Review

```
rekha@rekha-ThinkPad-P50:~/workingDir$ ls -l foo
-rw-rw-r--     1      rekha   test-demo 40   Aug  5      21:59      foo
```
Type Mode   links owner group   size mod-date mod-time name

```
rekha@rekha-ThinkPad-P50:~/workingDir$ cat /etc/group | grep test-demo
test-demo:x:1006:user3,rekha,user2
```
Groupname:passwd:groupID:users'list

```
rekha@rekha-ThinkPad-P50:~/workingDir$ cat /etc/passwd|tail -3
user2:x:1002:1003:,,,:/home/user2:/bin/bash
user3:x:1003:1005:user3,34,857345345,980932850285:/home/user3:/bin/bash
user1:x:1001:1002:user1,34,094809285034,203948208435:/home/user1:/bin/bash
```
Username:passwd:userID:groupID:user info:home directory:command/shell

Logout and login as user1 and try to write on file foo

```
E45: 'readonly' option is set (add ! to override)
```

Logout and login as user2 and try to write on file foo

Review

Can one user see other user’s files?

r-x:directories r--:files

```
rekha@rekha-ThinkPad-P50:/home/user1$ ls -l /home/user1
total 44
drwxr-xr-x 2 user1 user1 4096 Aug  5 22:34 Desktop
drwxr-xr-x 2 user1 user1 4096 Aug  5 22:34 Documents
drwxr-xr-x 2 user1 user1 4096 Aug  5 22:34 Downloads
-rw-r--r-- 1 user1 user1 8980 Aug  5 22:07 examples.desktop
drwxr-xr-x 2 user1 user1 4096 Aug  5 22:34 Music
drwxr-xr-x 2 user1 user1 4096 Aug  5 22:34 Pictures
drwxr-xr-x 2 user1 user1 4096 Aug  5 22:34 Public
drwxr-xr-x 2 user1 user1 4096 Aug  5 22:34 Templates
drwxr-xr-x 2 user1 user1 4096 Aug  5 22:34 Videos
rekha@rekha-ThinkPad-P50:/home/user1$ ls -l /home/user2
total 44
drwxr-xr-x 2 user2 user2 4096 Aug  5 21:46 Desktop
drwxr-xr-x 2 user2 user2 4096 Aug  5 21:46 Documents
drwxr-xr-x 2 user2 user2 4096 Aug  5 21:46 Downloads
-rw-r--r-- 1 user2 user2 8980 Jul 31 10:03 examples.desktop
drwxr-xr-x 2 user2 user2 4096 Aug  5 21:46 Music
drwxr-xr-x 2 user2 user2 4096 Aug  5 21:46 Pictures
drwxr-xr-x 2 user2 user2 4096 Aug  5 21:46 Public
drwxr-xr-x 2 user2 user2 4096 Aug  5 21:46 Templates
drwxr-xr-x 2 user2 user2 4096 Aug  5 21:46 Videos
```

Review

user2 can not delete file owned by user1 unless it is an sudoer.

```
rekha@rekha-ThinkPad-P50:/home/user1$ ls -ld
drwxr-xr-x 15 user1 user1 4096 Aug  5 23:30
rekha@rekha-ThinkPad-P50:/home/user1$ su user2
Password:
user2@rekha-ThinkPad-P50:/home/user1$ rm error
rm: remove write-protected regular empty file 'error'? yes
rm: cannot remove 'error': Permission denied
user2@rekha-ThinkPad-P50:/home/user1$ sudo rm error
[sudo] password for user2:
user2 is not in the sudoers file.  This incident will be reported.
user2@rekha-ThinkPad-P50:/home/user1$ more /etc/group|grep sudo
sudo:x:27:rekha,user3
user2@rekha-ThinkPad-P50:/home/user1$ su user3
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
user3@rekha-ThinkPad-P50:/home/user1$ ls
Desktop    Downloads  examples.desktop  Pictures  Templates
Documents  error      Music             Public    Videos
user3@rekha-ThinkPad-P50:/home/user1$ sudo rm error
[sudo] password for user3:
user3@rekha-ThinkPad-P50:/home/user1$ ls
Desktop    Downloads         Music     Public     Videos
Documents  examples.desktop  Pictures  Templates
```

Review

<u>Userful commands</u>

Adduser – creating a new user
Addgroup – creating a new group
Usermod – modify user account
id – display user and group information
groups – display all the groups user is in
passwd – change the user password

su – switch user
sudo – act as super user for this command

chmod - change file mode bits
umask – set file mode creation mask
chown – change file ownership on file
chgrp – change group ownership on file

File Descriptors

- Unix considers everything as a file system.
- Keyboard is a readonly file and screen is a write only file.
- Folders and input-output devices are also considered to be files.
- Whenever connection is opened on a file, the kernel allocates a file descriptor, an integer that specifies the access to that file such it being read only, write only etc.
- There is a difference between a file and an open "connection" to a file.
- Dedicated file descriptors:
  - ✓ file descriptor 0 is a processes' stdin
  - ✓ file descriptor 1 is a processes' stdout
  - ✓ file descriptor 2 is a processes' stderr

Files/Connections/descriptors/Tables

- File Descriptor Table: Each process has a file descriptor table that gives the mapping between the descriptor the process uses to refer to a file connection and the data structure inside the kernel that represents the actual file connection.

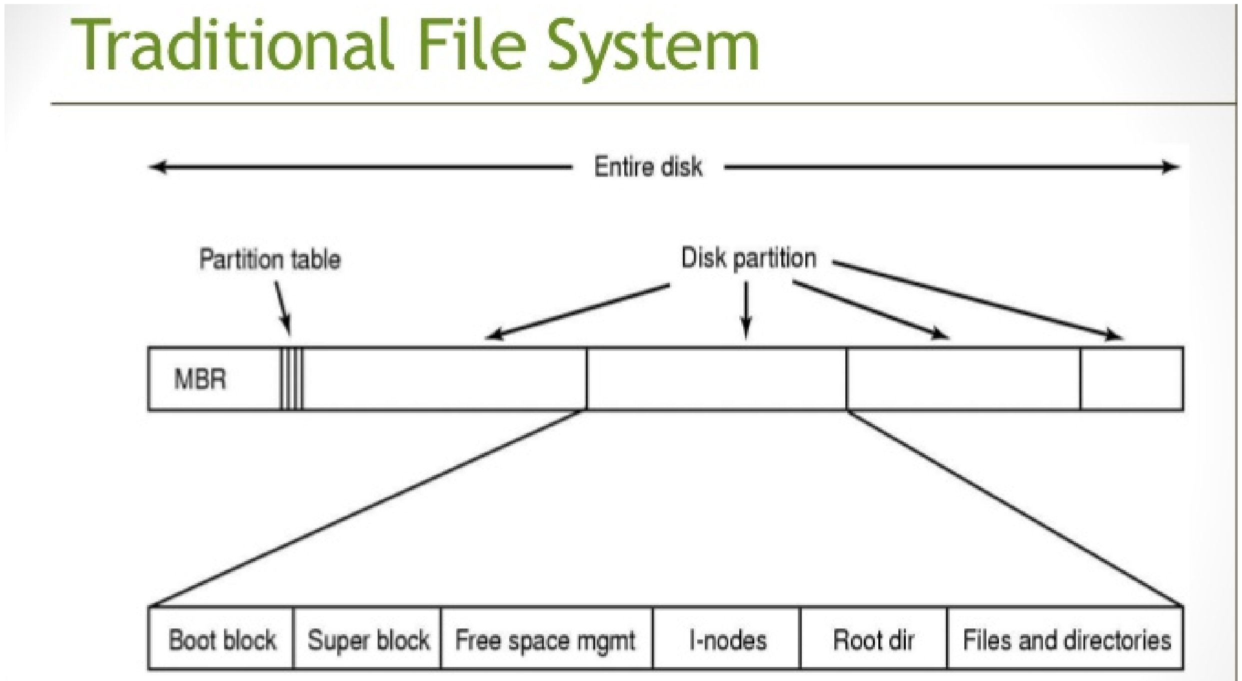
- System open-file table: OS has an entry for each open connection on this table. Each entry contains the connection status, e.g. read or write, the current offset in the file, and a pointer to a vnode, which is the OS's structure representing the file.

- Vnode table: To support filesystem independent layer, OS implements vnode table for each open file or device, which contains information about the type of file and pointers to functions that operate on the file. Typically for files, the vnode also contains a copy of the inode for the file, which has "physical" information about the file, e.g. where exactly on the disk the file's data resides.

- The physical device: inodes: File data may be widely distributed across the physical drive, but the inode for the file contains the locations of each of the data blocks comprising the file. Directories have directory blocks instead of data blocks, which contain inode/filename pairs.

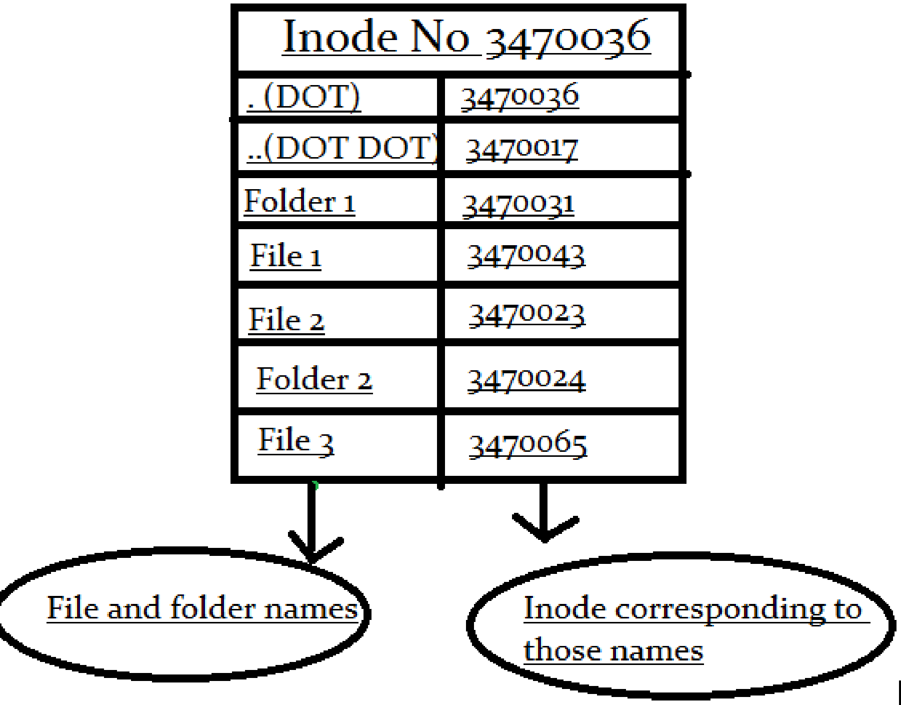
Files System

- The boot block is the first sector/block in a file system which contains the bootstrap code that is required to boot the system.
- Super block describes the state of the file system i.e. its size, maximum number of files that can be stored, the location of the inodes for the root directory and the free space information for inodes and data blocks both.
- File's inode is an index into a table of so-called inode blocks that describe all file properties except the file name.
- The data block is the end of the inode list and starting of the blocks that can be used to store the user files.



Files System : Inode

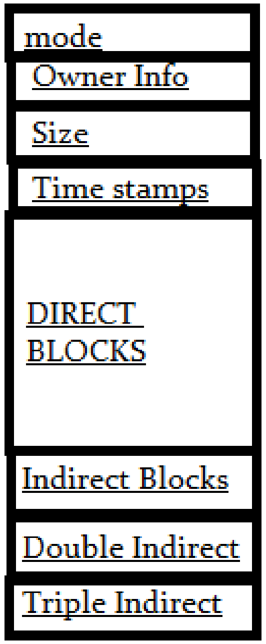
Inode Structure of a Directory:

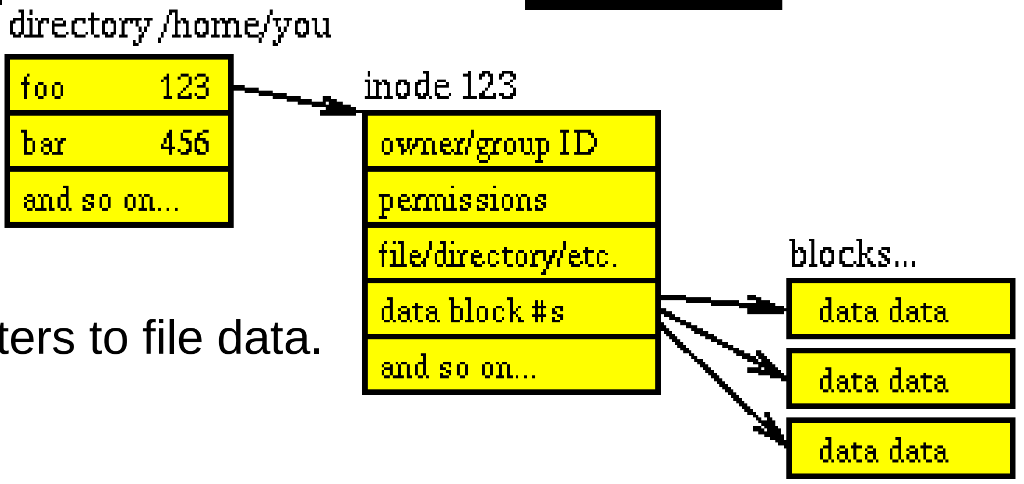


Inode No 3470036	
. (DOT)	3470036
..(DOT DOT)	3470017
Folder 1	3470031
File 1	3470043
File 2	3470023
Folder 2	3470024
File 3	3470065

File and folder names
Inode corresponding to those names

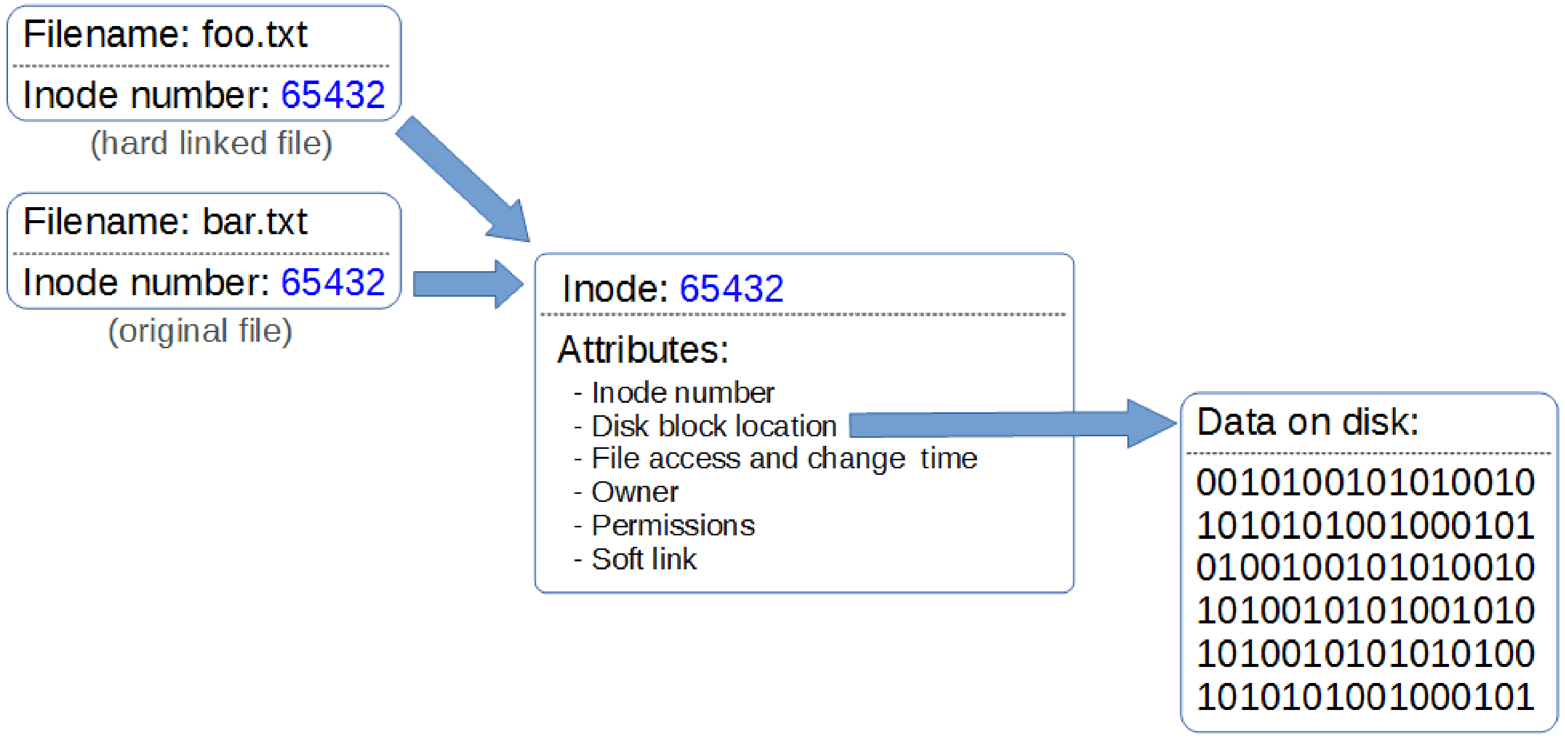
Inode Structure of a File:





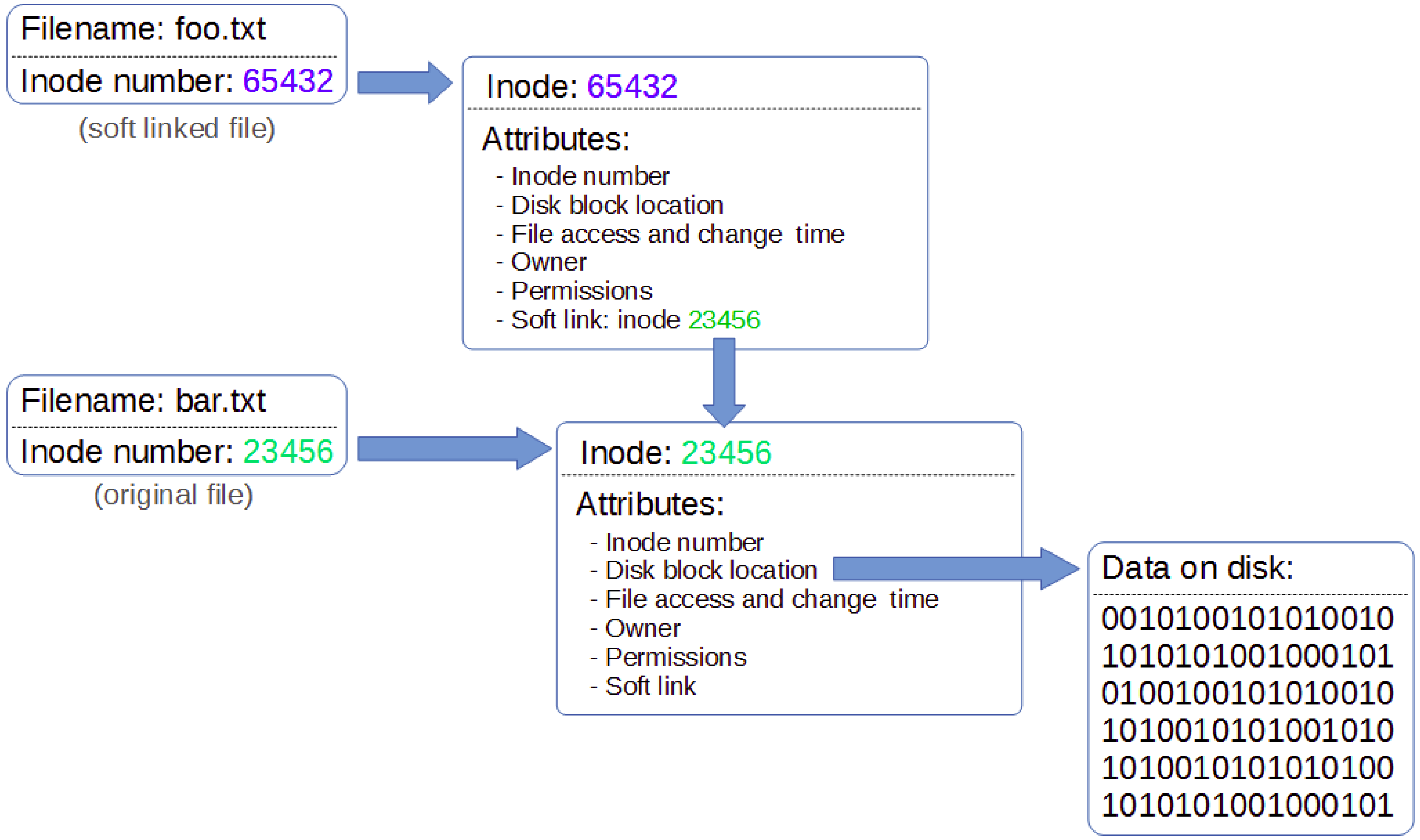
- Directories map names to inodes.
- Inodes exist in Static form on disk.
- Inodes contain file metadata and pointers to file data.
- Inods do not specify pathname.
- Multiple names and hard links

Files System: Hard links



- ln command
- The hard link directly points to the inode of the file.
- Creating a hard link has the effect of giving one file multiple names (e.g. different names in different directories) all of which independently connect to the same data on the disk, none of which depends on any of the others.
- In Unix, an alias is a shell concept and not an OS concept. Not all Unix shells support aliases (like the original Bourne shell).

Files System: Soft links



- ln -s command
- The soft link or symbolic link points to the inode through a file.
- A soft link is a short file that contains the text of a file name, or a location that gives direct access to yet another file name within some directory.

Files System: Dangling soft link

STEPS:

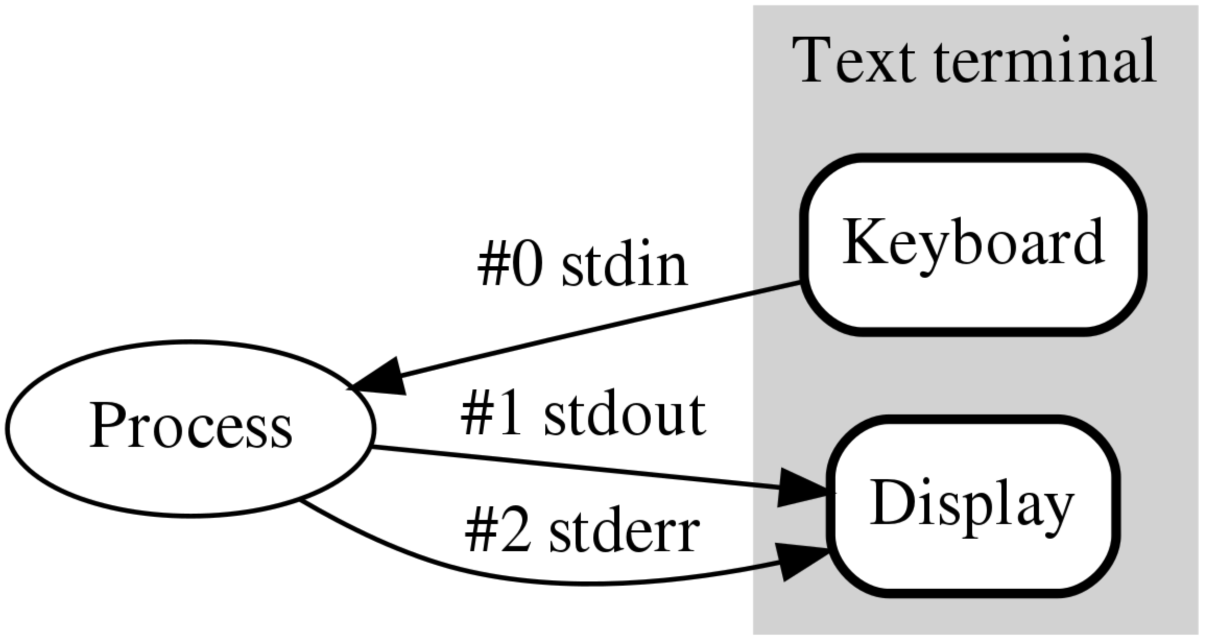
- Creating hard and soft links to foo and bar in workingDir/dir1/dir2
- List and access them

```
rekha@rekha-ThinkPad-P50:~/workingDir/dir1/dir2$ln ../../foo dir2-foo
rekha@rekha-ThinkPad-P50:~/workingDir/dir1/dir2$ln -s ../../bar dir2-bar
rekha@rekha-ThinkPad-P50:~/workingDir/dir1/dir2$ ls -l
total 4
lrwxrwxrwx 1 rekha rekha      9 Aug  6 11:00 dir2-bar -> ../../bar
-rw-rw-r-- 2 rekha test-demo 46 Aug  5 23:18 dir2-foo
```

- Move foo and bar from workingDir to dir1
- See how soft link gets broken and hard link stays intact.

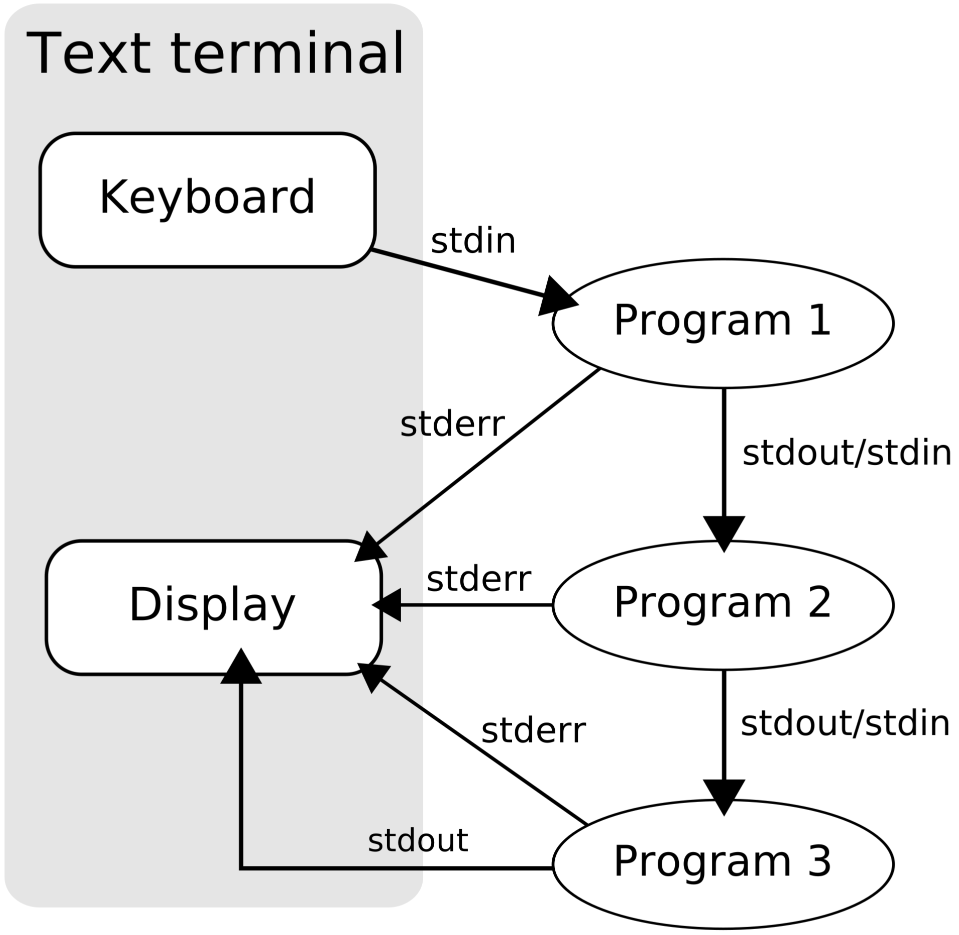
```
rekha@rekha-ThinkPad-P50:~/workingDir$mv foo  dir1
rekha@rekha-ThinkPad-P50/workingDir$mv bar  dir1
rekha@rekha-ThinkPad-P50:~/workingDir$cd dir1/dir2
rekha@rekha-ThinkPad-P50:~/workingDir$ls -l
total 4
lrwxrwxrwx 1 rekha rekha      9 Aug  6 11:00 dir2-bar -> ../../bar
-rw-rw-r-- 2 rekha test-demo 46 Aug  5 23:18 dir2-foo
```

Piping



- A way of unhooking a stream from its default device.
- Changing where input comes from/output goes to.
- The operators:
  - 0 Input redirection: 0< or just <
  - 1 Output redirection: 1> or >, 1>> or >>
  - 2 Error redirection: 2> or 2>>

Piping



- Data flows in only one direction normally.
- Processes have a common ancestor.
- FIFO
- Transient data
- Direct blocks

```
$ ps o pid,ppid,cmd | cat |sort

19164 19157 bash
22799 19164 ps o pid,ppid,cmd
22800 19164 cat
22801 19164 sort
  PID  PPID CMD
```

Piping :demo

- /proc contains directory for each running process in the system
- The set of file descriptors open in a process can be accessed under the path /proc/PID/fd/ where PID is process id.
- Init is the first process in the system with pid = 1. try **sudo ls -l /proc/1/fd | head -3**

```
rekha@rekha-ThinkPad-P50:~/workingDir/dir1/dir2$
[1] 14802
rekha@rekha-ThinkPad-P50:~/workingDir/dir1/dir2$ ps -au | grep sleep
rekha    14801  0.0  0.0   7288   648 pts/2    S    11:54   0:00 sleep 200
rekha    14802  0.0  0.0   7288   656 pts/2    S    11:54   0:00 sleep 400
rekha    14804  0.0  0.0  14224   952 pts/2    S+   11:54   0:00 grep --color=auto sleep
rekha@rekha-ThinkPad-P50:~/workingDir/dir1/dir2$ ls -l /proc/14801/fd
total 0
lrwx------ 1 rekha rekha 64 Aug  6 11:54 0 -> /dev/pts/2
l-wx------ 1 rekha rekha 64 Aug  6 11:54 1 -> pipe:[150696]
lrwx------ 1 rekha rekha 64 Aug  6 11:54 2 -> /dev/pts/2
rekha@rekha-ThinkPad-P50:~/workingDir/dir1/dir2$ ls -l /proc/14802/fd
total 0
lr-x------ 1 rekha rekha 64 Aug  6 11:54 0 -> pipe:[150696]
lrwx------ 1 rekha rekha 64 Aug  6 11:54 1 -> /dev/pts/2
lrwx------ 1 rekha rekha 64 Aug  6 11:54 2 -> /dev/pts/2
```

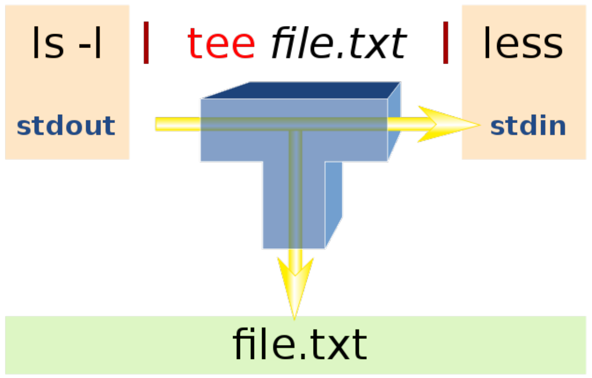
- tty - print the file name of the terminal connected to standard input

```
rekha@rekha-ThinkPad-P50:~/workingDir$ tty
/dev/pts/2
```

Filters: The tee filter

In computing, tee is a command in command-line interpreters (shells) using standard streams which reads standard input and writes it to both standard output and one or more files, effectively duplicating its input. It is primarily used in conjunction with pipes and filters.

```
$sleep 200 | tee file.txt | cat &
$ps -au
rekha@rekha-ThinkPad-P50:~/workingDir$ ls -l /proc/16405/fd
total 0
lrwx------ 1 rekha rekha 64 Aug  6 13:23 0 -> /dev/pts/2
l-wx------ 1 rekha rekha 64 Aug  6 13:23 1 -> pipe:[168737]
lrwx------ 1 rekha rekha 64 Aug  6 13:23 2 -> /dev/pts/2
rekha@rekha-ThinkPad-P50:~/workingDir$ ls -l /proc/16406/fd
total 0
lr-x------ 1 rekha rekha 64 Aug  6 13:23 0 -> pipe:[168737]
l-wx------ 1 rekha rekha 64 Aug  6 13:23 1 -> pipe:[168739]
lrwx------ 1 rekha rekha 64 Aug  6 13:23 2 -> /dev/pts/2
l-wx------ 1 rekha rekha 64 Aug  6 13:23 3 -> /home/rekha/workingDir/file.txt
rekha@rekha-ThinkPad-P50:~/workingDir$ ls -l /proc/16407/fd
total 0
lr-x------ 1 rekha rekha 64 Aug  6 13:23 0 -> pipe:[168739]
lrwx------ 1 rekha rekha 64 Aug  6 13:23 1 -> /dev/pts/2
lrwx------ 1 rekha rekha 64 Aug  6 13:23 2 -> /dev/pts/2
```



Filters: tr filter

tr [options] set1 set2

- Translates/deletes each character in set1 to set2
- a.k.a. search and replace
- Receives input only from stdin. From files?

```
rekha@rekha-ThinkPad-P50:~/workingDir$ cat foo
hello
mango
world
apple
hello
guava
pineapple
```

```
rekha@rekha-ThinkPad-P50:~/workingDir$ cat foo | tr 'ahpl' '1234'
2e44o
m1ngo
wor4d
1334e
2e44o
gu1v1
3ine1334e
```

```
rekha@rekha-ThinkPad-P50:~/workingDir$ echo "hello there" | tr -d ‘e’
hllo thr
```

System administration

- du to find the size occupied by file on disk
- df for seeing free disk space
- free to see the memory usage
- top to monitor running processes
- shutdown, reboot to restart machine
- su newuser switches the current user by launching a new shell as newuser
- passwd to change the password
- mount, umount for accessing other disks/partitions
- apt-get for software package installation/removal
- ps to see processes, kill to kill a process.
- fg to run a process in active mode and bg to run the process in background mode
- wget to download a file from a website
- ping
- host
- finger to get information on user