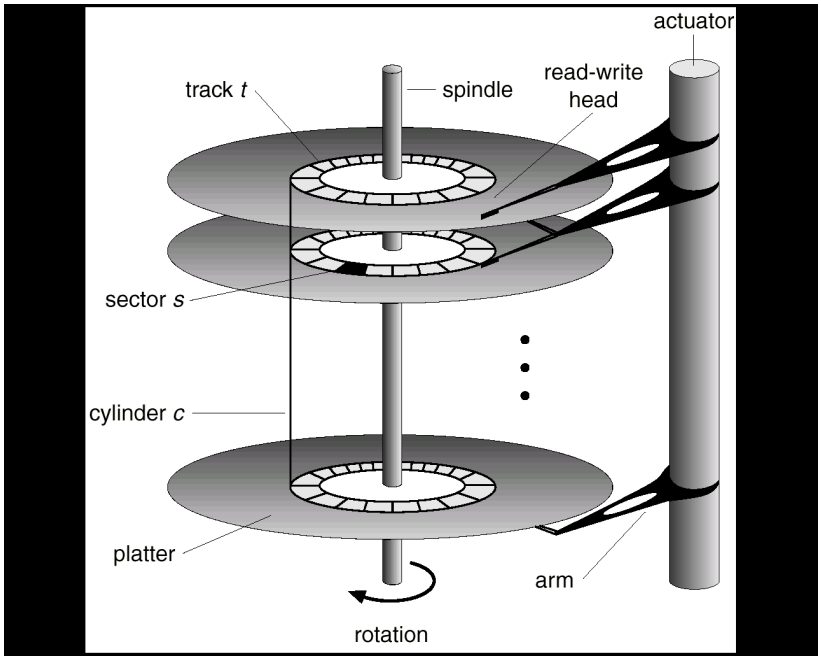
File Operations

1. Disk Geometry



2.Disk Partition

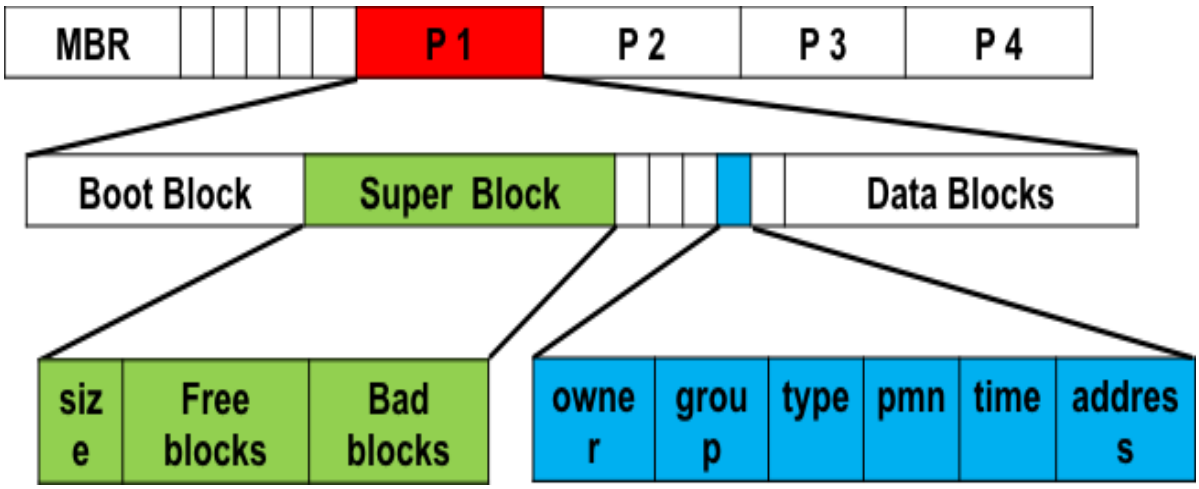
1.Swap Partition
2.Data Partition

Partition Table-

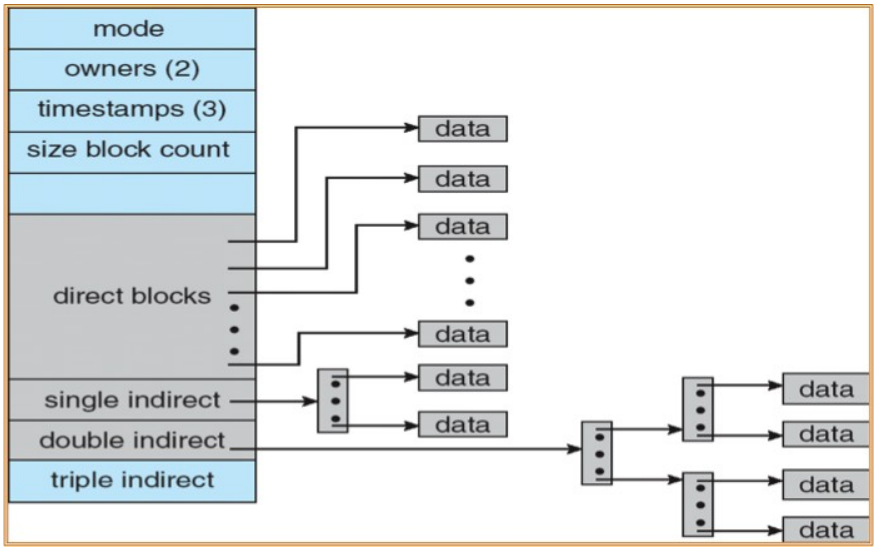
maintained on hdd, it is a structure which manages info about partitons of a hdd

man partx
man fs

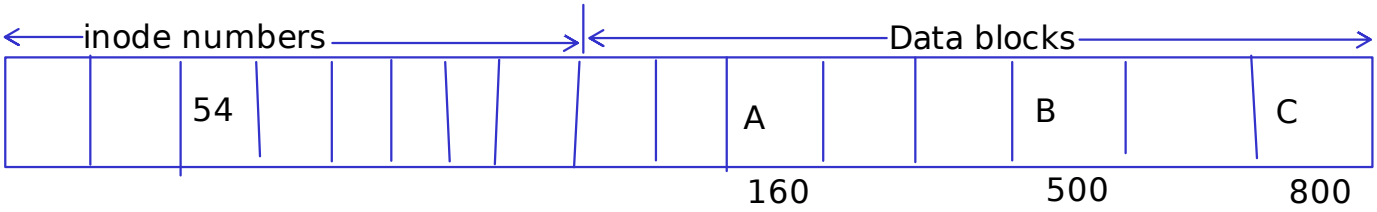
3.File System Architecture



Structure of Unix Inode



$echo "Test file system architecture" 1>/home/chaman/f1.txt



inode block #54

owner

group

Timestamp

File size

Permisions

Direct ptrs

Single I.D ptr
Double I.D ptr
Triple I.D ptr

Data block numbers

/home/chaman/

45	.
8	..
54	f1.txt

$cat /home/chaman/f1.txt

root

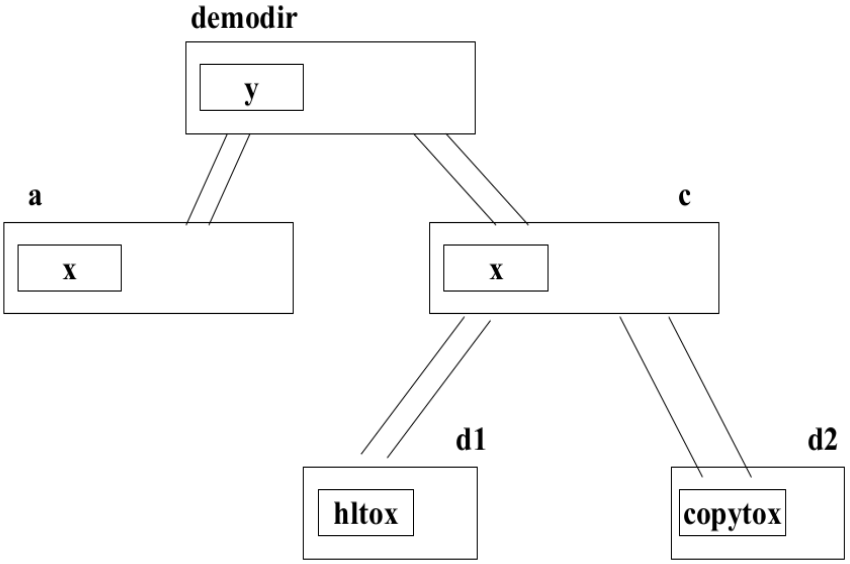
1	.
1	..
8	home

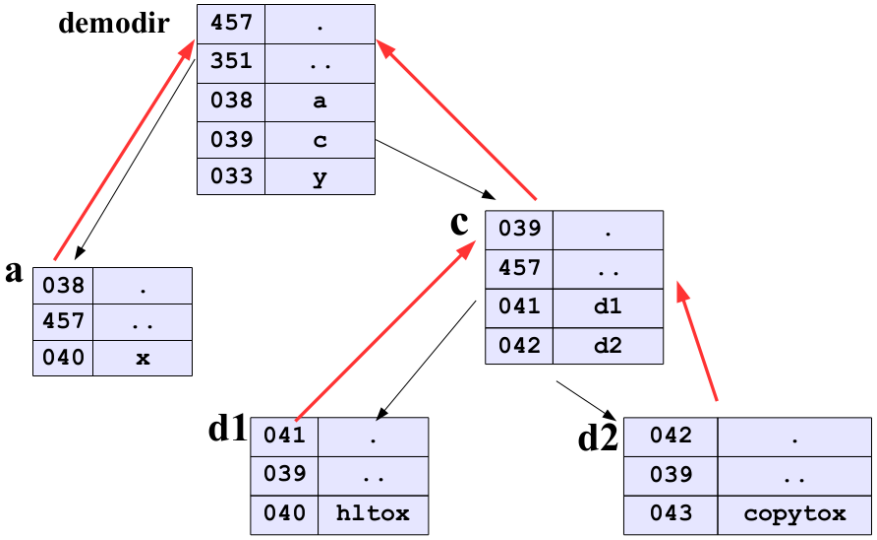
home

8	.
1	..
45	chaman

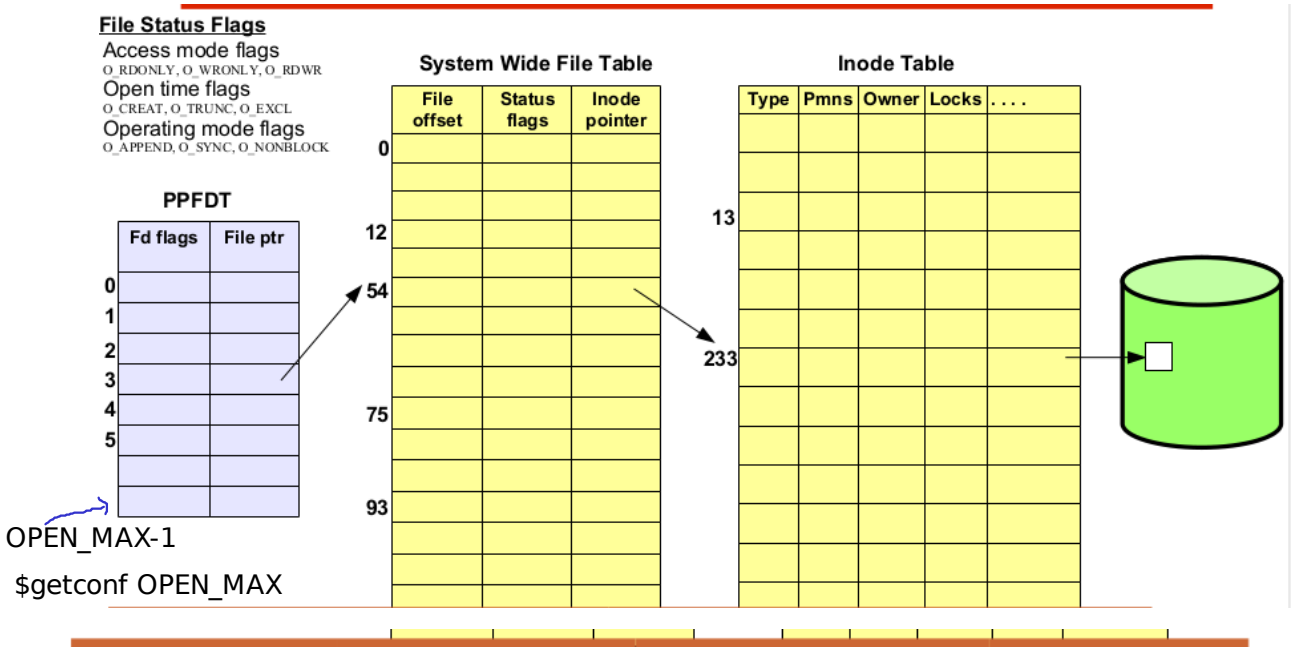
chaman

45	.
8	..
54	f1.txt


demodir
y
a
x
c
x
d1
hltox
d2
copytox

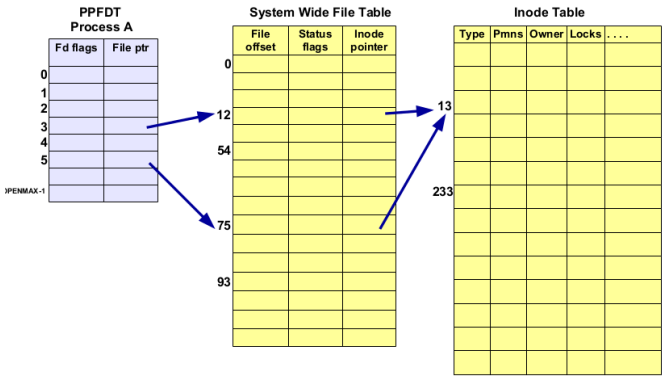

demodir
457 .
351 ..
038 a
039 c
033 y
a
038 .
457 ..
040 x
c
039 .
457 ..
041 d1
042 d2
d1
041 .
039 ..
040 hltox
d2
042 .
039 ..
043 copytox

File Descriptor to File Contents

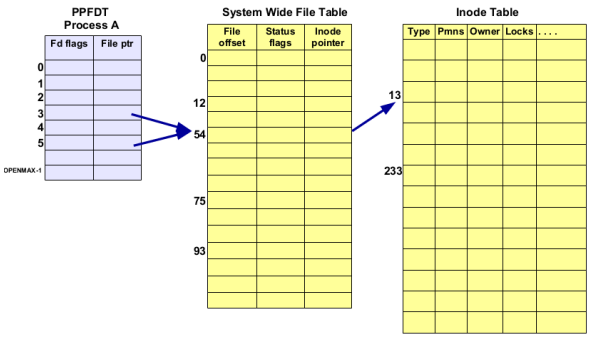


File Descriptor	Purpose	POSIX Name	stdio Stream
0	Standard input	STDIN_FILENO	stdin
1	Standard output	STDOUT_FILENO	stdout
2	Standard error	STDERR_FILENO	stderr

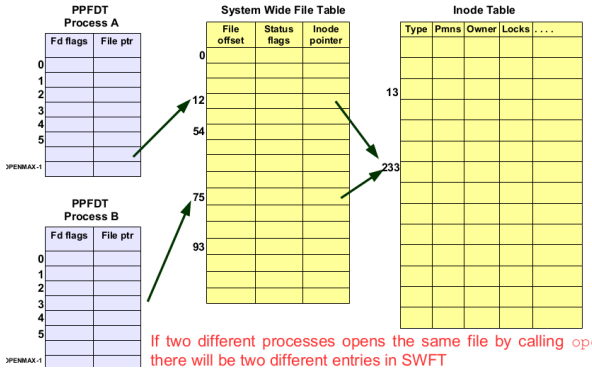
Relationship between fd and Open files



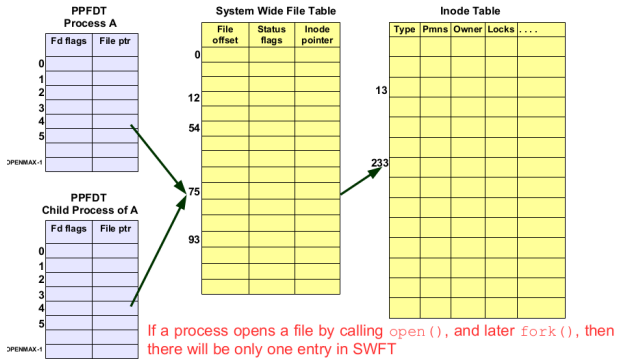
A process can open a file twice. If this is done by calling `open()` twice, then there will be two different entries in PPFTD as well as in SWFT for that single file



A process can open a file twice. If this is done by calling `dup()`, then there will be two entries in PPFDT but only one entry in SWFT



If two different processes opens the same file by calling `open()`, there will be two different entries in SWFT



If a process opens a file by calling `open()`, and later `fork()`, then there will be only one entry in SWFT

FILE TYPES IN UNIX

FILE TYPES IN UNIX

1.Regular file (-) Files that contain information entered in them by a user, an application program or a system utility program.

2.Directory (d) Contains a list of file names plus pointers to associated i-nodes. Directories are actually ordinary files with special write protection privileges so only the file system can write into them, while read access is available to user programs.

3.Symbolic Link (l) Links let you give a file more than one name.

4.Block Special File (b) A block special file consists of as sequence of numbered blocks. The key property of the block special file is that each block can be individually addressed and accessed, i.e. we can directly access block 154 without first having to read blocks 0 to 153. Block special files are typically used for disks. E.g. /dev/hda1, /dev/lp.

5.Character Special File (c) Used to communicate with h/w that input or output one character at a time. Keyboard, printers, mice, plotters, networks are examples of character special files.

6.Named Pipe (p) A file that passes data between processes. It stores no data itself, but passes data between process writing data into pipe and process reading from pipe. ls –l /dev | less

7.Socket (s) A stylized mechanism for inter-process communications