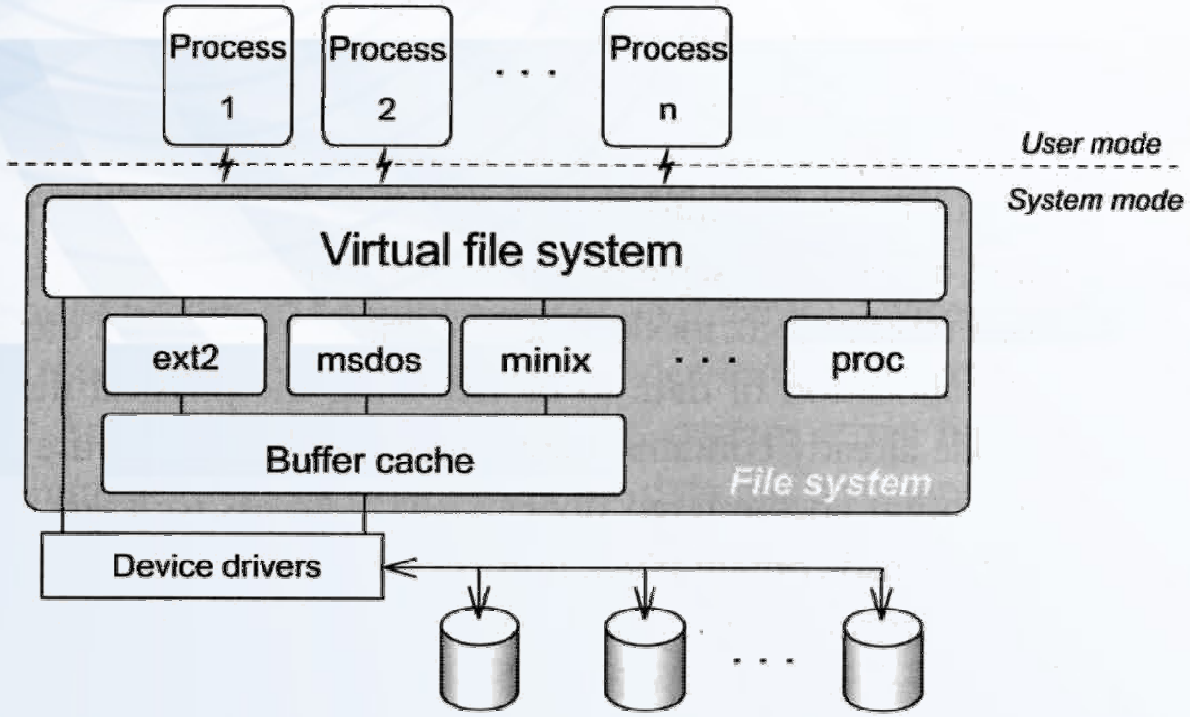
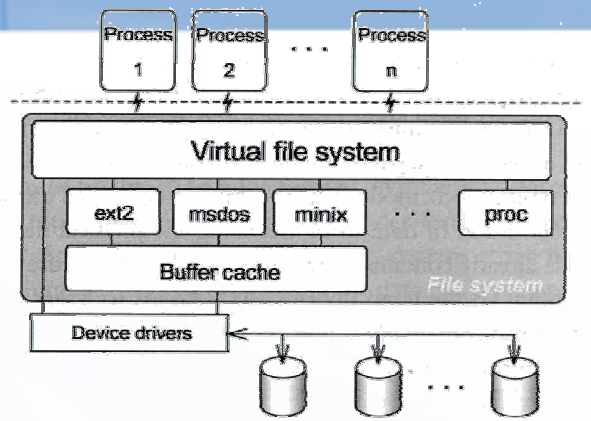
Chapter 5
The Filesystem

Application ⟺ Kernel ⟺ hardware

- Applications call system-calls to request service
- Kernel invokes corresponding drivers to fulfill this service



The basic purpose of filesystem



- Represent and organize the system's storage
  - Four main components:
    - **Namespace**
      - A way of naming things and arranging them in a hierarchy
    - **API**
      - A set of system calls for navigating and manipulating nodes
    - **Security model**
      - A scheme for protecting, hiding and sharing things
    - **Implementation**
      - Code that ties the logical model to an actual disk

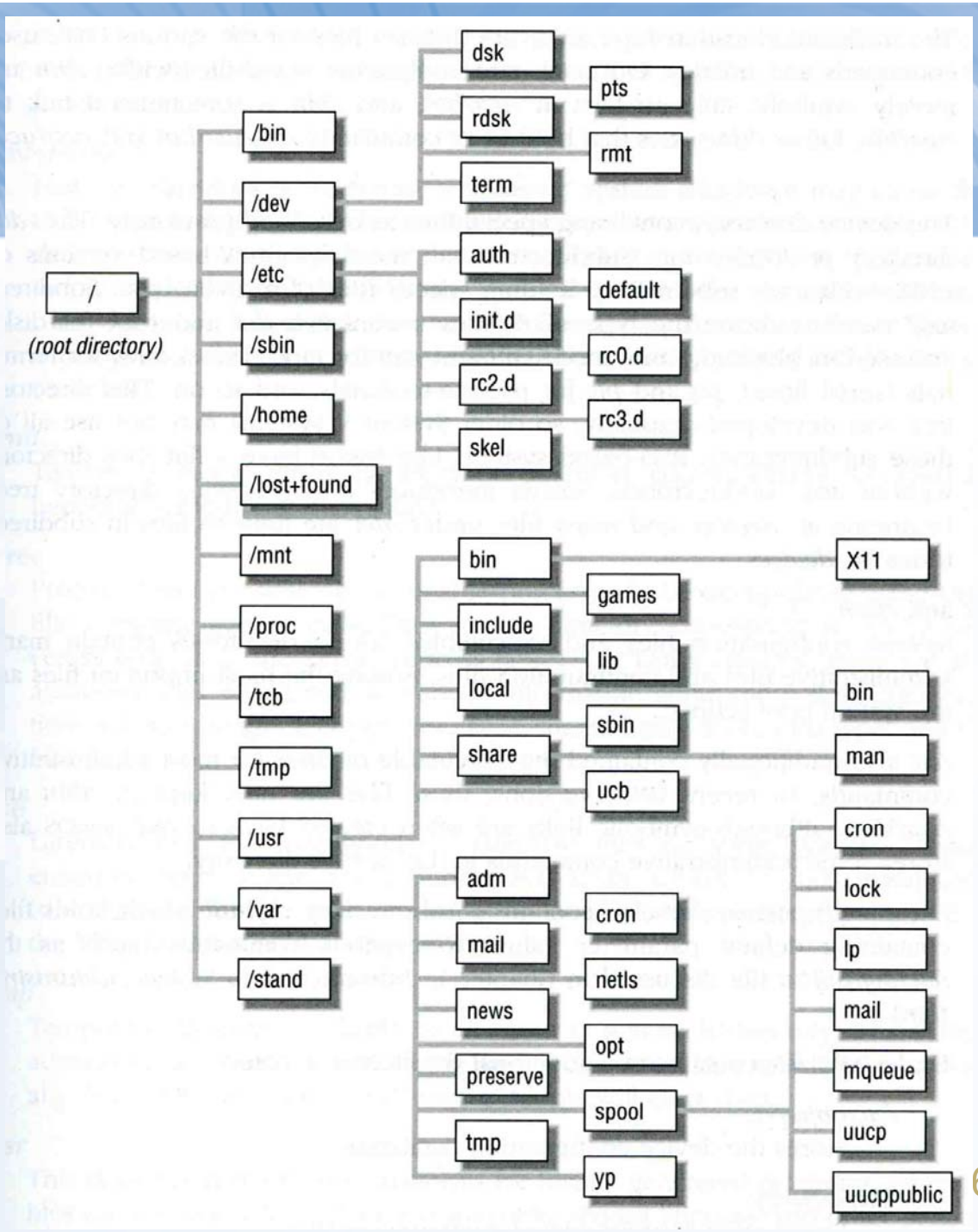
Take a ride of filesystem

- What you can find in a filesystem:
  - Files and directories
  - Hardware device files
  - Processes information
  - Interprocess communication channel
  - Shared memory segments
- We can use common filesystem interface to access such “object”
  - open、read、write、close、seek、ioctl…

pathname

- Two kinds
  - Absolute path ➔ start from /
    - **Such as /u/gcp/92/9217810/test/hehe.c**
  - Relative path ➔ start from your current directory
    - **Such as test/hehe.c**
- Pathname constrains
  - Single component: ≦ 255 characters
  - Single absolute path: ≦ 1023 characters

The organization of the file tree



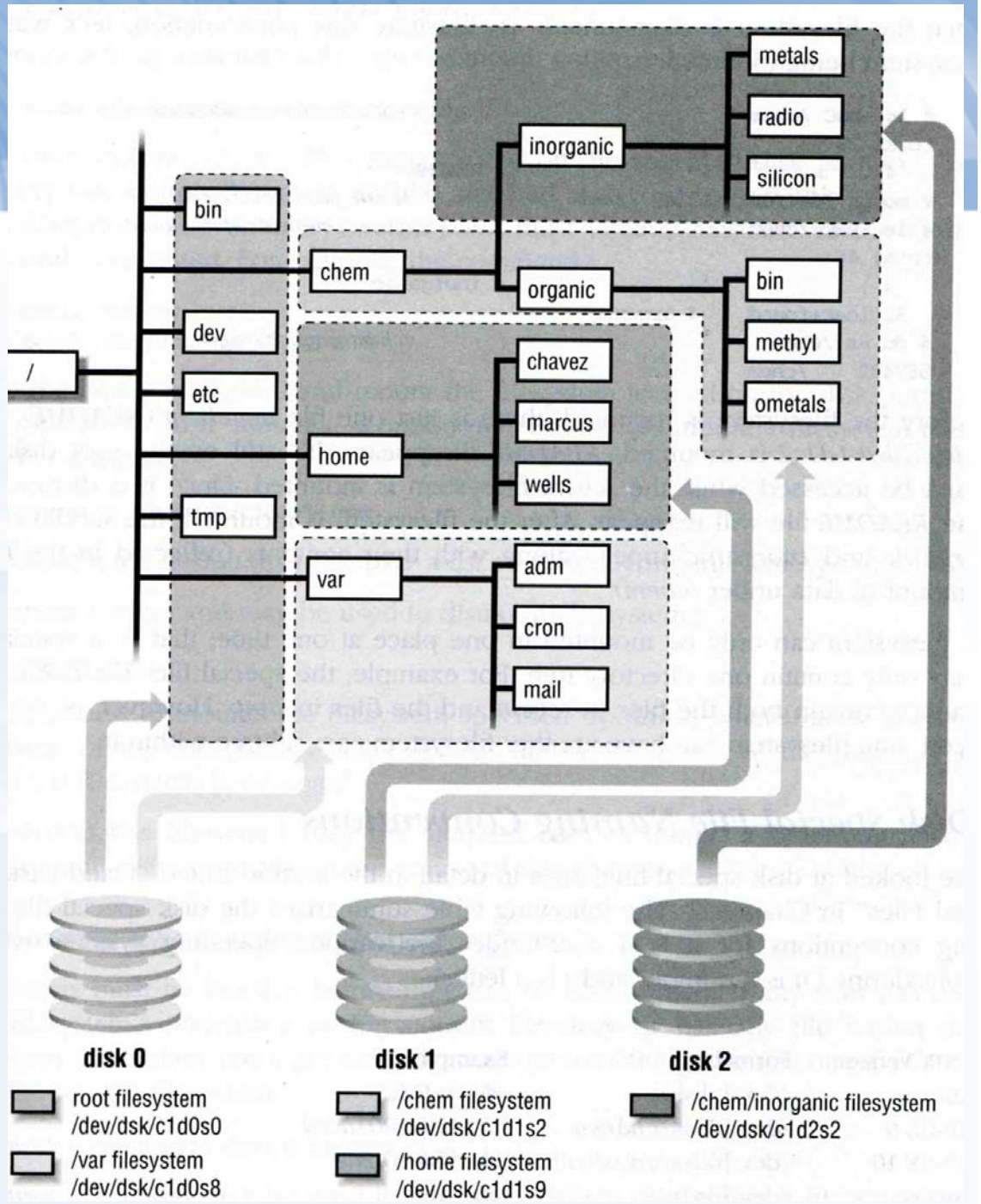
The organization of the file tree – standard directories and their contents

pathname	Contents
/	The root directory
/bin or /sbin	Commands needed for minimal system operability
/usr/bin	Executable files
/usr/local/bin	Local executable
/usr/local/sbin	Local system maintenance commands
/etc	Critical startup and configuration files
/usr/local/etc	Local system configuration files
/dev	Device entries for disks, terminals, modems, etc
/proc	Images of all running process
/usr/lib	Support libraries for standard UNIX programs
/usr/include	Libraries Header files
/var/log	Various system log files
/var/spool	Spooling directories for printers, mails, etc

Mounting file system

- The filesystem in composed of chunks
  - Most are disk partitions
  - Network file servers
  - Memory disk emulators
  - Kernel components
  - Etc,…
- mount command
  - Map the mount point of the existing file tree to the root of the newly attached filesystem
  - % mount /dev/ad2s1e /home2
  - The previous contents of the mount point become inaccessible

Example


/
bin
dev
etc
tmp
chem
home
var
inorganic
organic
chavez
marcus
wells
adm
cron
mail
metals
radio
silicon
bin
methyl
ometals
disk 0
disk 1
disk 2
root filesystem
/dev/dsk/c1d0s0
/var filesystem
/dev/dsk/c1d0s8
/chem filesystem
/dev/dsk/c1d1s2
/home filesystem
/dev/dsk/c1d1s9
/chem/inorganic filesystem
/dev/dsk/c1d2s2

fstab – filesystem table

- Automatically mounted at boot time
  - /etc/fstab
    - **Filesystem in this file will be checked and mounted automatically at boot time**

ccbsd2's /etc/fstab

# Device	Mountpoint	FStype	Options	Dump	Pass#
/dev/ad0s1b	none	swap	sw	0	0
/dev/ad0s1a	/	ufs	rw	1	1
/dev/acd0c	/cdrom	cd9660	ro,noauto	0	0
proc	/proc	procfs	rw	0	0
ccduty:/bsdhome	/bsdhome	nfs	rw,noauto	0	0

Unmounting file system

- umount command
  - % umount node
    - **Ex: umount /home2**
- Busy filesystem
  - Someone's current directory is there or there is opened file
  - Use "umount –f"
  - We can use "lsof" or "fstat" like utilities to figure out who makes it busy

lsof, fuser and fstat commands

> lsof (/usr/ports/sysutils/lsof)

```
tytsai@tybsd:~> lsof /home
COMMAND  PID   USER   FD   TYPE    DEVICE      SIZE/OFF   NODE NAME
tcsh     125   tytsai  cwd  VDIR  116,196612    512       425600 /home/tytsai
ssh      292   tytsai  cwd  VDIR  116,196612    512       425600 /home/tytsai
tcsh     319   tytsai  cwd  VDIR  116,196612    512       425600 /home/tytsai
lsof     1259  tytsai  cwd  VDIR  116,196612    512       425600 /home/tytsai
lsof     1260  tytsai  cwd  VDIR  116,196612    512       425600 /home/tytsai
```

> fstat (FreeBSD)

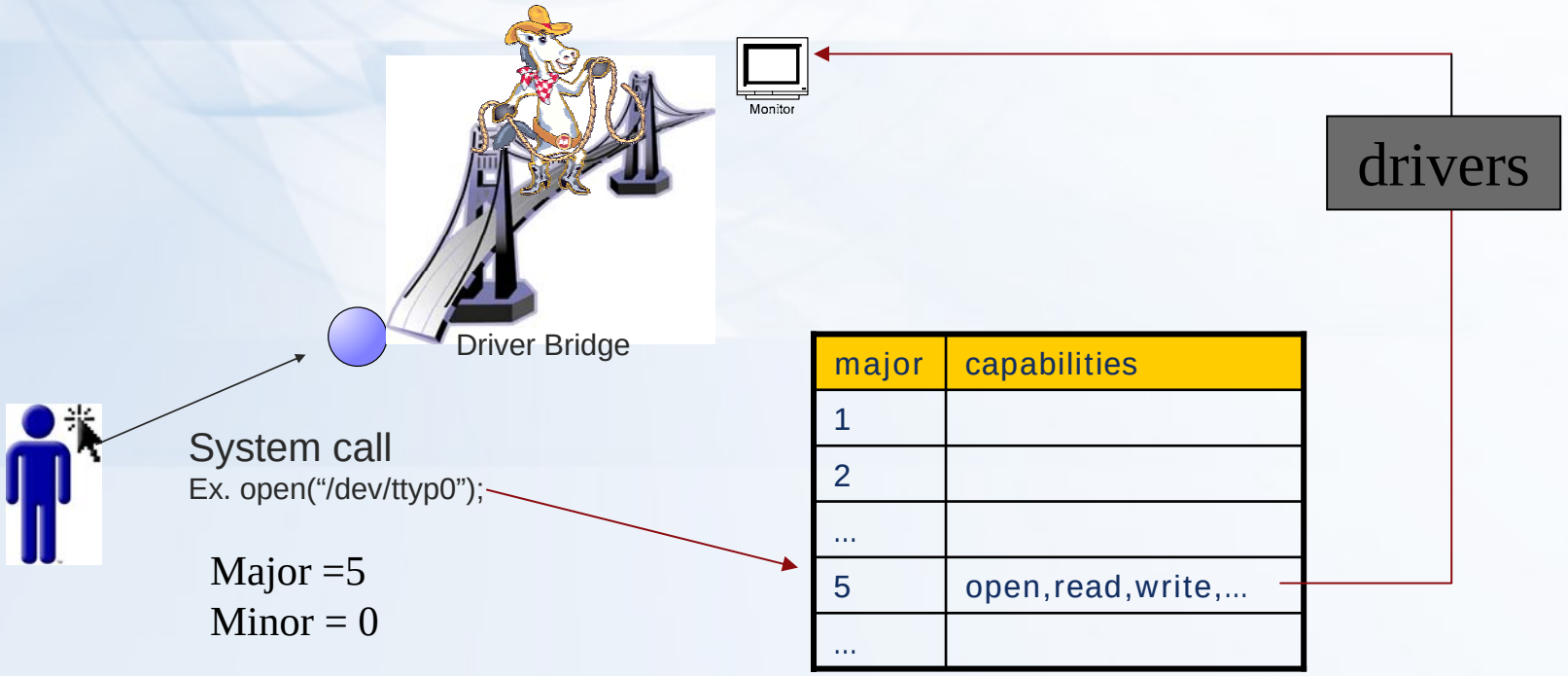
```
tytsai@tybsd:~> fstat -f /home
USER     CMD        PID   FD   MOUNT      INUM   MODE        SZ|DV R/W
tytsai   fstat      1249   wd   /home    425600   drwxr-xr-x   512    r
tytsai   tcsh       319   wd   /home    425600   drwxr-xr-x   512    r
tytsai   ssh        292   wd   /home    425600   drwxr-xr-x   512    r
tytsai   tcsh       125   wd   /home    425600   drwxr-xr-x   512    r
```

Type of files

- Regular files
- Directories
  - Include “.” and “..”
- Character and Block device files
- UNIX domain sockets
- Named pipes
- Symbolic links

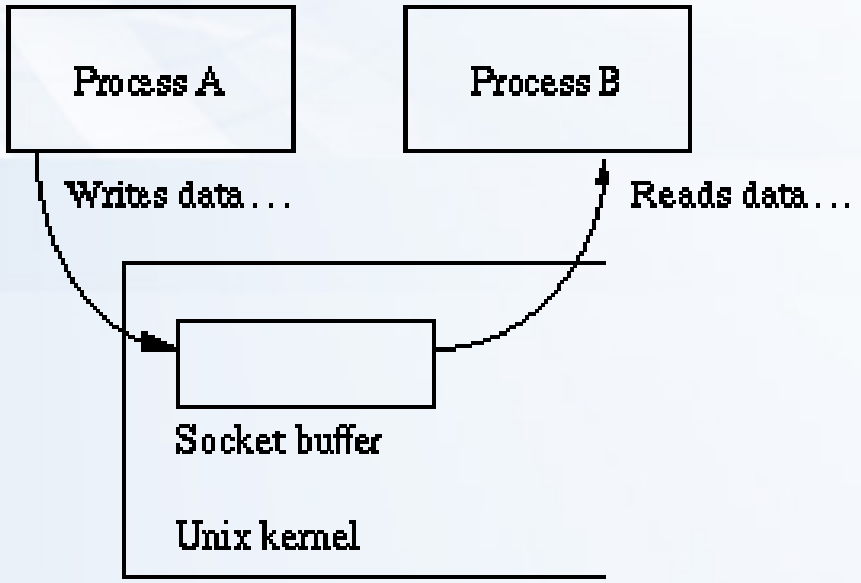
Type of files – character and block device files

- Use “mknod” to build special file
  - % mknod name [c|b] major minor
    - **The same major number will use the same driver**



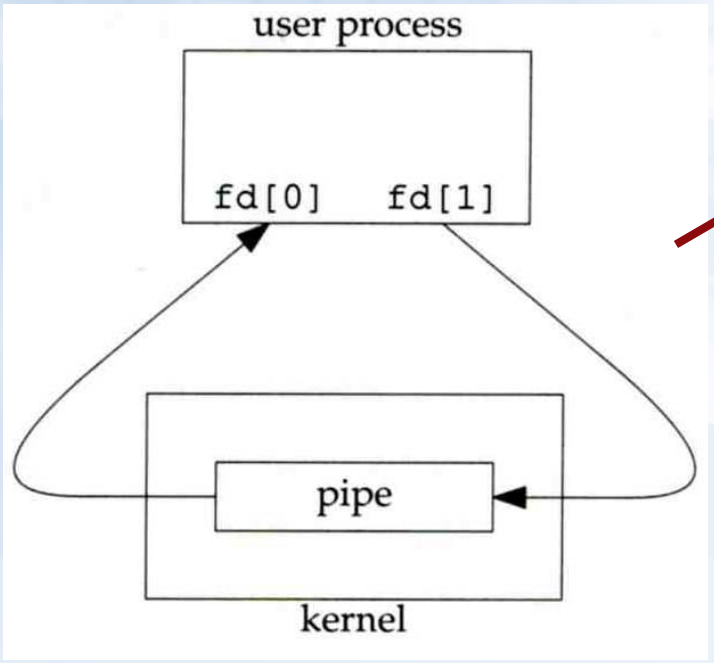
Type of files – UNIX domain sockets

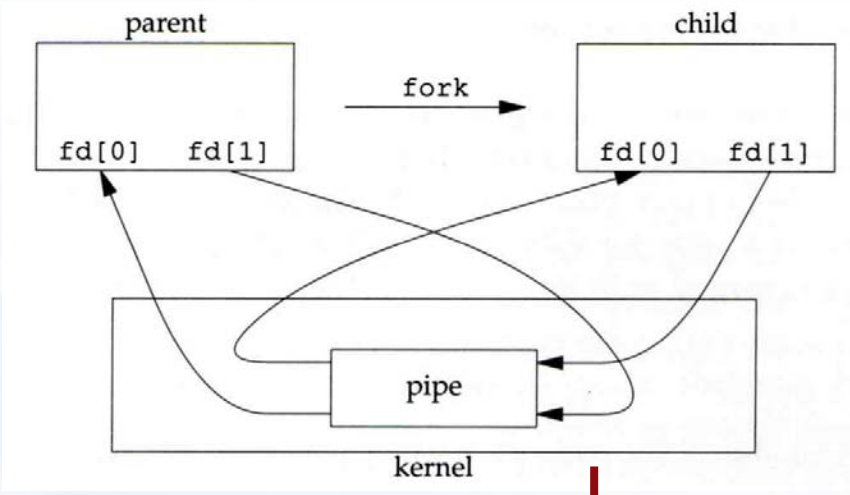
- UNIX domain socket
  - Created by socket()
  - Local to a particular host
  - Be referenced through a filesystem object rather than a network port

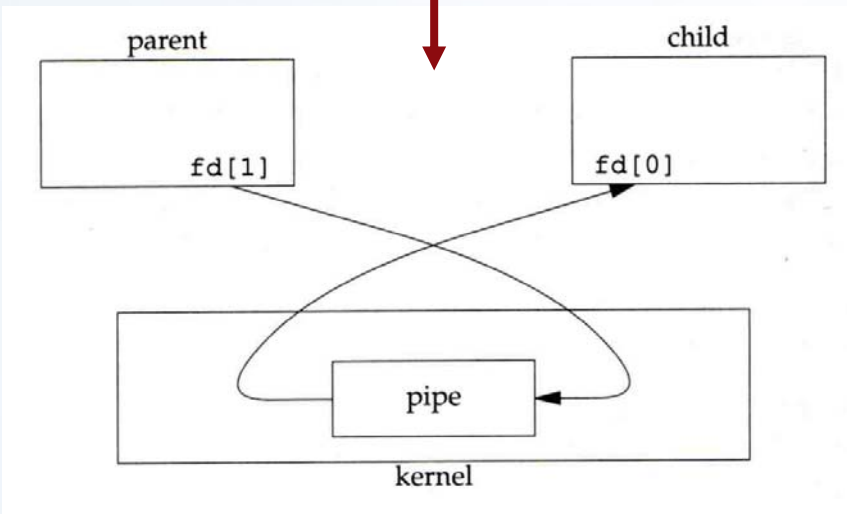


Type of files – Named pipes

- Let two processes do “FIFO” communication





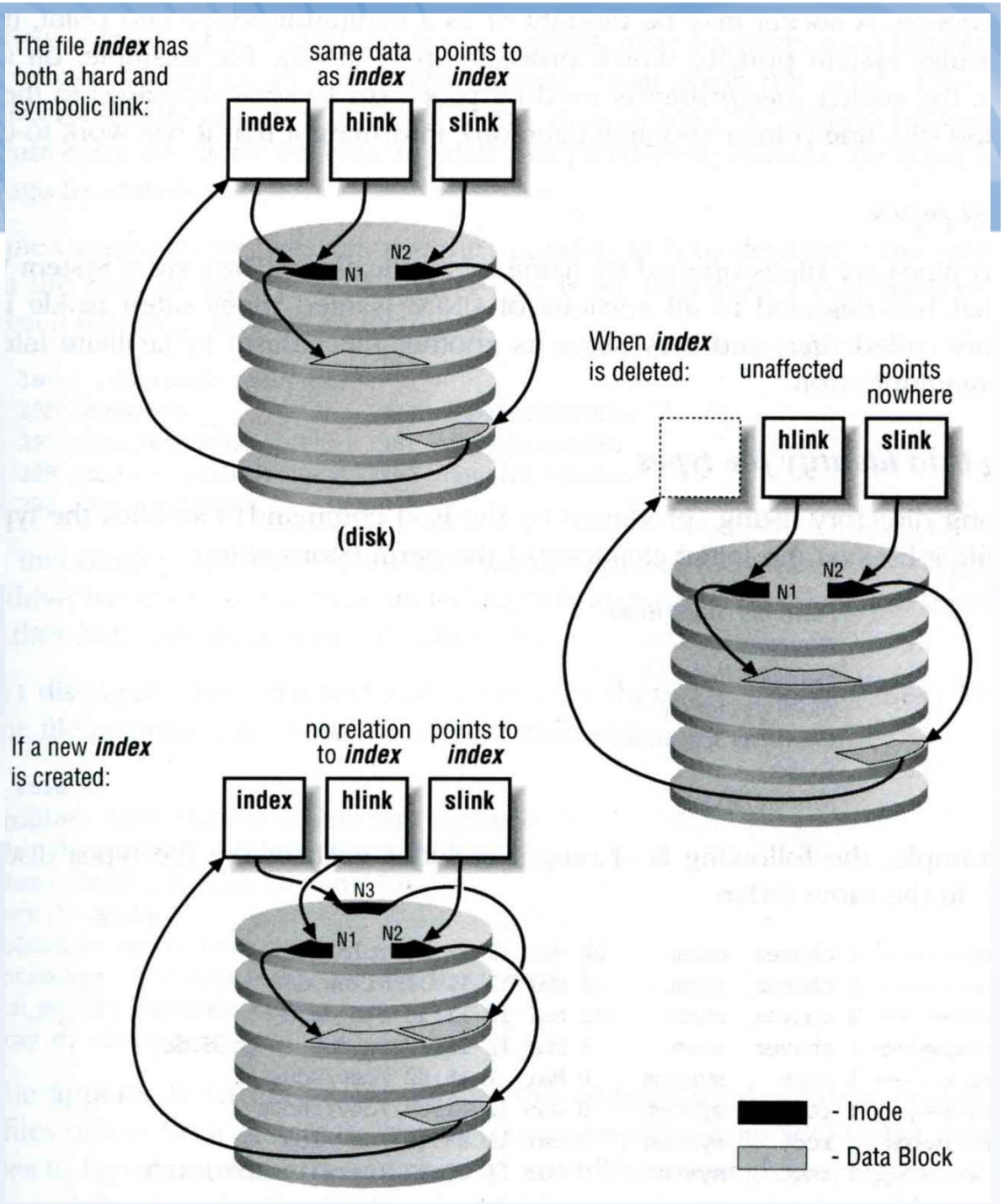


Type of files – symbolic link (1)

- Link
  - Hard link
    - **associate two or more filenames with the same inode**
    - **% ln ori-file hard-file**
  - Soft (symbolic) link
    - **A file which points to another pathname**
    - **% ln –s ori-file soft-file**

Type of files – symbolic link (2)

- Ex:
  - % touch index
  - % ln index hlink
  - % ln –s index slink



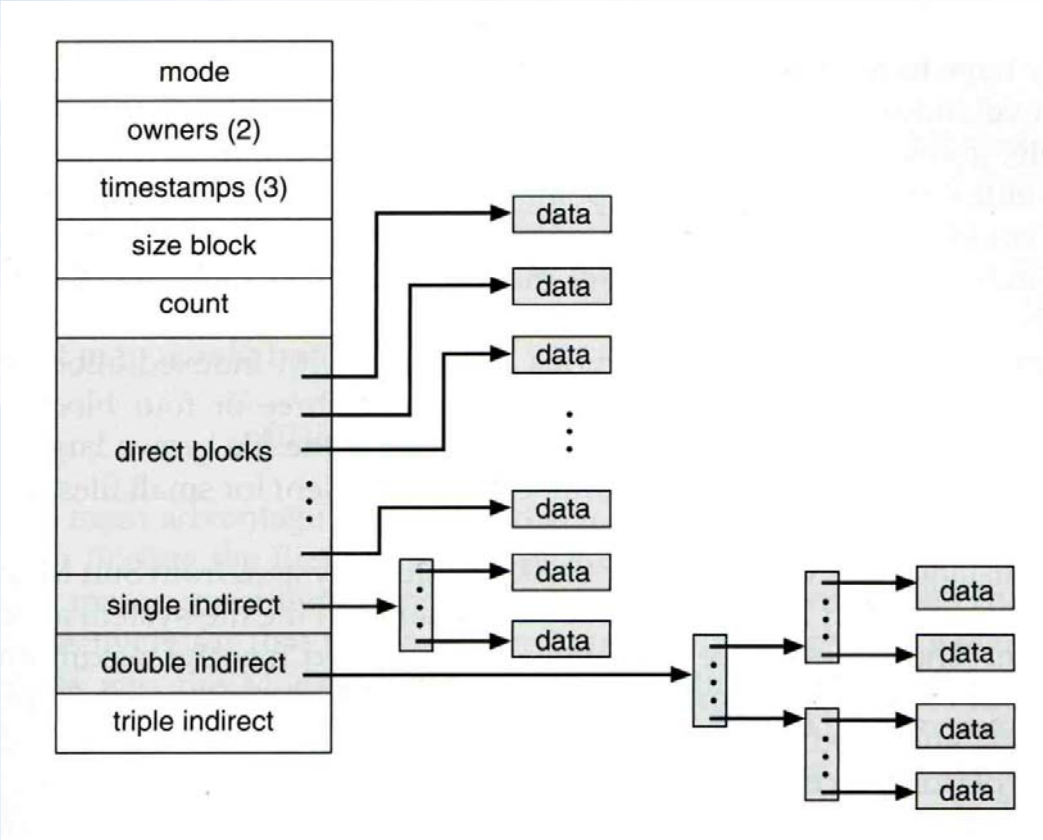
File type encoding used by ls

File type	Symbol	Created by	Removed by
Regular file	-	editors, cp, etc	rm
Directory	d	mkdir	rmdir
Character device file	c	mknod	rm
Block device file	b	mknod	rm
UNIX domain socket	s	socket(2)	rm
Named pipe	p	mknod	rm
Symbolic link	l	ln -s	rm

```
tytsai@tybsd:/var/run> ls -al
total 98
drwxr-xr-x   4       root  wheel     512  Oct  4 08:50 ./
drwxr-xr-x  20       root  wheel     512  Sep 18 22:37 ../
srw-rw-rw-   1       root  wheel       0  Oct  4 08:50 log=
```

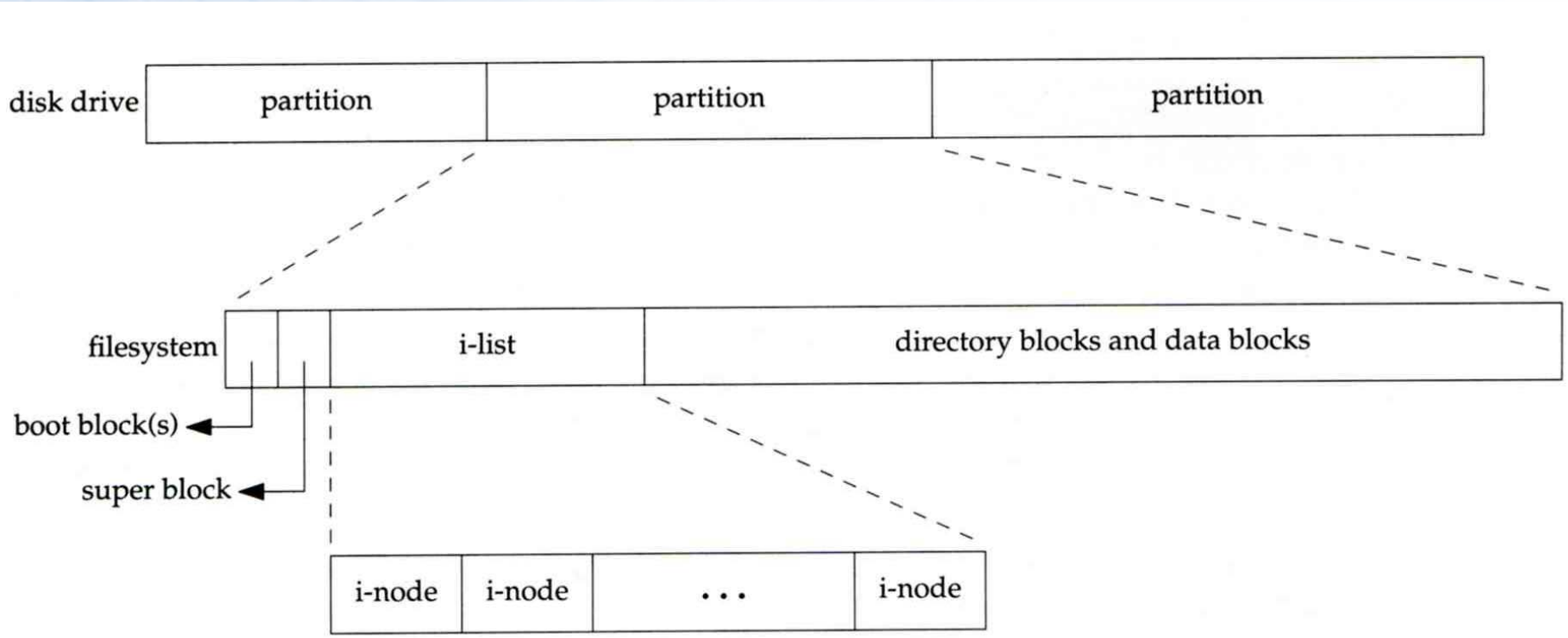
inode

- A structure that records information of a file
  - You can use “ls –il” to see each file’s inode number



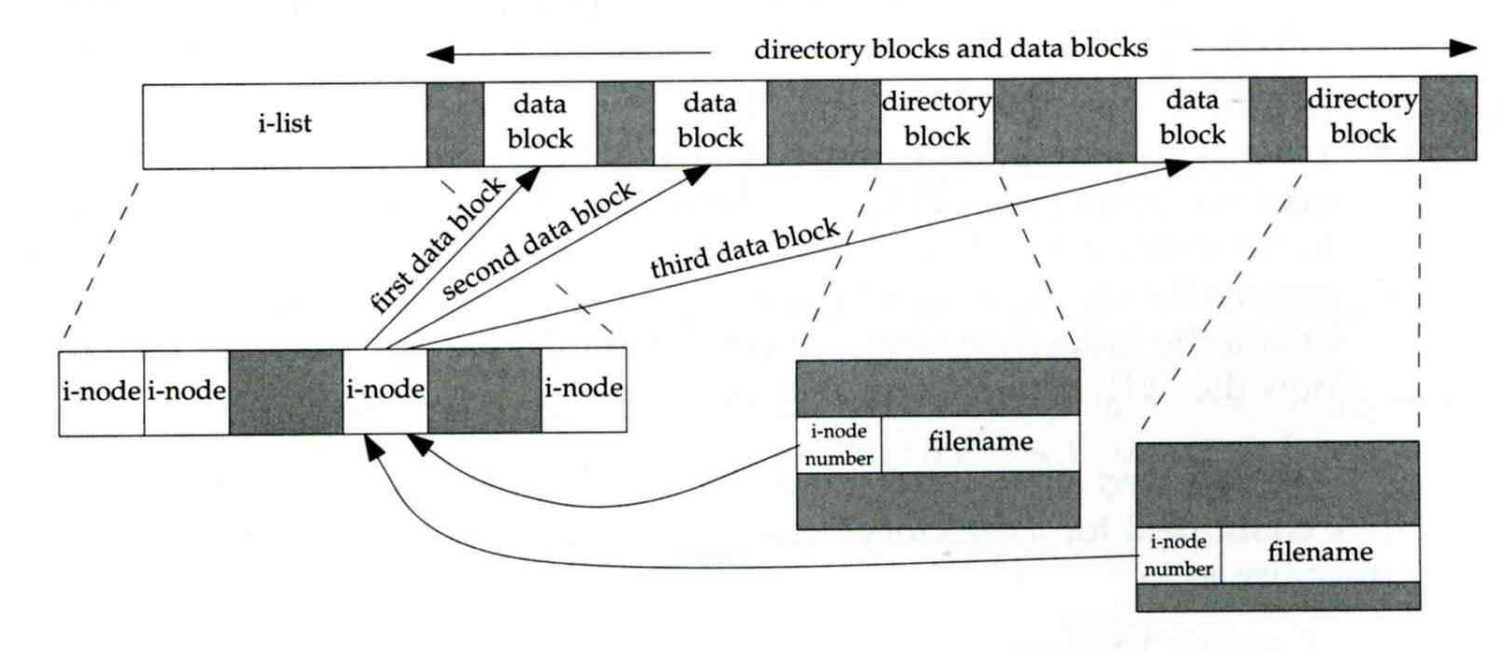
Disk, inode and file (1)

- Filesystem
  - Inode list
  - Data block



Disk, inode and file (2)

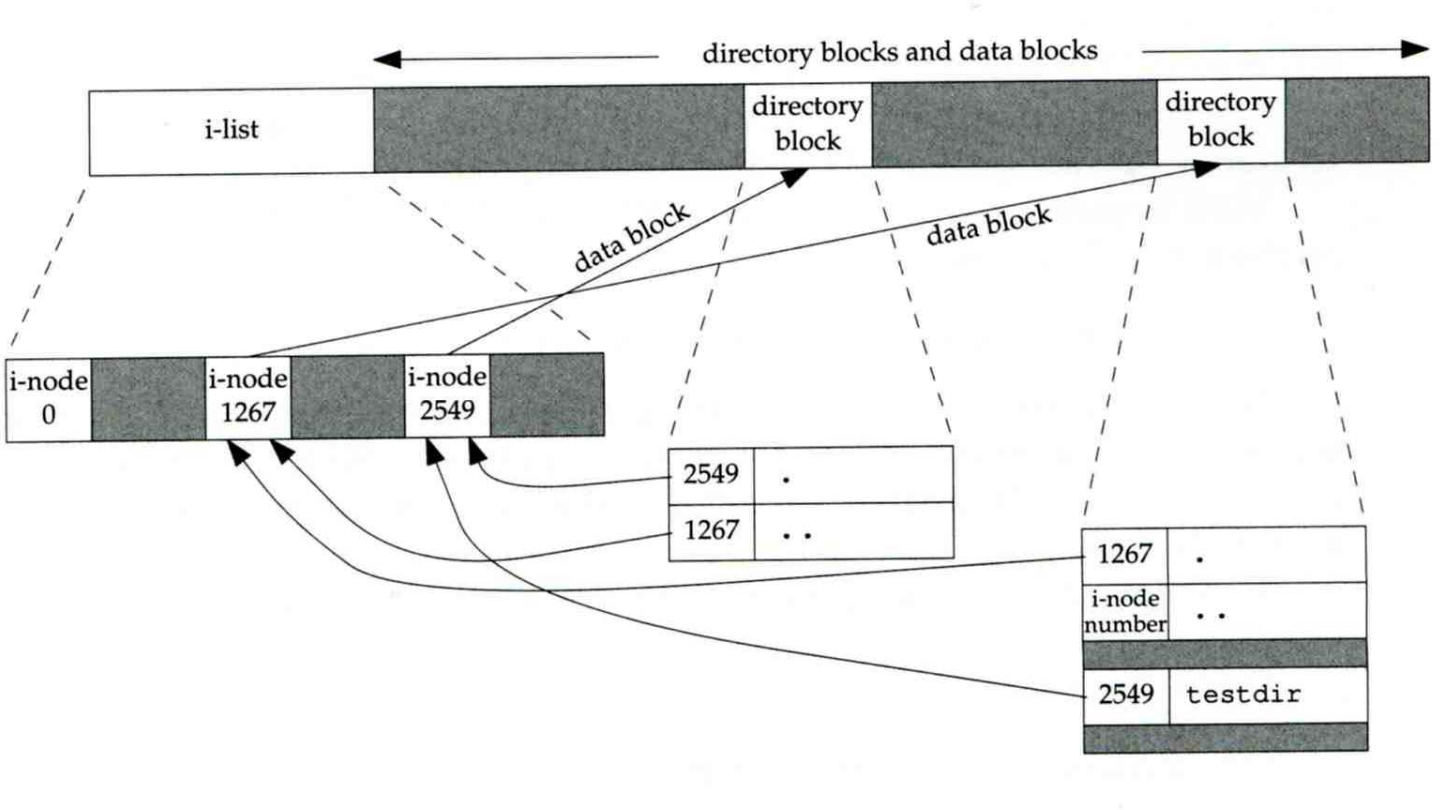
> More detail of inode and data block



Disk, inode and file (3)

> Example

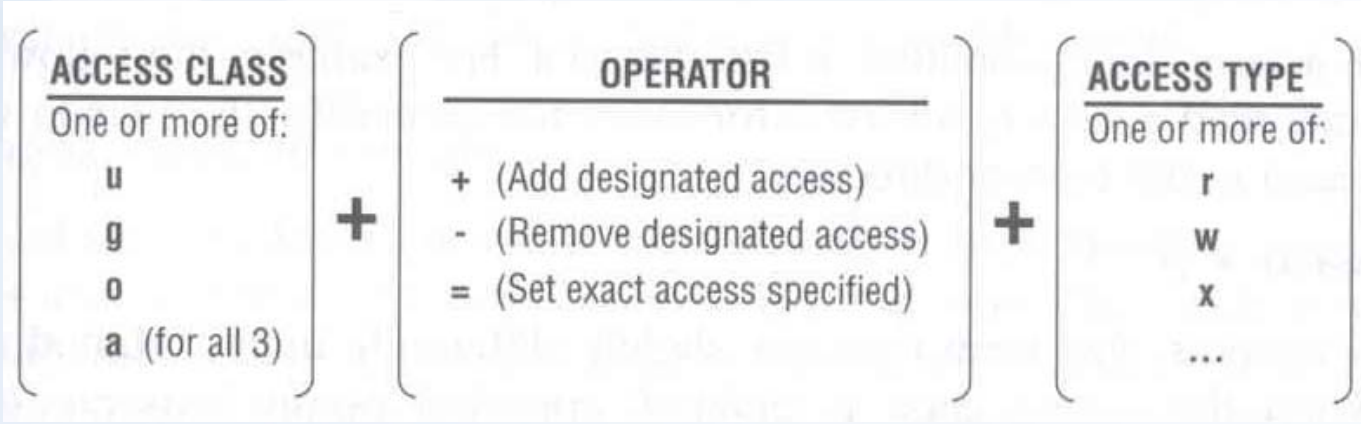
- .
- ..
- testdir



/home/tytsai/testdir

File Access Mode (1)

- rwx r-x r-x
  - User, group, other privileges
- chmod command
  - % **chmod** *access-string* *file*
    - % chmod u+x test.sh
    - % chmod go-w .tcshrc
    - % chmod u+w,r-w hehe haha
    - % chmod –R 755 public_html/



File Access Mode (2)

- setuid, setgid, sticky bit
  - setuid, setgid on file
    - **The effective uid/gid of resulting process will be set to the UID/GID of the file**
    - **setuid**
      - passwd, chsh, crontab
    - **setgid**
      - top, fstat, write
  - setgid on directory
    - **Cause newly created files within the directory to be the same group as directory**
  - sticky on directory
    - **Do not allow to delete or rename a file unless you are**
      - The owner of the file
      - The owner of the directory
      - root

File Access Mode (3)

- Decimal argument of chmod
  - setuid: 4000
  - setgid: 2000
  - stiky : 1000

Mode	Attribute	Mode	Attribute
755	- rwx r-x r-x	644	- rw- r-- r--
4755	- rws r-x r-x	600	- rw- --- ---
2755	- rwx r-s r-x	400	- r-- r-- r--
2775	d rwx rws r-x	1777	d rwx rwx rwt
755	d rwx r-x r-x	4555	- r-s r-x r-x
750	d rwx r-x ---	711	- rwx --x --x
700	d rwx --- ---	711	d rwx --x --x

chown and chgrp

- Change the file ownership and group ownership
  - % chown –R tytsai /home/tytsai
  - % chgrp –R csie /home/tytsai
  - % chown –R tytsai:csie /home/tytsai

FreeBSD bonus flags

- chflags command
  - schg — system immutable flag — (root only)
  - sunlnk — system undeletable flag — (root only)
  - sappnd — system append-only flag — (root only)
  - uappend — user append-only flag — (root, user)
  - uunlnk — user undeletable flag — (root, user)
  - …
- /kernel
  - ls –o
  - chflags noschg /kernel ← unlock

```
tytsai@tybsd:/> ls -lo /
-r-xr-xr-x   1 root  wheel  schg      2839142 Sep 20 14:04 kernel
```