UNIX

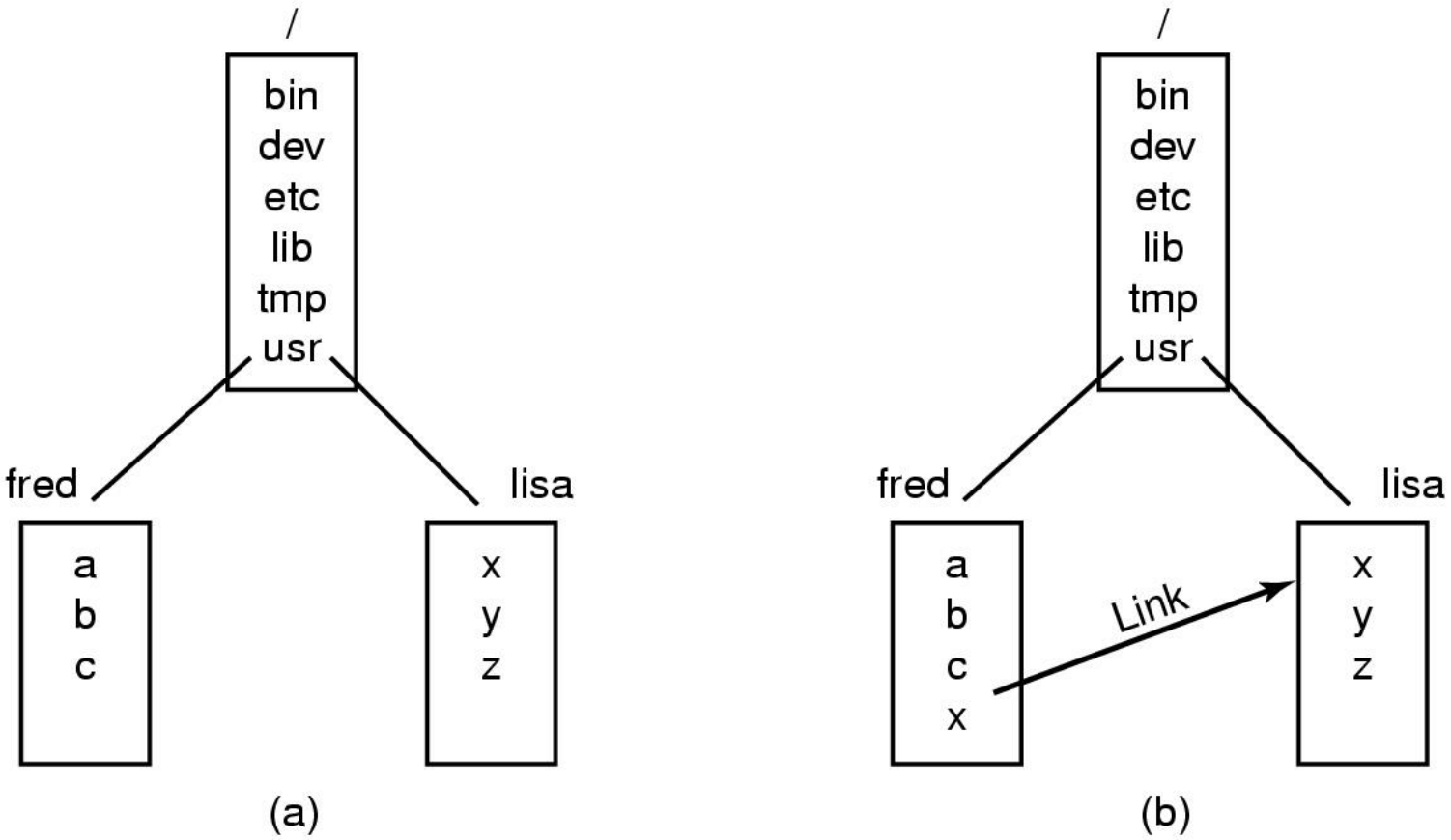
File System ed Input/Output

- Caratteristiche del FS Unix
- Implementazione
- Il file system di Linux
- Organizzazione dell' I/O

Il file system di UNIX (1)

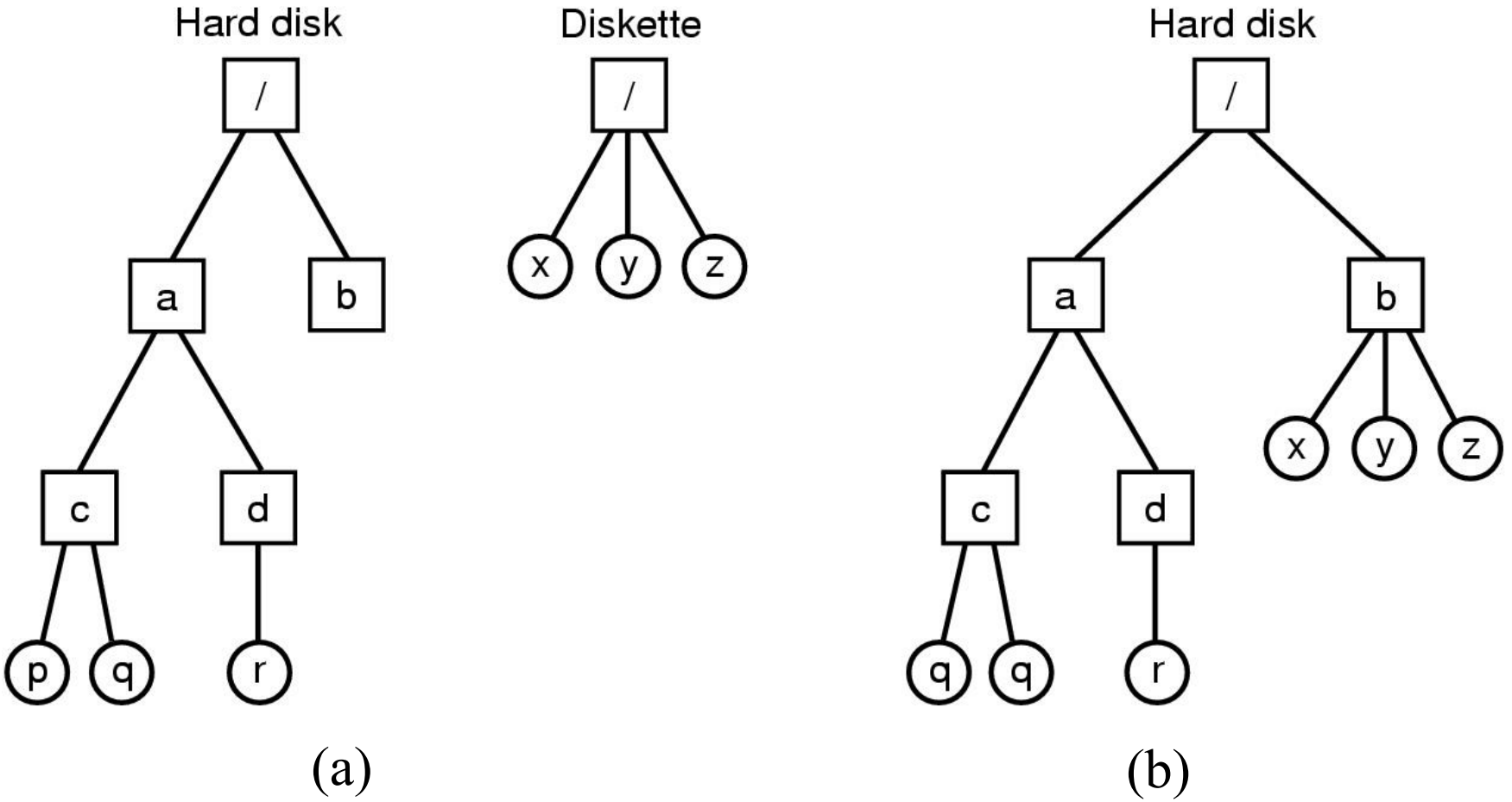
- Gerarchico
- / è la root directory ed il separatore
  - /usr/bal/file.c
- Case sensitive
- Ammette link hard e simbolici
- Permette di integrare file system diversi in un unico albero (*mounting*)

Il file system di UNIX (2)



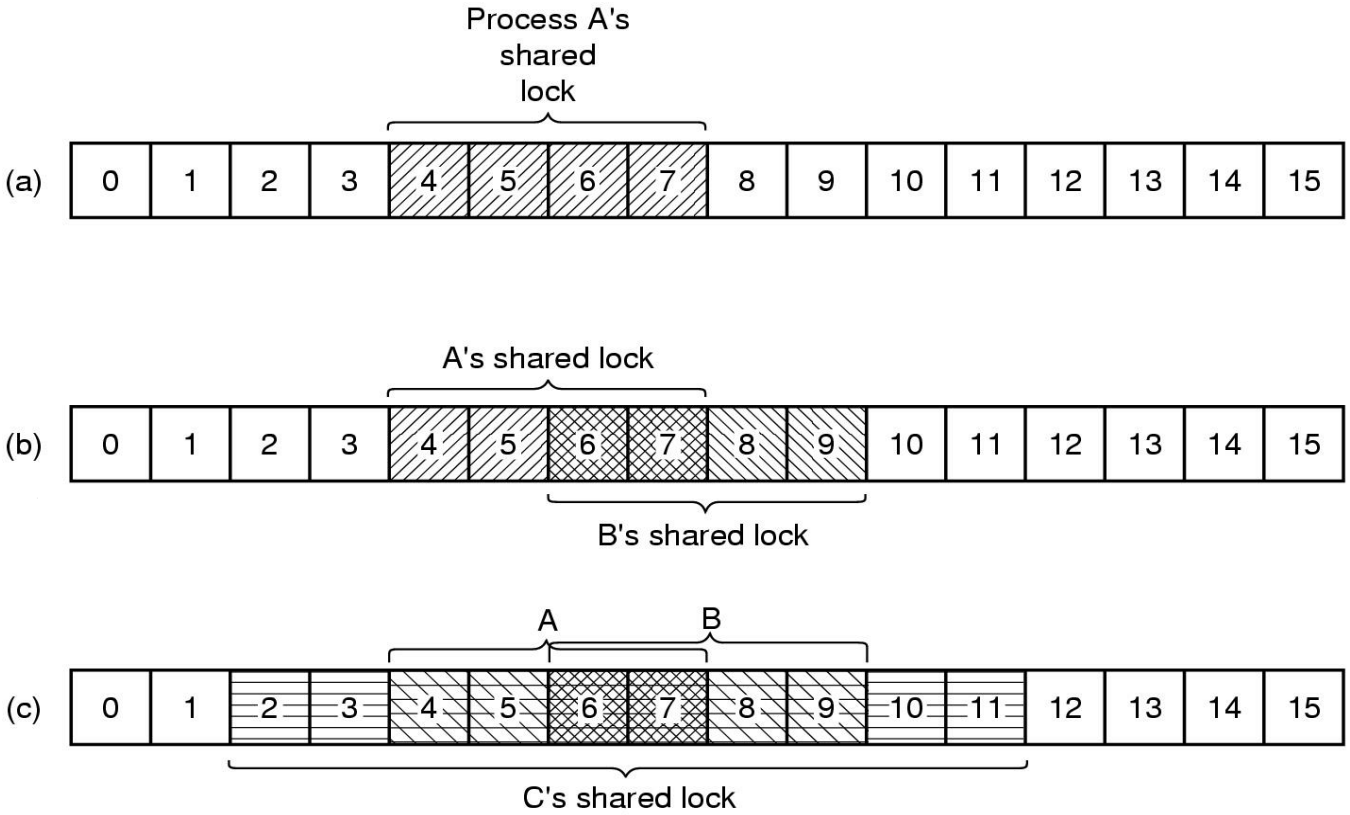
(a) prima del *linking*. (b) dopo il *linking*

Il file system di UNIX (3)



(a) prima del *mounting*. (b) dopo il *mounting*

Il *lock* dei file



(a) un file con un *lock*

(b) aggiunta di un secondo *lock*

(c) un terzo *lock*

Protezione dei file in UNIX

Binary	Symbolic	Allowed file accesses
111000000	rwx------	Owner can read, write, and execute
111111000	rwxrwx---	Owner and group can read, write, and execute
110100000	rw-r-----	Owner can read and write; group can read
110100100	rw-r--r--	Owner can read and write; all others can read
111101101	rwxr-xr-x	Owner can do everything, rest can read and execute
000000000	---------	Nobody has any access
000000111	------rwx	Only outsiders have access (strange, but legal)

Esempi di modi di protezione dei file

SC per la gestione dei file

System call	Description
fd = creat(name, mode)	One way to create a new file
fd = open(file, how, ...)	Open a file for reading, writing or both
s = close(fd)	Close an open file
n = read(fd, buffer, nbytes)	Read data from a file into a buffer
n = write(fd, buffer, nbytes)	Write data from a buffer into a file
position = lseek(fd, offset, whence)	Move the file pointer
s = stat(name, &buf)	Get a file's status information
s = fstat(fd, &buf)	Get a file's status information
s = pipe(&fd[0])	Create a pipe
s = fcntl(fd, cmd, ...)	File locking and other operations

- **s** è un codice di errore
- **fd** è un descritore di file
- **position** è un offset all'interno del file

La System Call stat

Device the file is on
I-node number (which file on the device)
File mode (includes protection information)
Number of links to the file
Identity of the file's owner
Group the file belongs to
File size (in bytes)
Creation time
Time of last access
Time of last modification

e tipo file

I campi ritornati dalla system call stat

Le SC per la gestione delle directoty

System call	Description
s = mkdir(path, mode)	Create a new directory
s = rmdir(path)	Remove a directory
s = link(oldpath, newpath)	Create a link to an existing file
s = unlink(path)	Unlink a file
s = chdir(path)	Change the working directory
dir = opendir(path)	Open a directory for reading
s = closedir(dir)	Close a directory
dirent = readdir(dir)	Read one directory entry
rewinddir(dir)	Rewind a directory so it can be reread

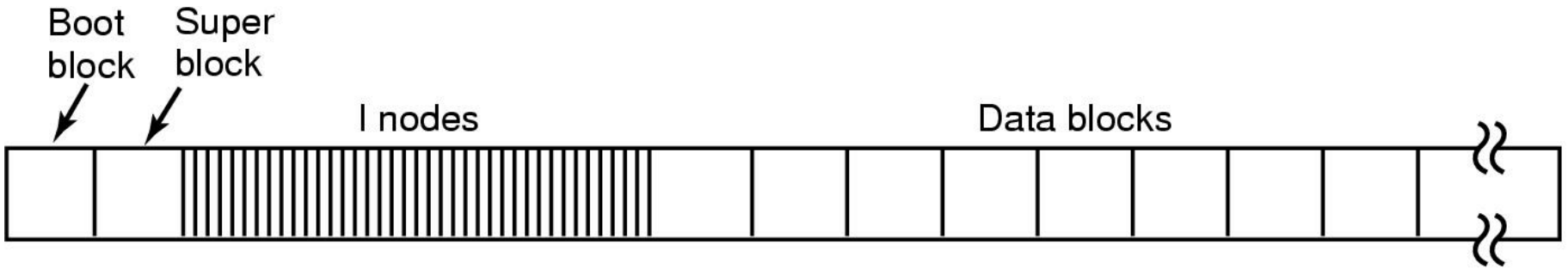
- **s** è un codice di errore
- **dir** identifica una directory
- **dirent** è un elemento di una directory

System Call per la protezione dei file

System call	Description
s = chmod(path, mode)	Change a file's protection mode
s = access(path, mode)	Check access using the real UID and GID
uid = getuid( )	Get the real UID
uid = geteuid( )	Get the effective UID
gid = getgid( )	Get the real GID
gid = getegid( )	Get the effective GID
s = chown(path, owner, group)	Change owner and group
s = setuid(uid)	Set the UID
s = setgid(gid)	Set the GID

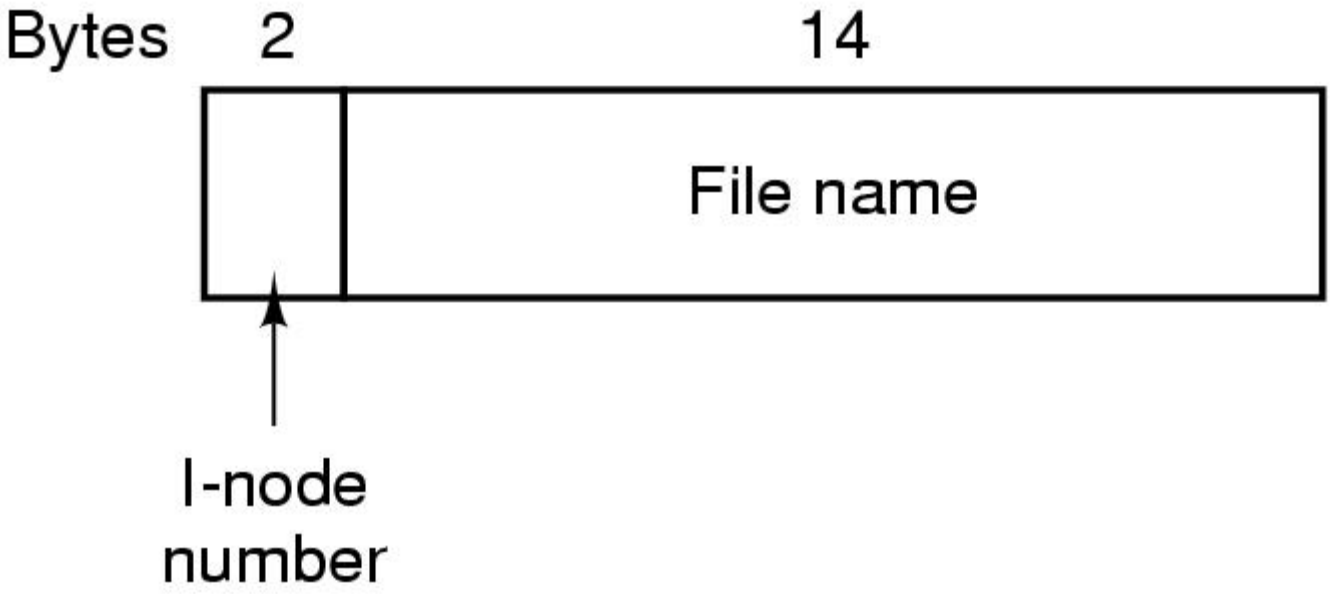
- **s** è un codice di errore
- **uid** e **gid** sono *user* e *group identifier* (UID e GID)

Implementazione del File System di UNIX (1)



Organizzazione del disco nei sistemi UNIX

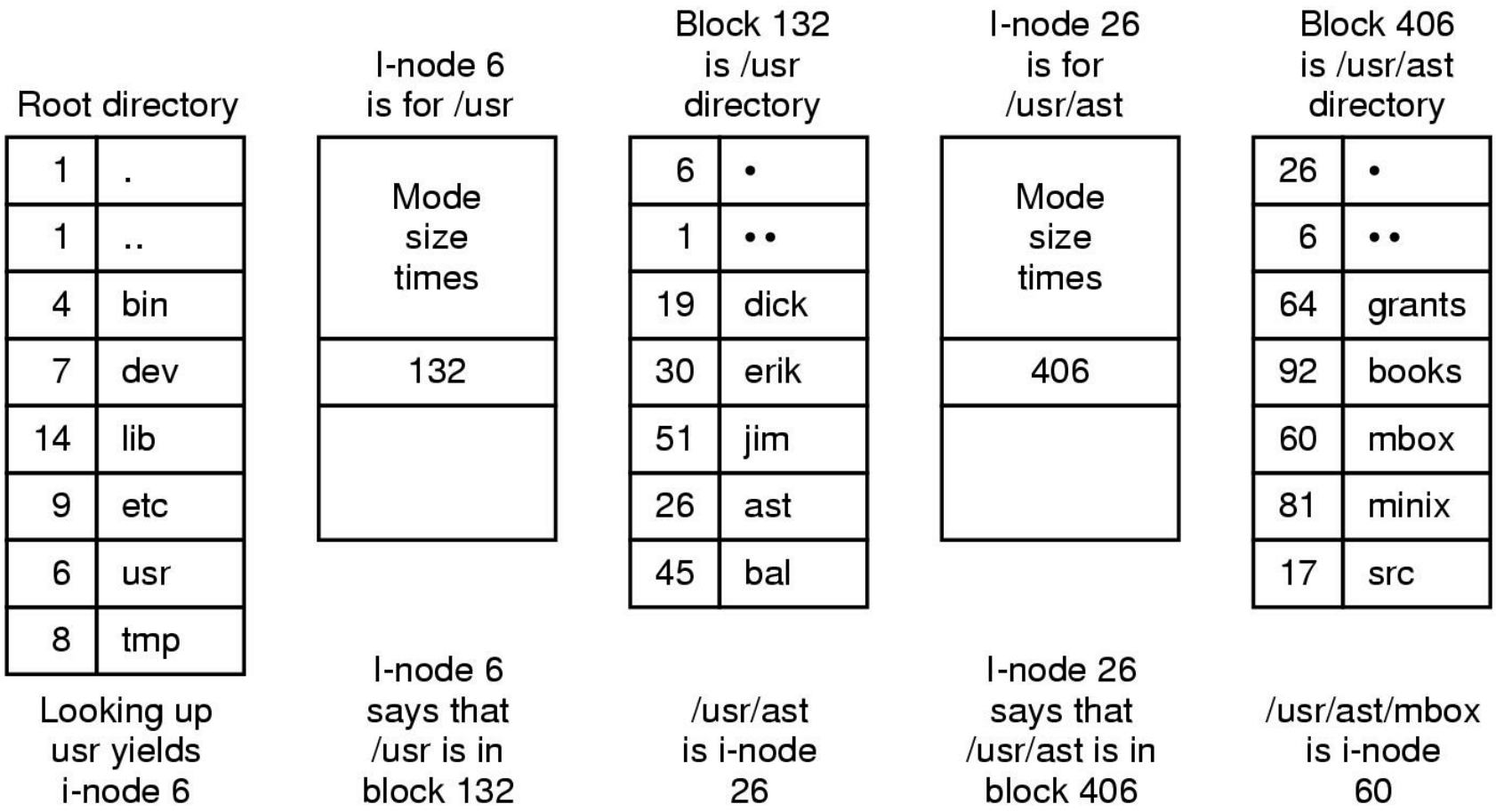
Implementazione del File System di UNIX (2)



La rappresentazione di una directory entry in UNIX V7

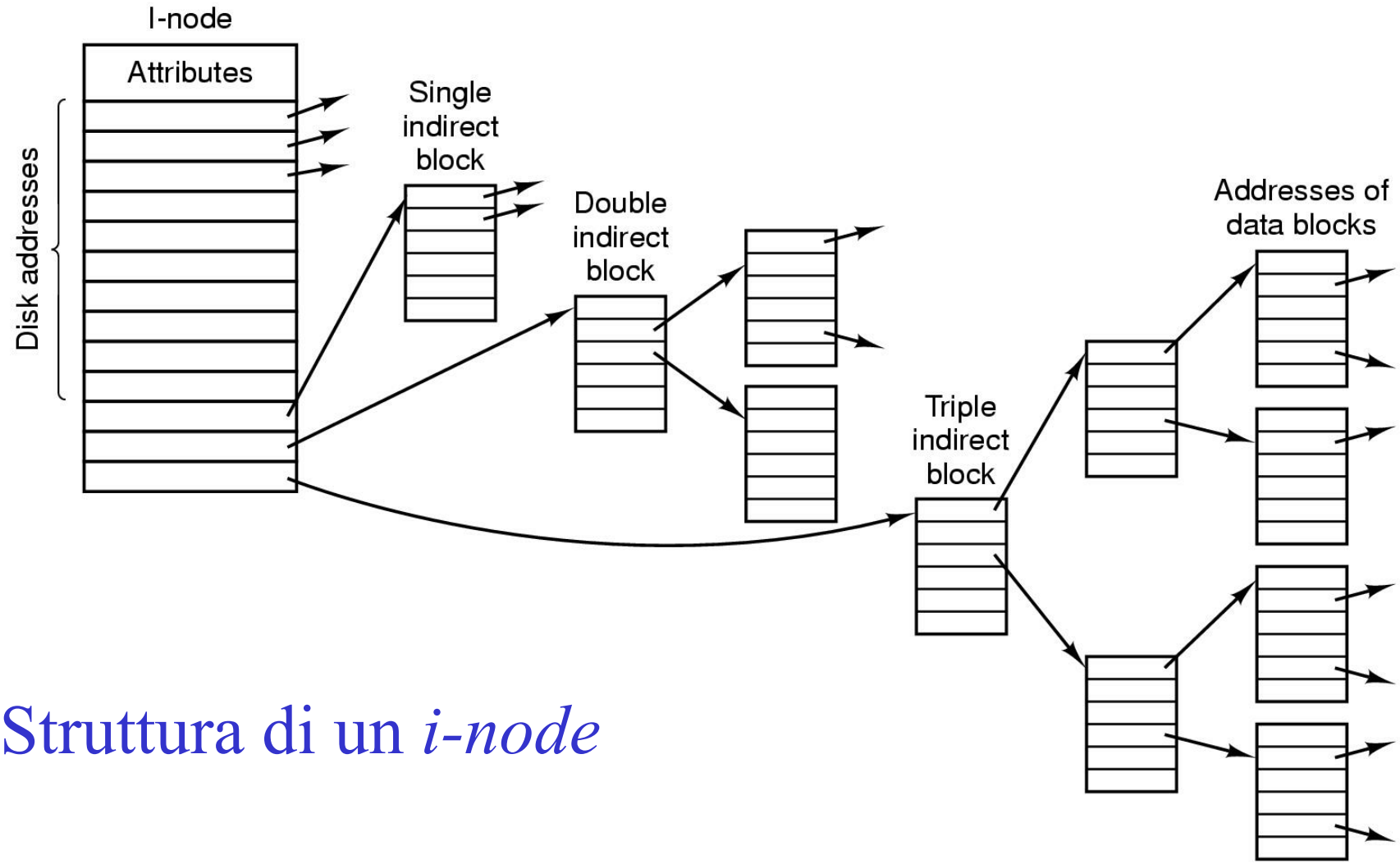
- file name max 14 caratteri

Implementazione del File System di UNIX (5)



I passi necessari per leggere */usr/ast/mbox*

Implementazione del File System di UNIX (3)



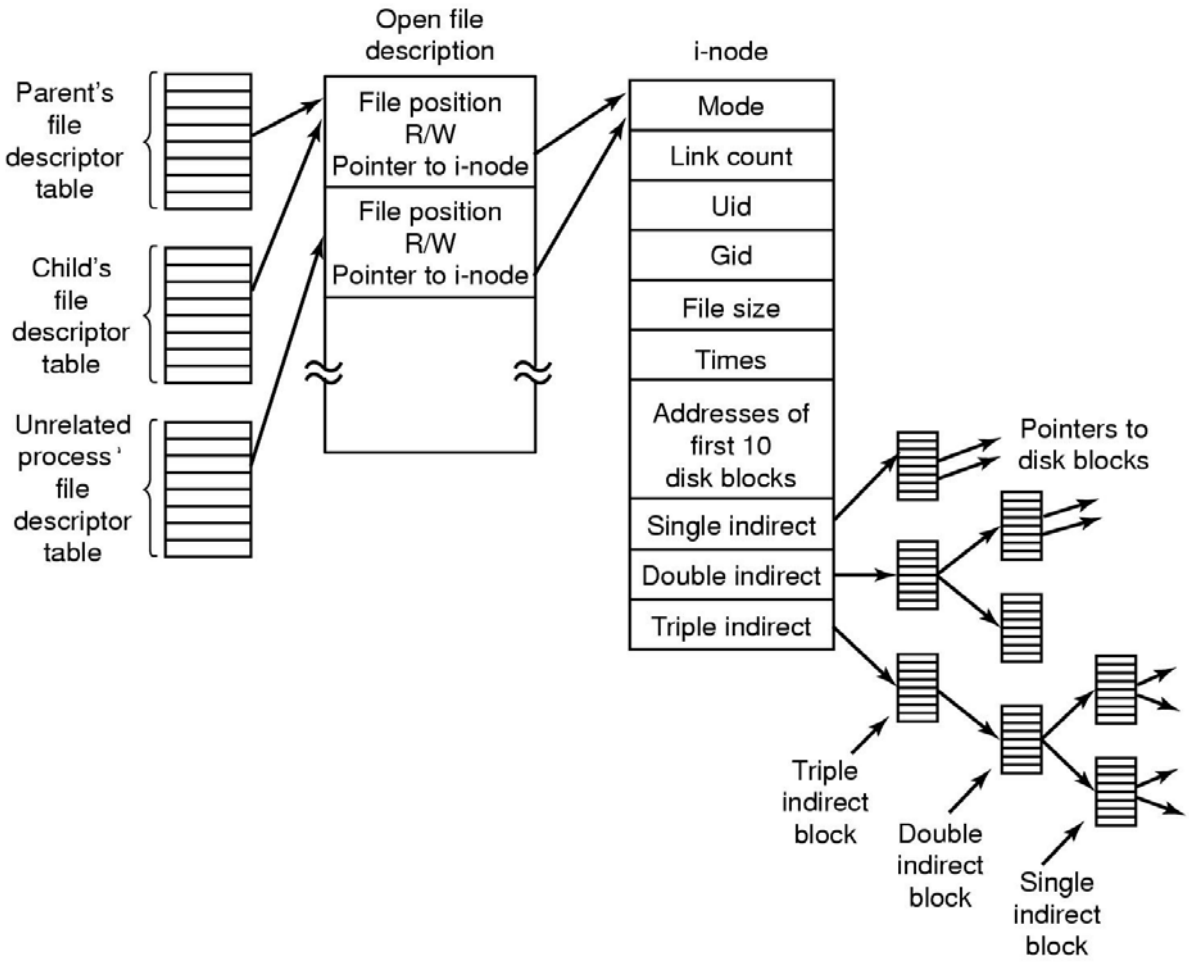
Struttura di un *i-node*

Implementazione del File System di UNIX (4)

Field	Bytes	Description
Mode	2	File type, protection bits, setuid, setgid bits
Nlinks	2	Number of directory entries pointing to this i-node
Uid	2	UID of the file owner
Gid	2	GID of the file owner
Size	4	File size in bytes
Addr	39	Address of first 10 disk blocks, then 3 indirect blocks
Gen	1	Generation number (incremented every time i-node is reused)
Atime	4	Time the file was last accessed
Mtime	4	Time the file was last modified
Ctime	4	Time the i-node was last changed (except the other times)

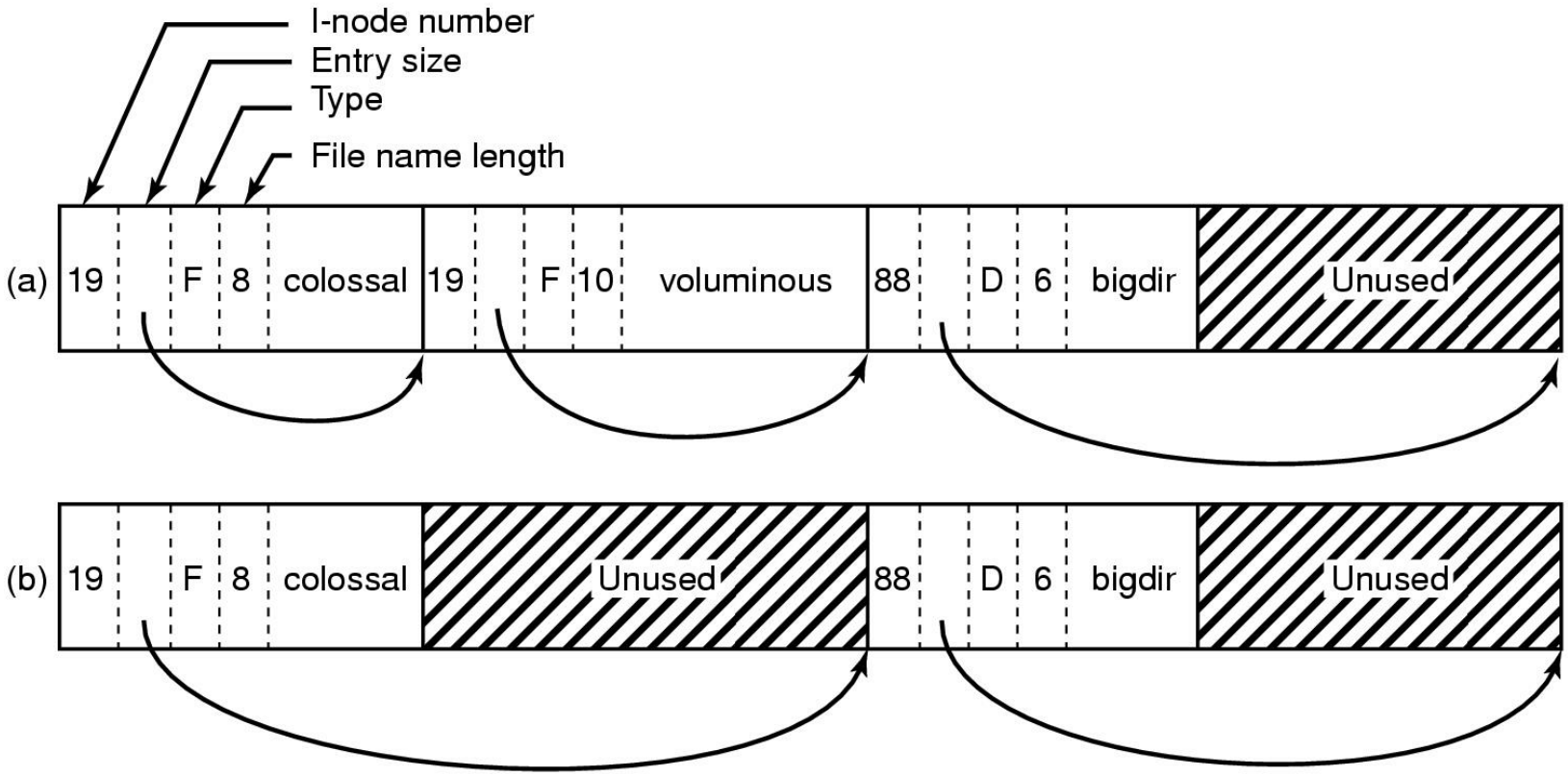
Informazioni contenute in un *i-node*

Implementazione del File system in UNIX (6)



Relazione fra *file descriptor table* e *open file description table*

Il Fast File System di Berkley (1)

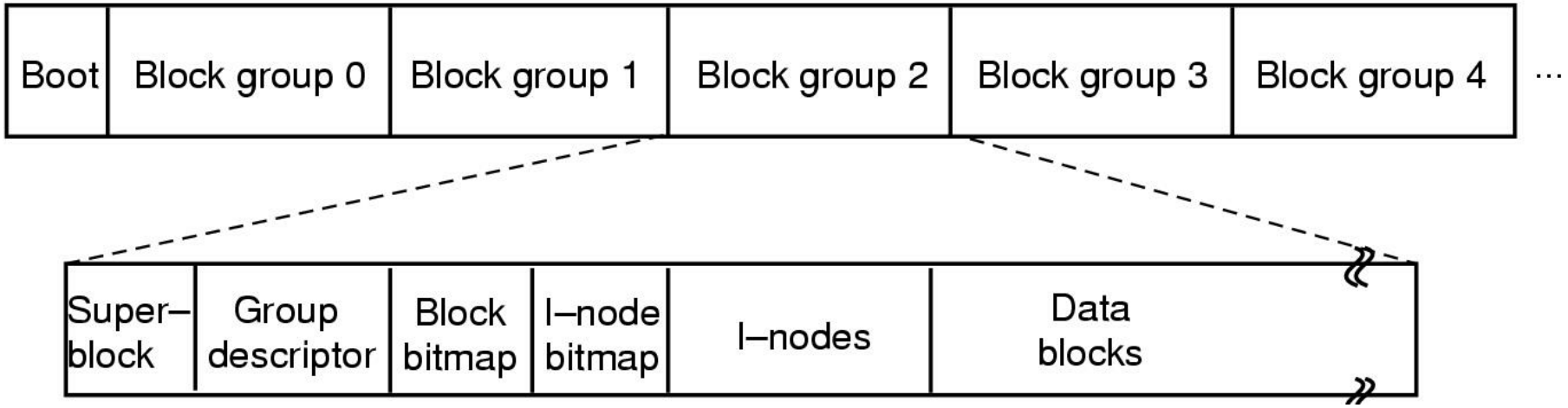


- Una directory BSD con 3 file
- La stessa directory dopo che il file *voluminous* è stato rimosso

Il Fast File System di Berkley (2)

- Caching dei nomi dei file nelle directory
- Divisione del disco in gruppi di cilindri
  - ciascuno con superblock, inode e blocchi
- Due diverse ampiezze di blocco

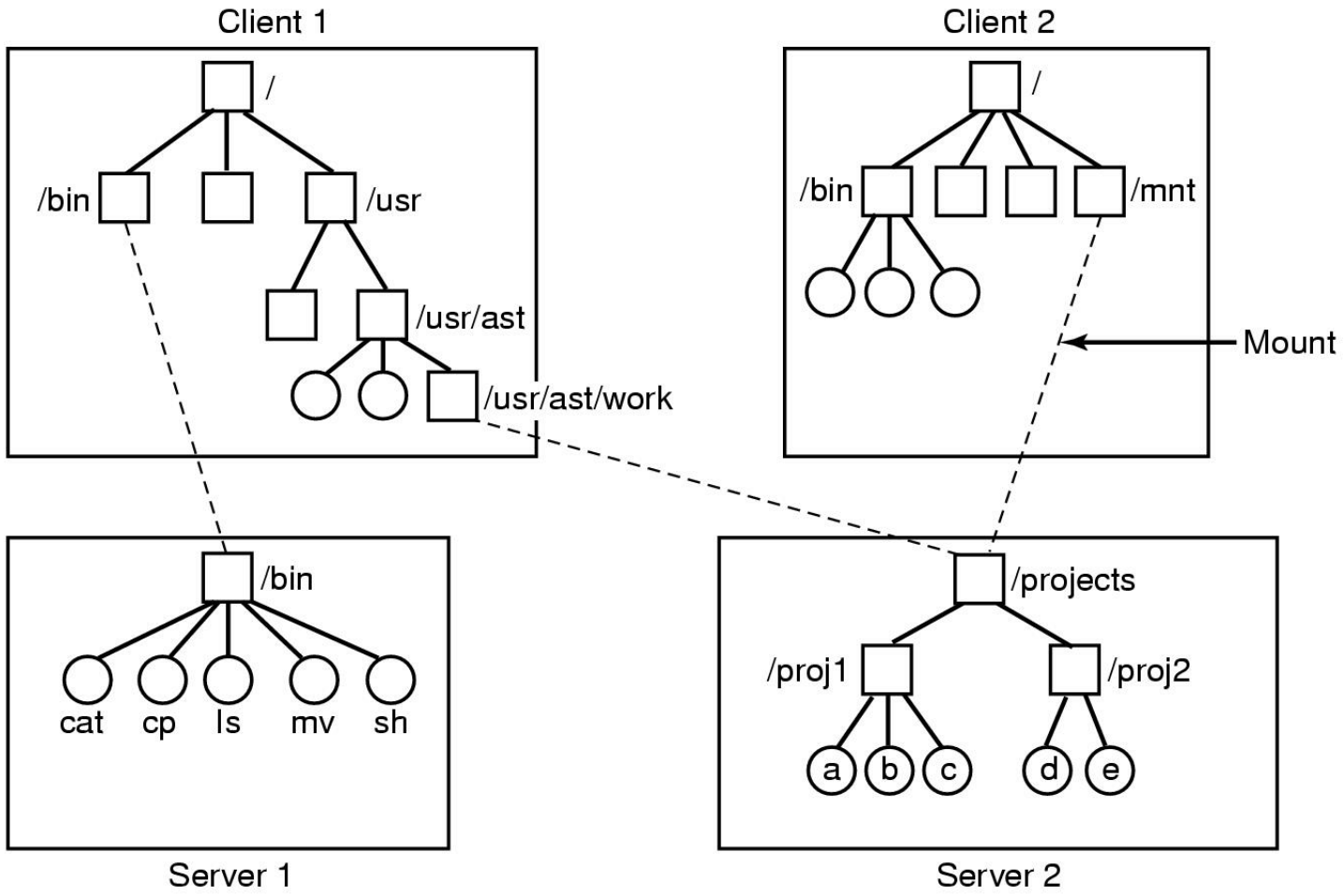
Il File System di Linux



Organizzazione del file system Ext2 :

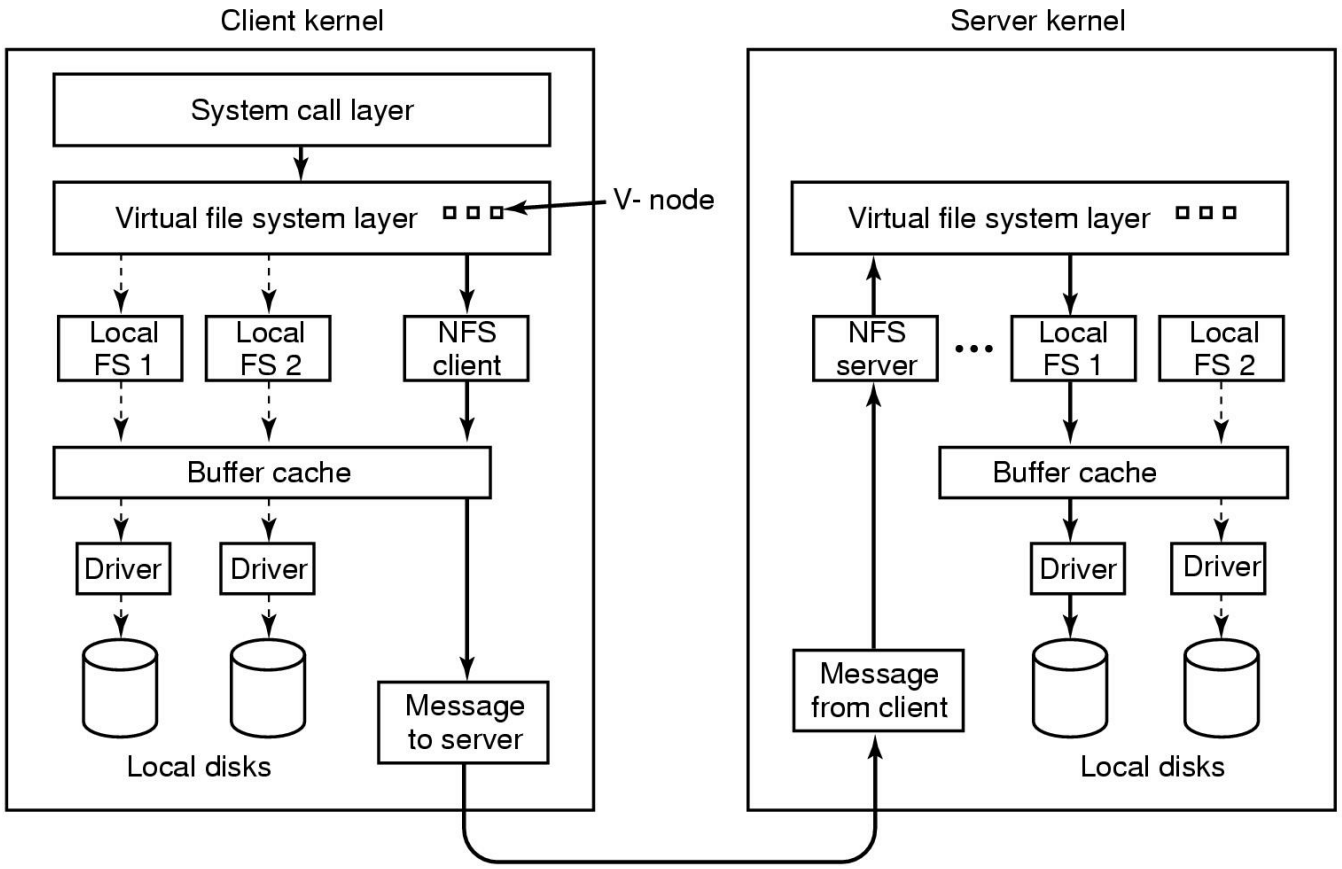
- group descriptor : indirizzo delle bitmap del gruppo, numero di directory, i-node e blocchi, indirizzo del primo i-node
- le directory sono distribuite uniformemente fra i gruppi

Network File System (NFS) (1)



Esempi di *mounting* di file system remoti

Network File System (NFS) (2)



La struttura del livello NFS

Il file system di UNIX (4)

Directory	Contents
bin	Binary (executable) programs
dev	Special files for I/O devices
etc	Miscellaneous system files
lib	Libraries
usr	User directories

Alcune directory fondamentali per la maggior parte dei sistemi UNIX

I/O in UNIX(1)

- I dispositivi sono file speciali
- L'accesso ai dispositivi viene effettuato con gli stessi comandi e le stesse chiamate di sistema utilizzate per operare sui file normali
  - open, close, read, write
  - cp file /dev/lp
- Altre chiamate di sistema permettono di settare dei parametri dipendenti dal dispositivo
  - ioctl, ...

I/O in UNIX(2)

- L'i-node corrispondente a un file speciale contiene
  - tipo di dispositivo (b- block, c-character)
  - *major device number* : indice che identifica il driver del dispositivo
  - *minor device number* : indice che identifica un dispositivo fra quelli gestiti dallo stesso driver
- Il major number è usato come indice nelle tabelle del kernel *bdevsw*, *cdevsw*
  - stabiliscono la corrispondenza fra SC e funzioni esportate dai driver

I/O in UNIX(3)

Device	Open	Close	Read	Write	Ioctl	Other
Null	null	null	null	null	null	...
Memory	null	null	mem_read	mem_write	null	...
Keyboard	k_open	k_close	k_read	error	k_ioctl	...
Tty	tty_open	tty_close	tty_read	tty_write	tty_ioctl	...
Printer	lp_open	lp_close	error	lp_write	lp_ioctl	...

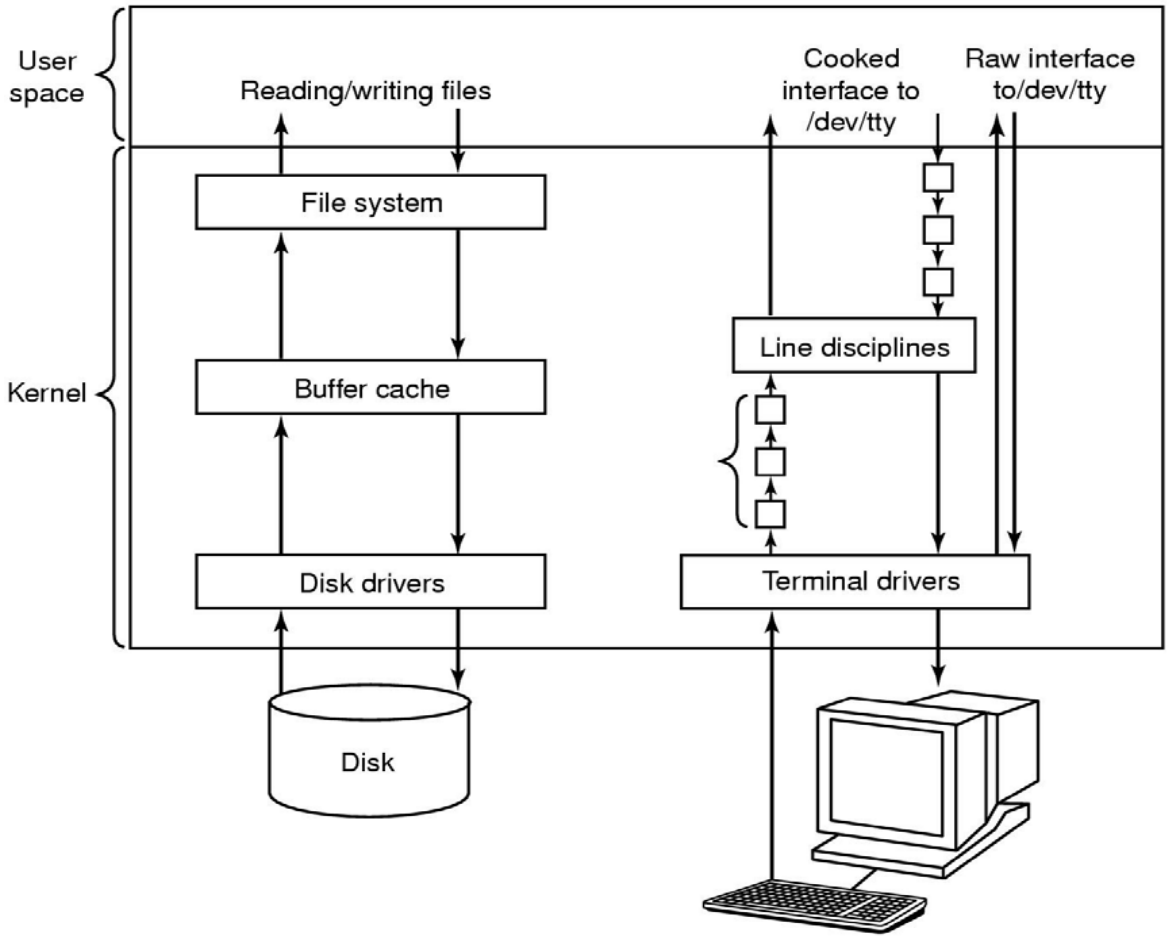
Alcuni dei campi di una tipica tabella *cdevsw*

UNIX: Gestione dei terminali

Function call	Description
s = cfsetospeed(&termios, speed)	Set the output speed
s = cfsetispeed(&termios, speed)	Set the input speed
s = cfgetospeed(&termios, speed)	Get the output speed
s = cfgtetispeed(&termios, speed)	Get the input speed
s = tcsetattr(fd, opt, &termios)	Set the attributes
s = tcgetattr(fd, &termios)	Get the attributes

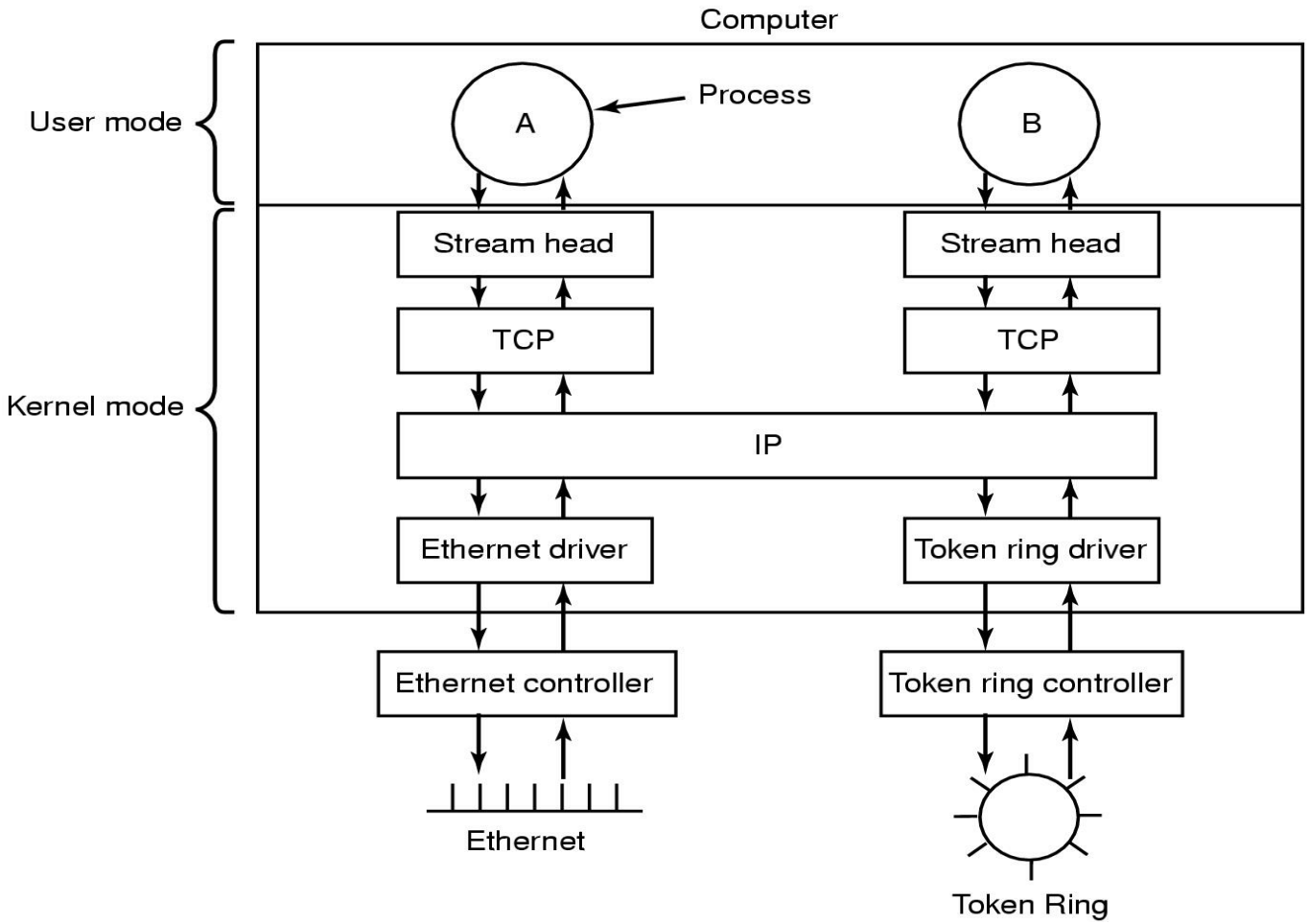
Le principali chiamate POSIX per la gestione dei terminali

I/O in UNIX (4)



Il sistema di I/O in BSD UNIX

Stream



Un esempio di stream in System V