names, if it is in the same directory
- same or different name in another location
- Has a different inode number

#### syntax:

- cp <source> <destination>
- There can be more than one source file
- If there is more than one source file, then the destination should be a directory

## File commands

#### cp:

#### example:

```
$ cp testfile sample          (sample can be a file or directory)  
$ cp testfile1 testfile2 sample (sample should be a directory)
```

#### options to explore:

-r, -i, -f

## File commands

### **moving files:**

#### **mv:**

- moving a file/directory or a group of files/directories
- renaming file or directory

#### **syntax:**

- mv <source> <destination>
- If there is more than one source, the destination should be a directory.

#### **example:**

- mv testfile sample

#### **options to explore**

- -i, -f

## File commands

### **linking files:**

#### **In**

make links between files

#### **Types of links:**

##### **• hard link**

- alias name for the file

- The inode number remains the same

- The reference count of the file increases by 1

#### **Restrictions in hard link:**

- Can't be created for directories
- Only a single copy of the file is maintained, others are just references pointing to the original file
- Can't be referred across file systems

## File commands

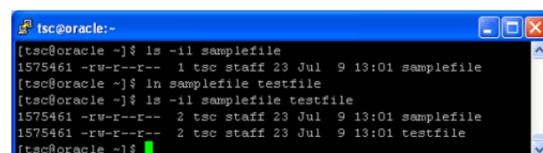
### **Hard link continuation**

#### **syntax:**

- ln <sourcefile>[ <destinationfile>]
- If link name is not given then the source name is taken as the link name, if the link is not created in the same directory

#### **example:**

```
$ ln samplefile testfile
```



```
tsc@oracle:~]$ ls -il samplefile
1575461 -rw-r--r-- 1 tsc staff 23 Jul  9 13:01 samplefile
[tsc@oracle ~]$ ln samplefile testfile
[tsc@oracle ~]$ ls -il samplefile testfile
1575461 -rw-r--r-- 2 tsc staff 23 Jul  9 13:01 samplefile
1575461 -rw-r--r-- 2 tsc staff 23 Jul  9 13:01 testfile
[tsc@oracle ~]$
```

## File commands

### soft or symbolic link

- used for creating shortcuts for the files and directories
- Inode numbers are different
- The soft link file's content is only the path of the original file
- The type of the file for soft link is denoted by l

### Syntax

- ln -s <sourcefile> <destinationfile>

### example:

• \$ ln -s studentfile student

## File commands

### Example of soft link

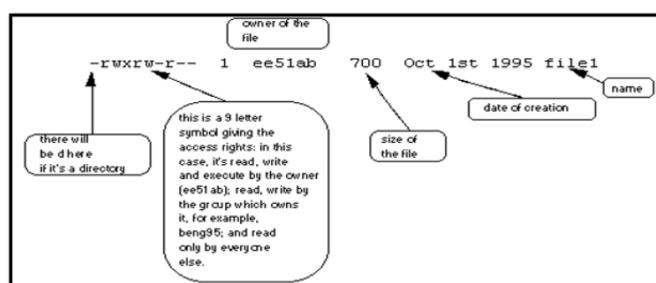
```
tsc@oracle:~$ ls -il studentfile
1575452 -rw-r--r-- 1 tsc staff 0 Jul  9 12:45 studentfile
[tsc@oracle ~]$ ln -s studentfile student
[tsc@oracle ~]$ ls -il studentfile student
1575450 lrwxrwxrwx 1 tsc staff 11 Jul  9 15:34 student -> studentfile
1575452 -rw-r--r-- 1 tsc staff 0 Jul  9 12:45 studentfile
[tsc@oracle ~]$
```

## Permission

Permission is used to describe the file's read, write and execute rights available to three categories of users – user, group and others

Permission can be altered only by the owner of the file with chmod

The permission of a file is displayed in the 1<sup>st</sup> column of the ls -l command



## Permission continued ...



### Users of a file:

- owner:
  - Every file created in the UNIX system will have one user as a owner
- group:
  - Every user belongs to at least one primary group
- others:
  - The users who don't belong to the file's group and are not the owners are called others. Generally they are termed as public

**id** → Command displays the user and group id

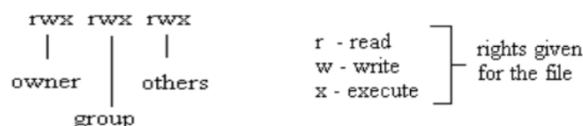
```
$ id  
uid=555(tsc) gid=503(staff) groups=503(staff)
```

## Permission continued ...



### chmod:

change file access permissions



syntax:

```
chmod <permission> <filename>
```

## Permission continued ...



Permission for file:

- |                 |  |
|-----------------|--|
| <b>read:</b>    | reads the contents of the file (example : cat, vi) |
| <b>write:</b>   | modifies the contents                              |
| <b>execute:</b> | allows to execute the file if it is executable     |

Permission for directory:

- |                 |  |
|-----------------|--|
| <b>read:</b>    | reads the contents of directory (example : ls)                             |
| <b>write:</b>   | modifies the directory content (example: cp, mv, create and remove a file) |
| <b>execute:</b> | open or enter the directory (example: cd)                                  |

## Permission continued ...



- **symbolic notation**
  - Used to assign or remove particular rights from a particular category of user
- **octal notation**
  - Need not know what the current permissions are
  - All the permissions for all the category of users should be explicitly set

### Types of assigning permission

## Permission continued ...



#### **symbolic notation:**

- ◆ r read
- ◆ w write
- ◆ x execute
- ◆ u owner
- ◆ g group
- ◆ o others
- ◆ + add permission
- ◆ - remove permission
- ◆ = assign permission
- ◆ a assign permission to all the user (u g o)

## Permission continued ...



#### **example:**

- chmod u+x inputfile → (add execute permission to the owner)
- chmod a+w inputfile → (add write permission to all the users)
- chmod u+x, g-r inputfile → (add execute permission to the owner and remove read permission from the group)
- chmod go-rw inputfile → (remove the read and write permission from the group and others)
- chmod ugo=r inputfile → (assign read permission to all the users. After executing this command only, read permission will be present for all the users)

## Permission continued ...



- **octal notation:**

	binary notation	octal notation
read	100	4
write	010	2
execute	001	1

- To grant all the permissions to a category (read, write, execute) :  
 $4+2+1 = 7$

**example:**

- chmod 644 inputfile (644 → 6 for user, 4 for group, 4 for others)
- chmod 622 inputfile

## Permission continued ...



### umask

- umask stands for user mask
- umask tells the restricted permission for each user
- the default umask value is 022
- if the umask value is 022 then
  - the default file permission is 644
  - the default directory permission is 755
- Every file or directory created will have this default permission
- The maximum default file permission is 666
- The maximum default directory permission is 777