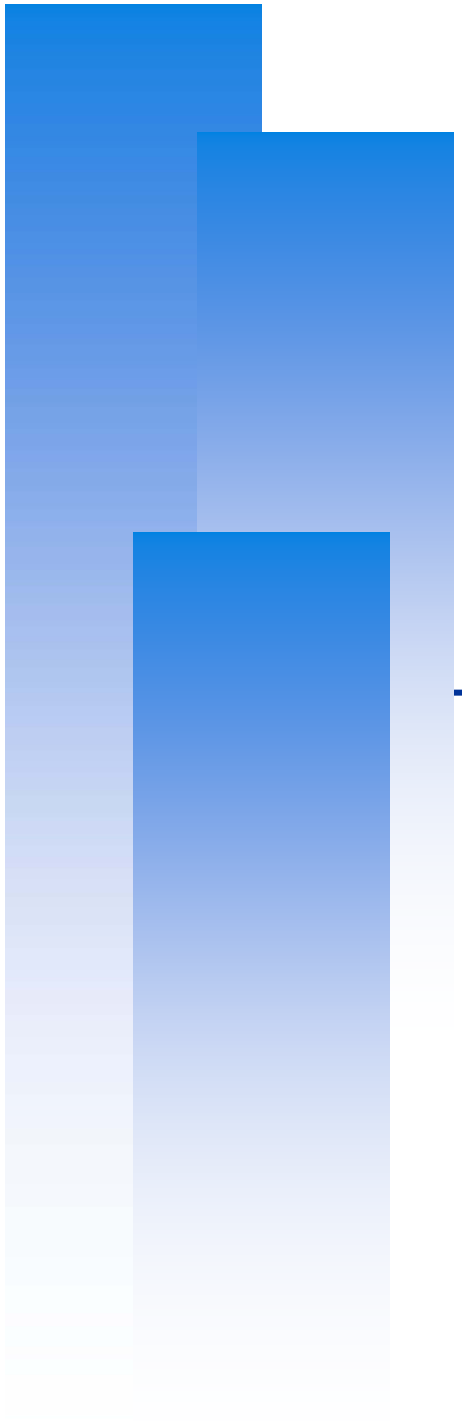


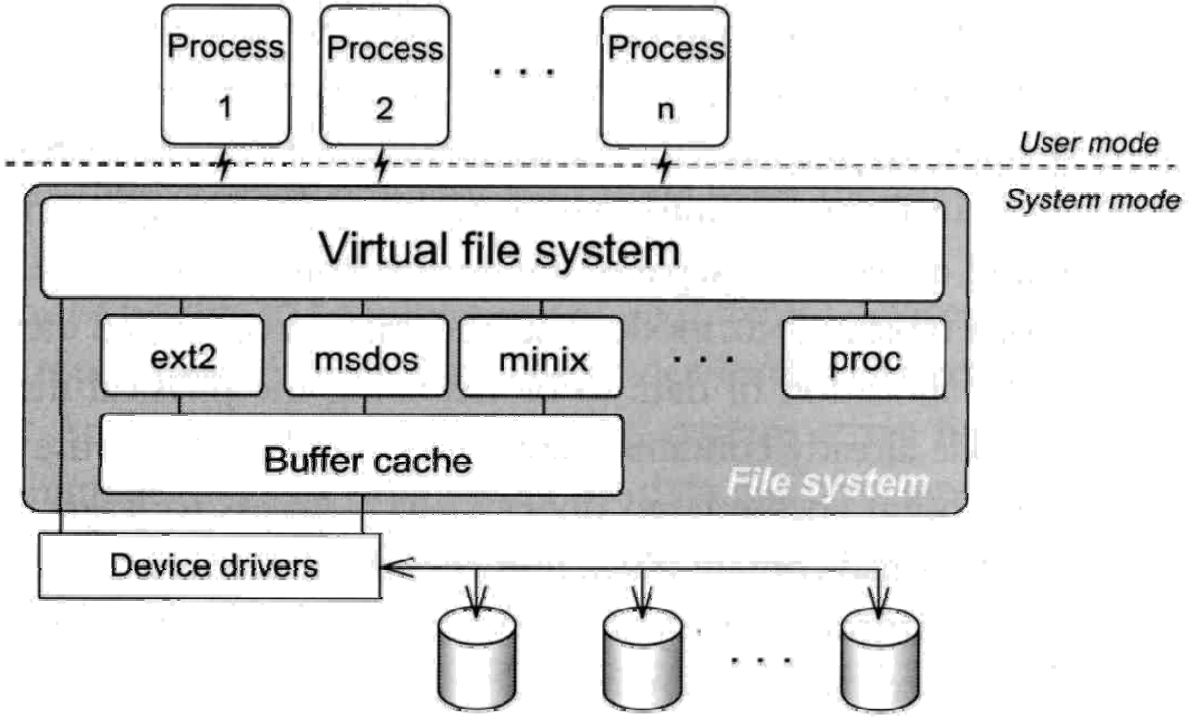
Chapter 5
The Filesystem

Outline

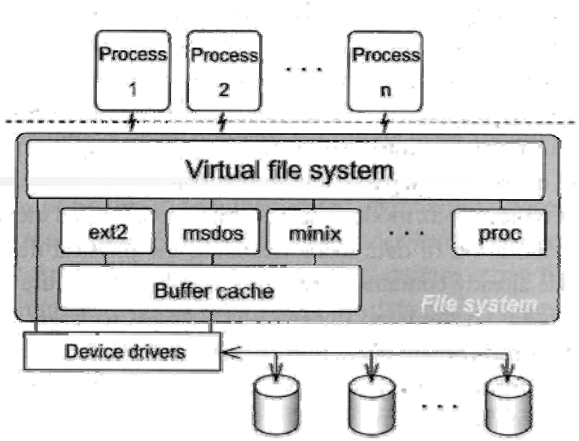
- File System Architecture
- Pathname
- File Tree
- Mounting
- File Types
- inode and file
- Link
- File Access Mode
- Changing File Owner
- FreeBSD bonus flags

File System Architecture (1)

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



File System Architecture (2)



- 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

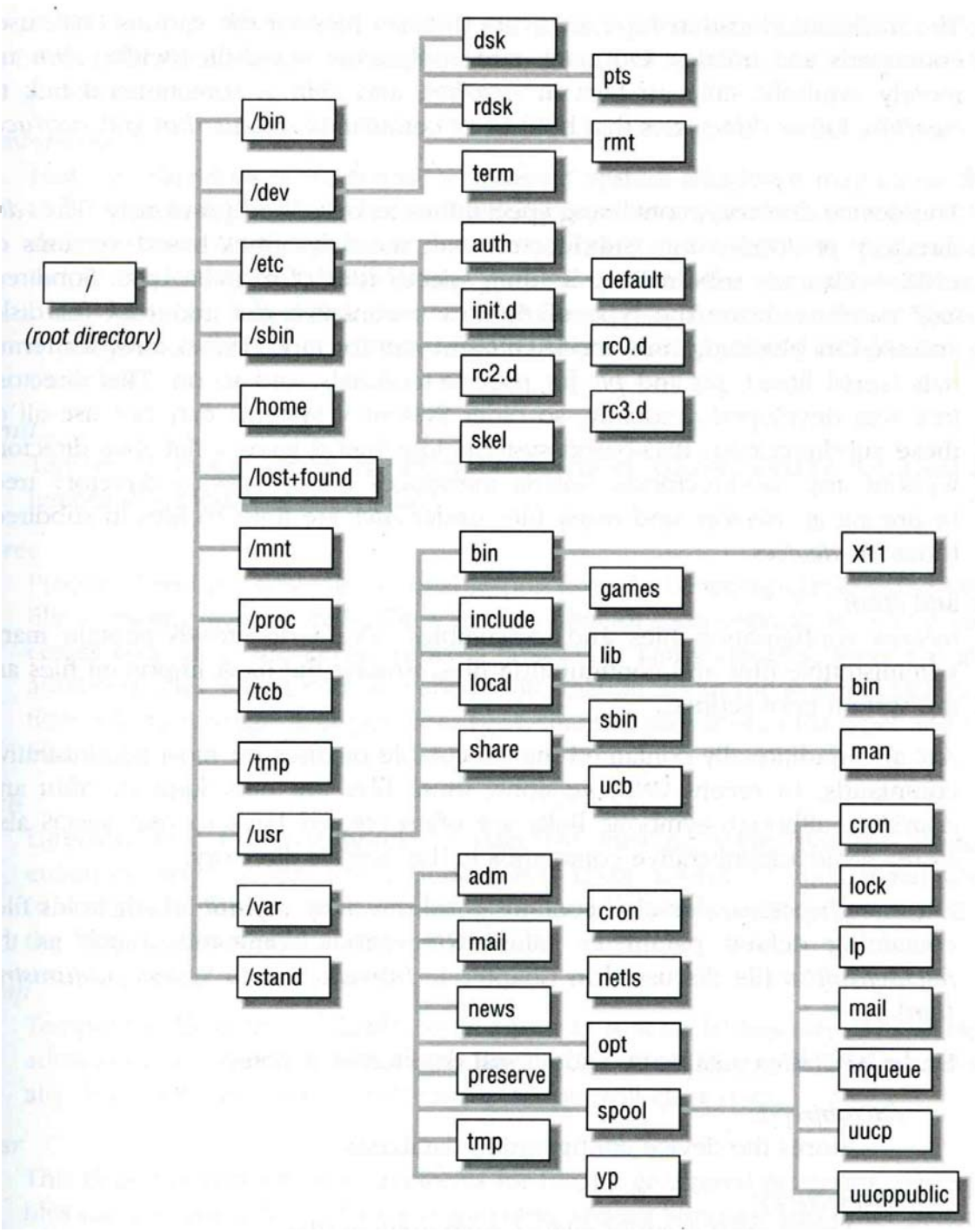
File System Architecture (3)

- Objects in the 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 of path
  - 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
- Constrains of pathname
  - Single component: ≦ 255 characters
  - Single absolute path: ≦ 1023 characters

File Tree (1)


/
(root directory)
/bin
/dev
/etc
/sbin
/home
/lost+found
/mnt
/proc
/tcb
/tmp
/usr
/var
/stand
dsk
rdsk
term
pts
rmt
auth
init.d
rc2.d
skel
default
rc0.d
rc3.d
bin
include
local
share
games
lib
sbin
ucb
X11
bin
man
adm
mail
news
preserve
tmp
cron
netls
opt
spool
yp
cron
lock
lp
mail
mqueue
uucp
uucppublic

File Tree (2)

- standard directories and their contents

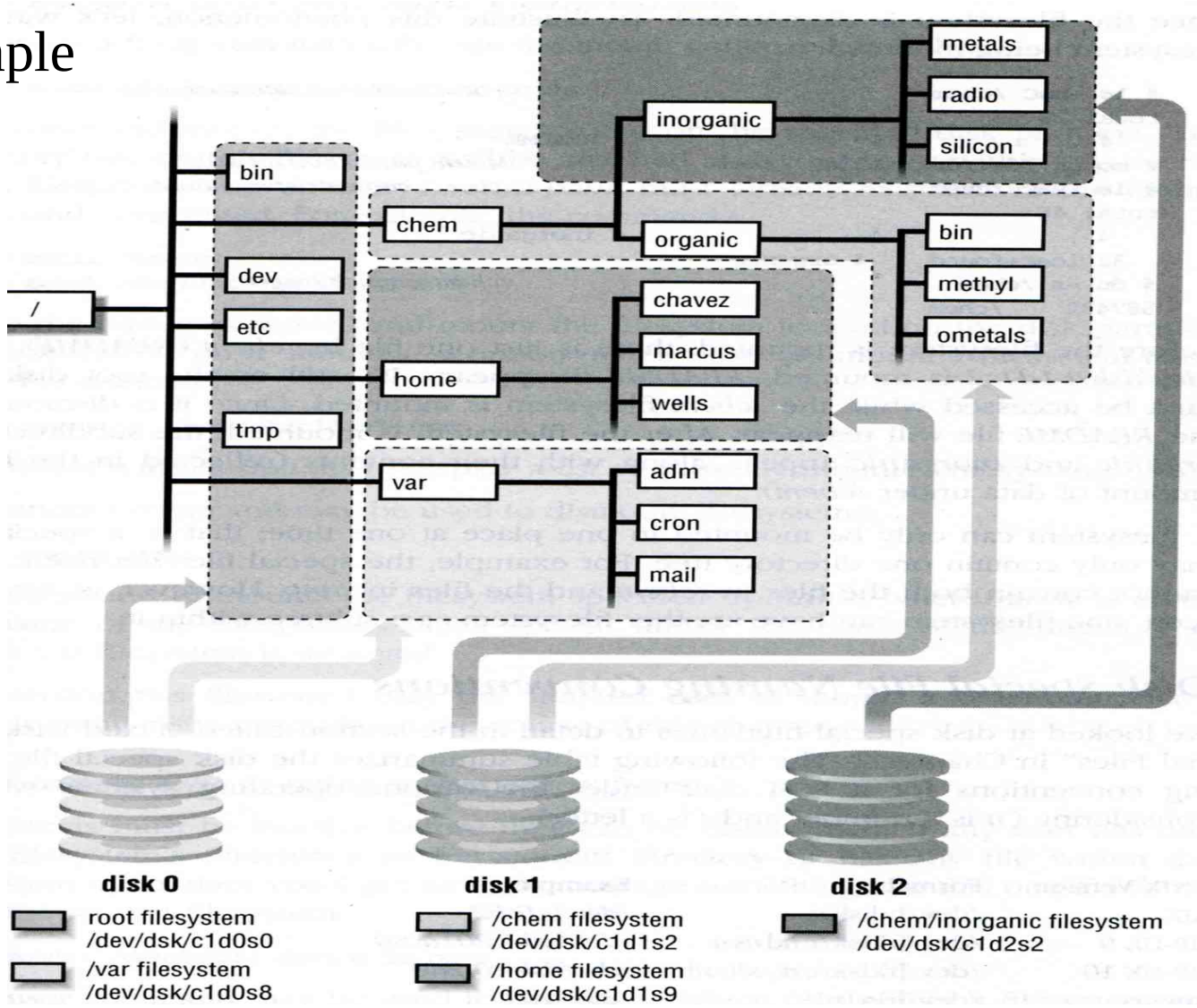
pathname	Contents
/	The root directory
/bin or /sbin	Commands needed for minimal system operability
/usr/bin	Executable files
/usr/local/bin	Local executable
/usr/local/sbin	Local system maintenance commands
/etc	Critical startup and configuration files
/usr/local/etc	Local system configuration files
/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 (1)

- 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

Mounting file system (2)

- Example



Mounting file system (3)

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

Ex. bsd1's /etc/fstab

```
# Device        Mountpoint    FStype    Options       Dump    Pass#
/dev/ad0s1b     none          swap      sw            0       0
/dev/ad0s1a     /             ufs       rw            1       1
/dev/ad0s1e     /backup       ufs       rw            2       2
/dev/ad0s1d     /home         ufs       rw            2       2
/dev/acd0       /cdrom        cd9660    ro,noauto     0       0
csduty:/bsdhome /bsdhome      nfs       rw,noauto     0       0
```

Mounting file system (4)

- Unmounting File Stsyem
  - "umount" command
    - % umount node | device
      - Ex: umount /home, umount /dev/ad0s1e
  - 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

Mounting file system (5)

- lsof, fuser and fstat commands
  - lsof (/usr/ports/sysutils/lsof) – list open files

```
% lsof /home/chwong
COMMAND   PID   USER   FD   TYPE DEVICE SIZE/OFF   NODE NAME
lsof    27097   root  cwd   VDIR   0,75      512 730114 /home/chwong
lsof    27098   root  cwd   VDIR   0,75      512 730114 /home/chwong
tcsh    44998 chwong  cwd   VDIR   0,75      512 730114 /home/chwong
sh      55546  mysql  cwd   VDIR   0,75      512 730114 /home/chwong
tcsh    99025 chwong  cwd   VDIR   0,75      512 730114 /home/chwong
```

  - fuser (/usr/ports/sysutils/fuser)
    - list IDs of all processes that have one or more files open

```
% fuser /home/chwong
/home/chwong: 55546c 44998c 99025c 27169c
```

  - fstat (FreeBSD)

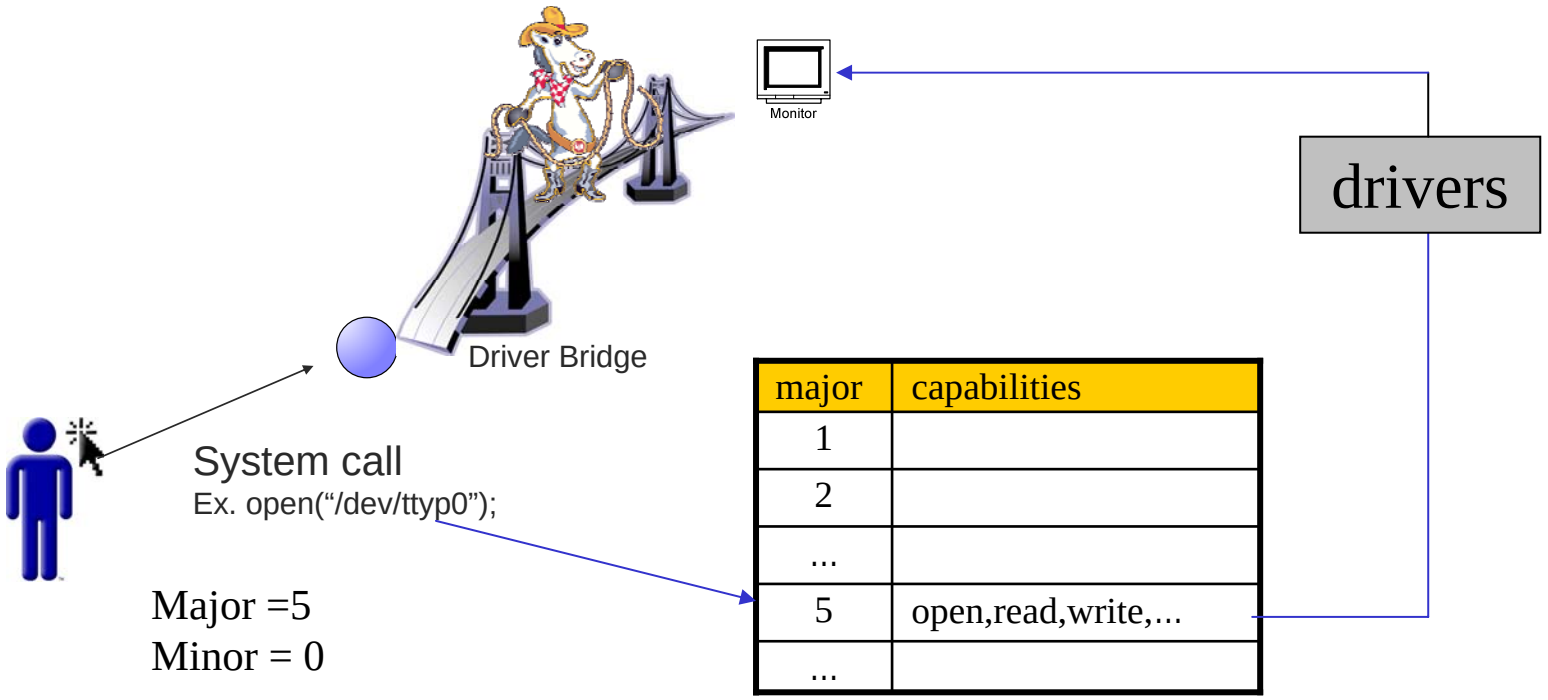
```
% fstat /home/chwong
USER     CMD          PID   FD MOUNT      INUM MODE         SZ|DV R/W NAME
chwong   fstat      27101   wd /        730114 drwxr-xr-x     512   r  /home/chwong
chwong   tcsh       99025   wd /        730114 drwxr-xr-x     512   r  /home/chwong
chwong   tcsh       44998   wd /        730114 drwxr-xr-x     512   r  /home/chwong
mysql    sh         55546   wd /        730114 drwxr-xr-x     512   r  /home/chwong
```

File Types (1)

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

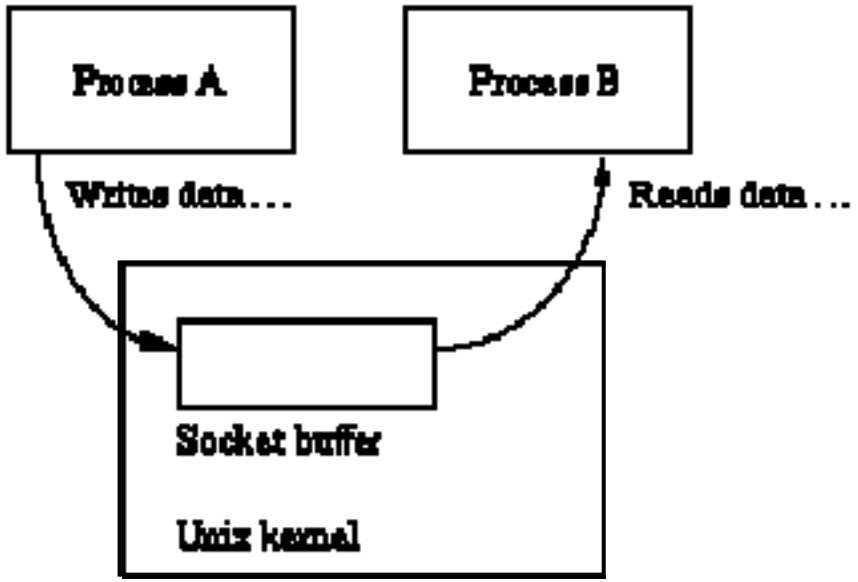
File Types (2)

- 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



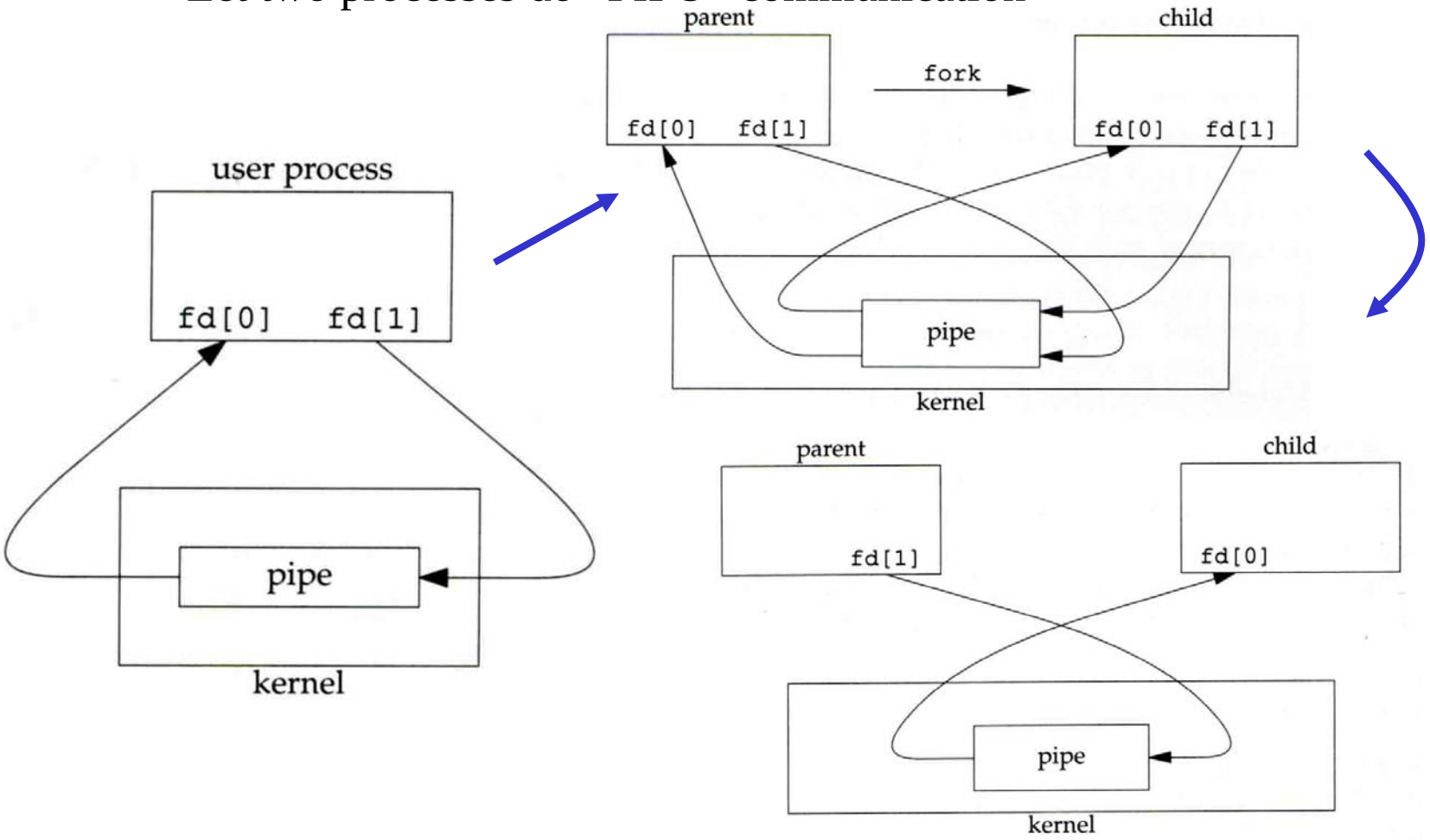
File Types (3)

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



File Types (4)

- Named Pipes
  - Let two processes do “FIFO” communication



File Types (5)

- Symbolic Link
  - A file which points to another pathname
  - % ln –s ori-file soft-file
  - Like “short-cut” in Windows

File Types (6)

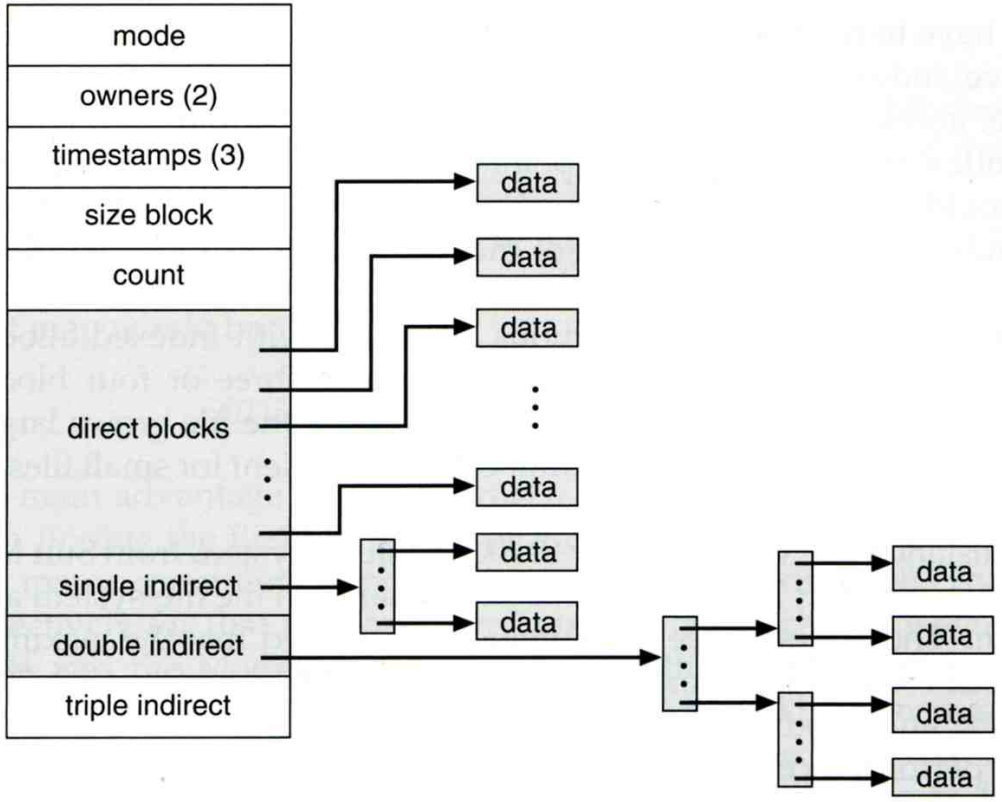
- File type encoding used by ls

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

```
chwong@chbsd:/var/run> ls -al
total 36
drwxr-xr-x   6 root   wheel      512 Oct  2 23:32 .
drwxr-xr-x  22 root   wheel      512 Oct  1 05:00 ..
srw-rw-rw-   1 root   wheel        0 Sep 30 21:00 log
```

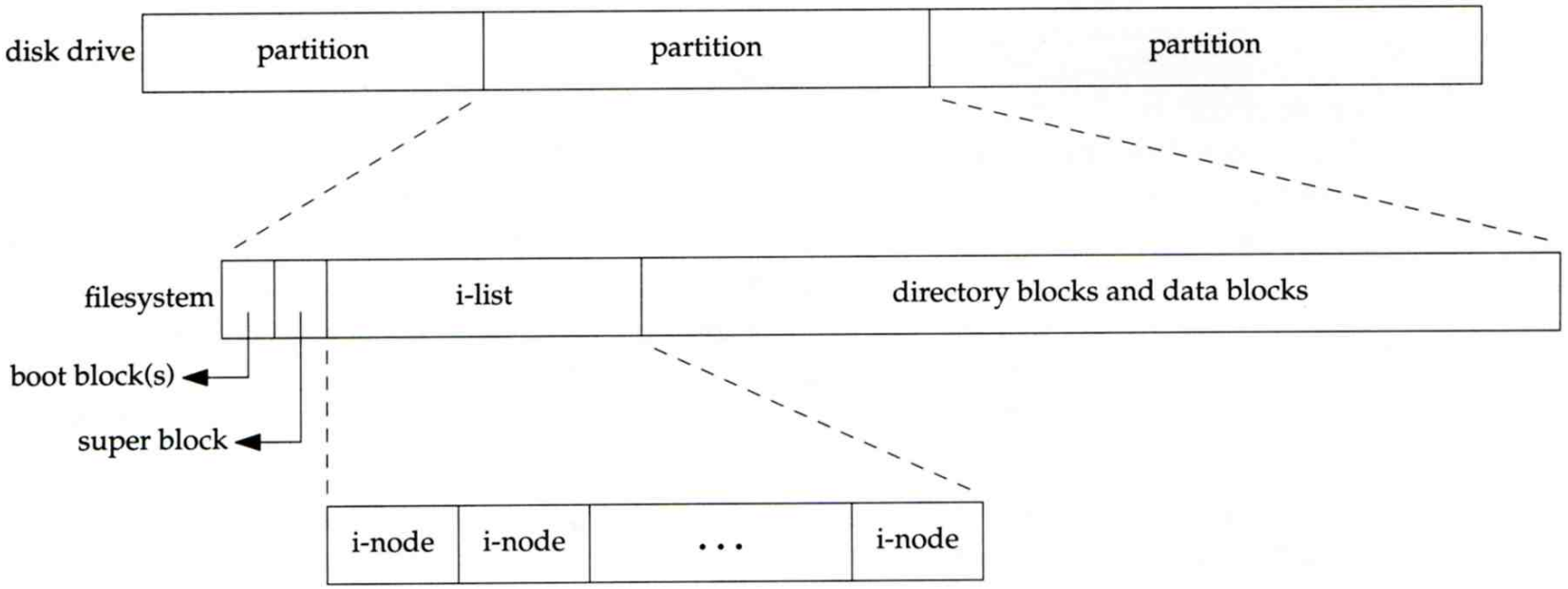
inode and file (1)

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



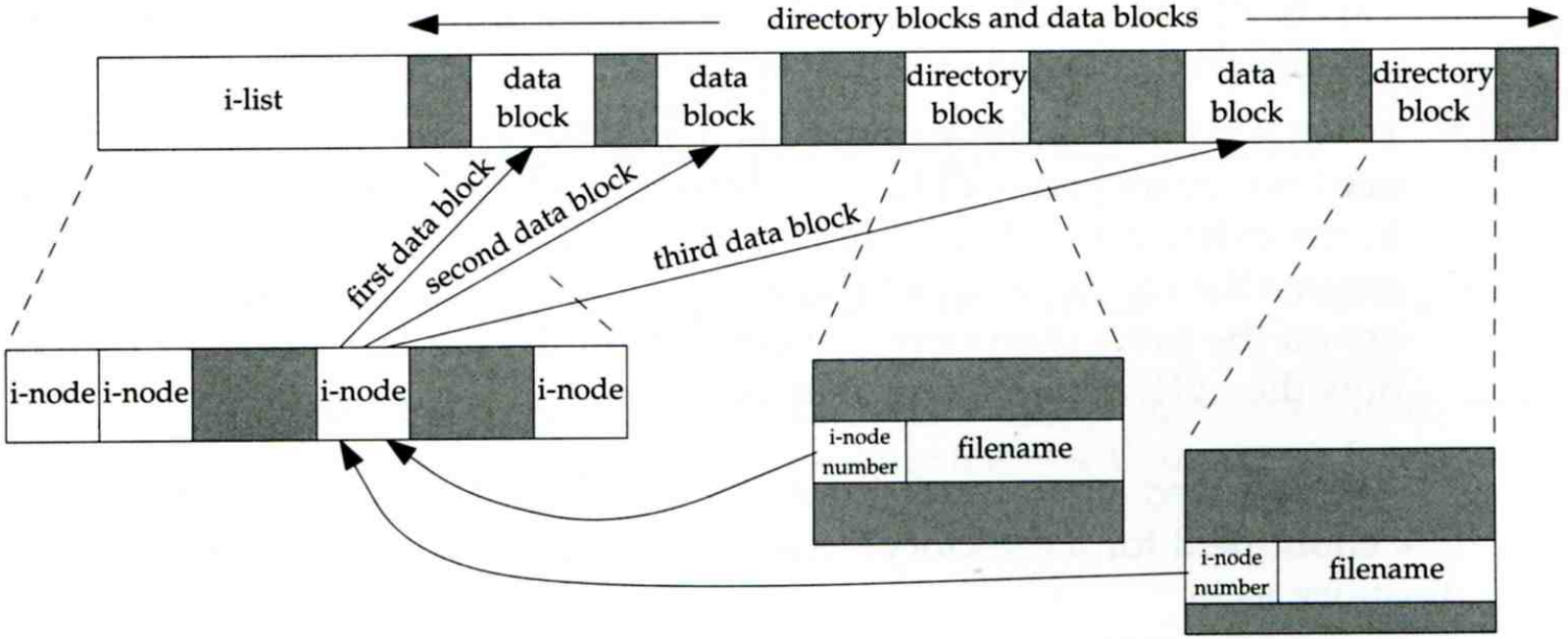
inode and file (2)

- Filesystem
  - Boot blocks
  - Super block
  - Inode list
  - Data block



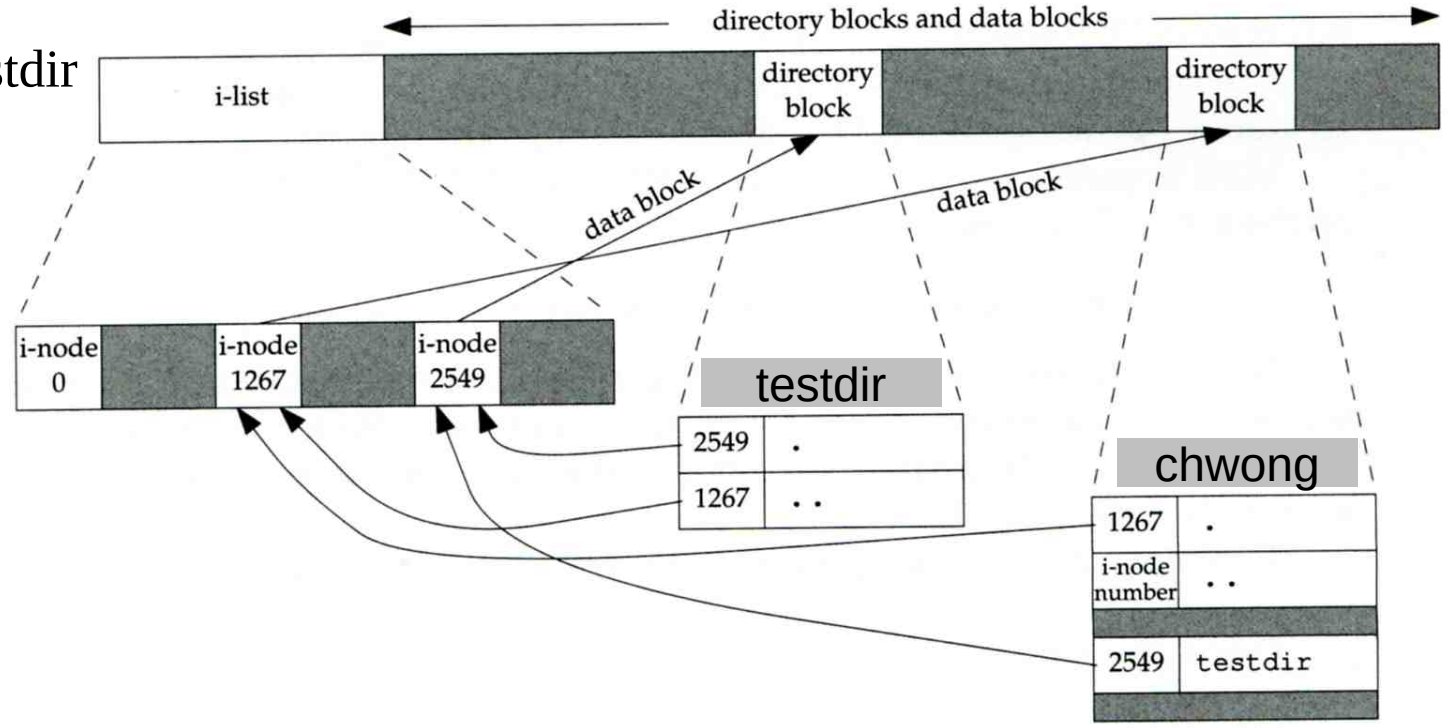
inode and file (3)

- More detail of inode and data block



inode and file (4)

- Example
  - .
  - ..
  - testdir

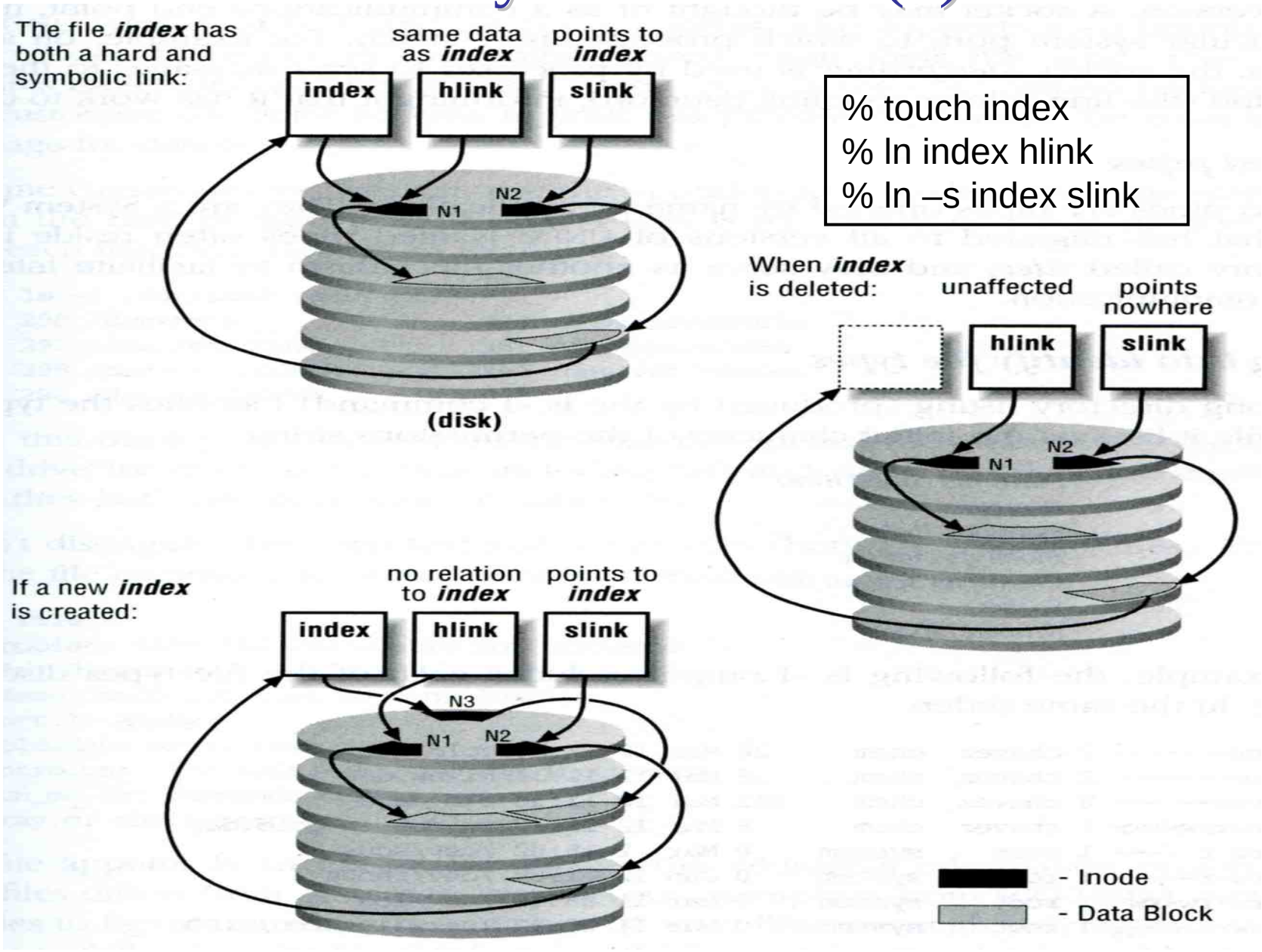


/home/chwong/testdir

Hard Link V.S. Symbolic Link (1)

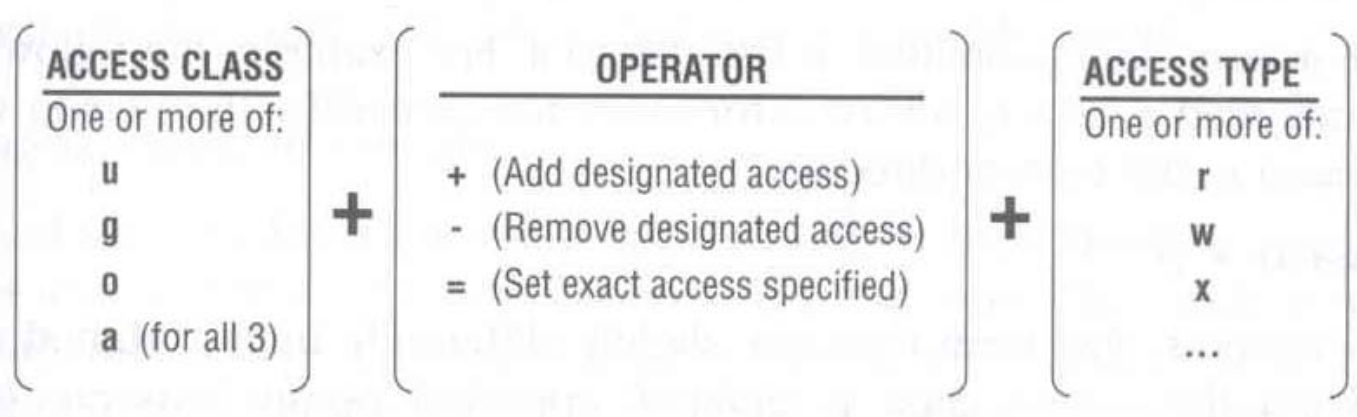
- Link
  - Hard link
    - associate two or more filenames with the same inode
    - % ln ori-file hard-file
  - Soft (symbolic) link
    - A file which points to another pathname
    - % ln –s ori-file soft-file

Hard Link V.S. Symbolic Link (2)



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

File Access Mode (4)

- Assign default permissions: umask
  - Shell built-in command
  - Inference the default permissions given to the files newly created.
  - The newly created file permission:
    - Use full permission bit (file: 666, dir: 777) xor umask value.
  - Example:

umask	New File	New Dir
022	- rw- r-- r--	d rwx r-x r-x
033	- rw- r-- r--	d rwx r-- r--
066	- rw- --- ---	d rwx --x --x
000	- rw- rw- rw-	d rwx rwx rwx
477	- r-- --- ---	d r-x --- ---
777	- --- --- ---	d --- --- ---

Changing File Owner

- Changing File Owner
  - Commands:
    - chown -- change user owner
    - chgrp -- change group owner
- Change the file ownership and group ownership
  - % chown –R chwong /home/chwong
  - % chgrp –R cs /home/chwong
  - % chown –R chown:cs /home/chwong

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)
  - …

```
chbsd [/home/chwong/test] -chwong- touch file
chbsd [/home/chwong/test] -chwong- ls -lo
-rw-r--r--  1 chwong  user  - 0 Oct  3 18:23 file
chbsd [/home/chwong/test] -chwong- chflags uunlnk file
chbsd [/home/chwong/test] -chwong- ls -lo
-rw-r--r--  1 chwong  user  uunlnk 0 Oct  3 18:23 file
chbsd [/home/chwong/test] -chwong- rm -f file
rm: file: Operation not permitted
chbsd [/home/chwong/test] -chwong- sudo rm -f file
rm: file: Operation not permitted
chbsd [/home/chwong/test] -chwong- chflags nouunlnk file
chbsd [/home/chwong/test] -chwong- rm -f file
chbsd [/home/chwong/test] -chwong-
```