

Programación Avanzada (TC2025)

Tema 2. Arquitectura de un sistema operativo

Instituto Tecnológico y de Estudios Superiores de Monterrey. Campus Santa Fe
Departamento de Tecnologías de Información y Electrónica
Dr. Vicente Cubells (vcubells@itesm.mx)

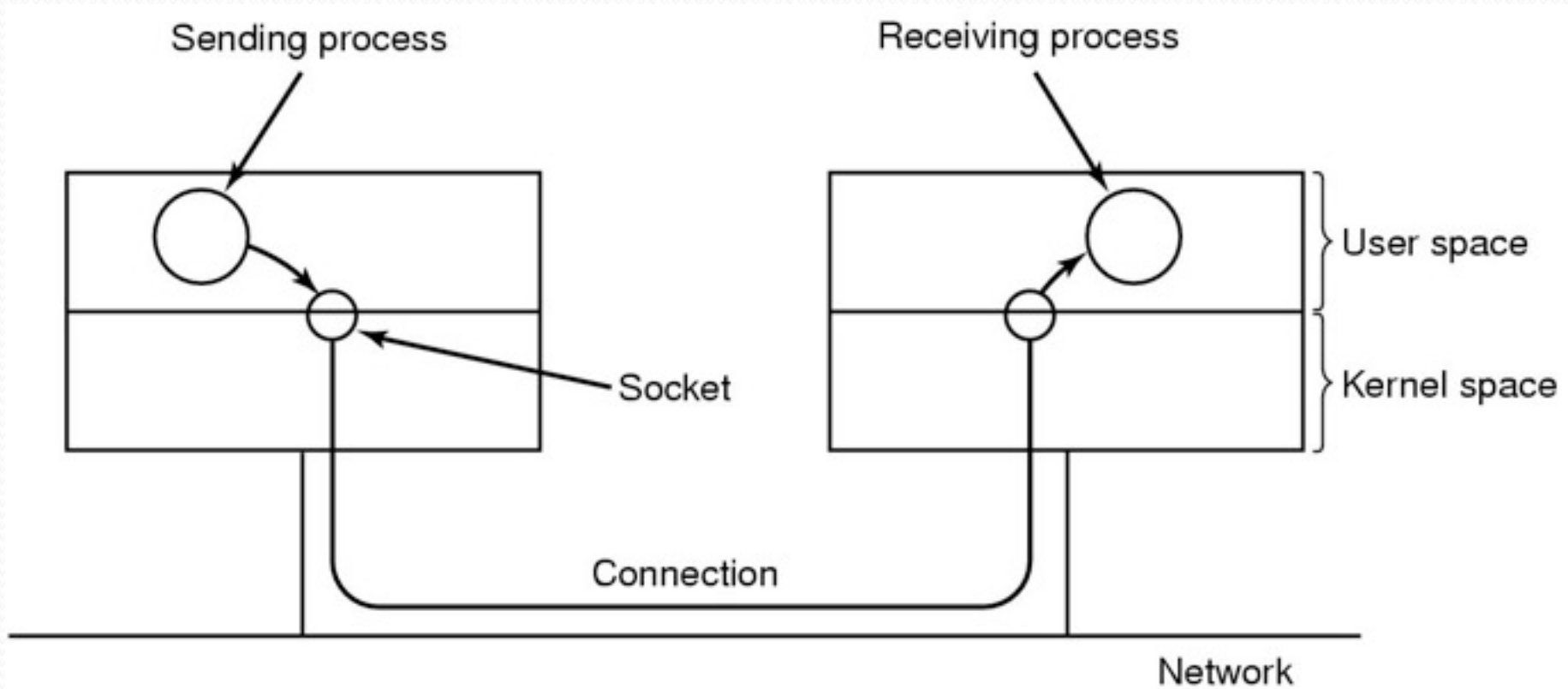
Temario

- Entrada y Salida
 - Conceptos fundamentales
 - Conectividad de redes
 - Llamadas al sistema
 - Implementación
- Sistema de archivos
 - Conceptos fundamentales
 - Llamadas al sistema
 - Implementación del sistema de archivos
 - Ejemplos de sistemas de archivos
- Seguridad
 - Conceptos fundamentales
 - Llamadas al sistema
 - Implementación

E/S en UNIX

- Todo dispositivo se modela como un archivo
 - Archivos de dispositivos (/dev)
 - Archivos de bloques
 - HDD
 - Archivos de caracteres
 - Impresora, teclado, mouse, redes, etc.
 - A cada archivo de le asocia un controlador de dispositivo

Networking



Uso de sockets

Más de sockets

- Tres tipos de sockets
 - Flujo de bytes confiable orientado a conexiones
 - Flujo de paquetes confiable orientado a conexiones
 - Transmisión no confiable de paquetes
- Llamadas al sistema
 - listen
 - connect
 - close

Administración de la Terminal

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

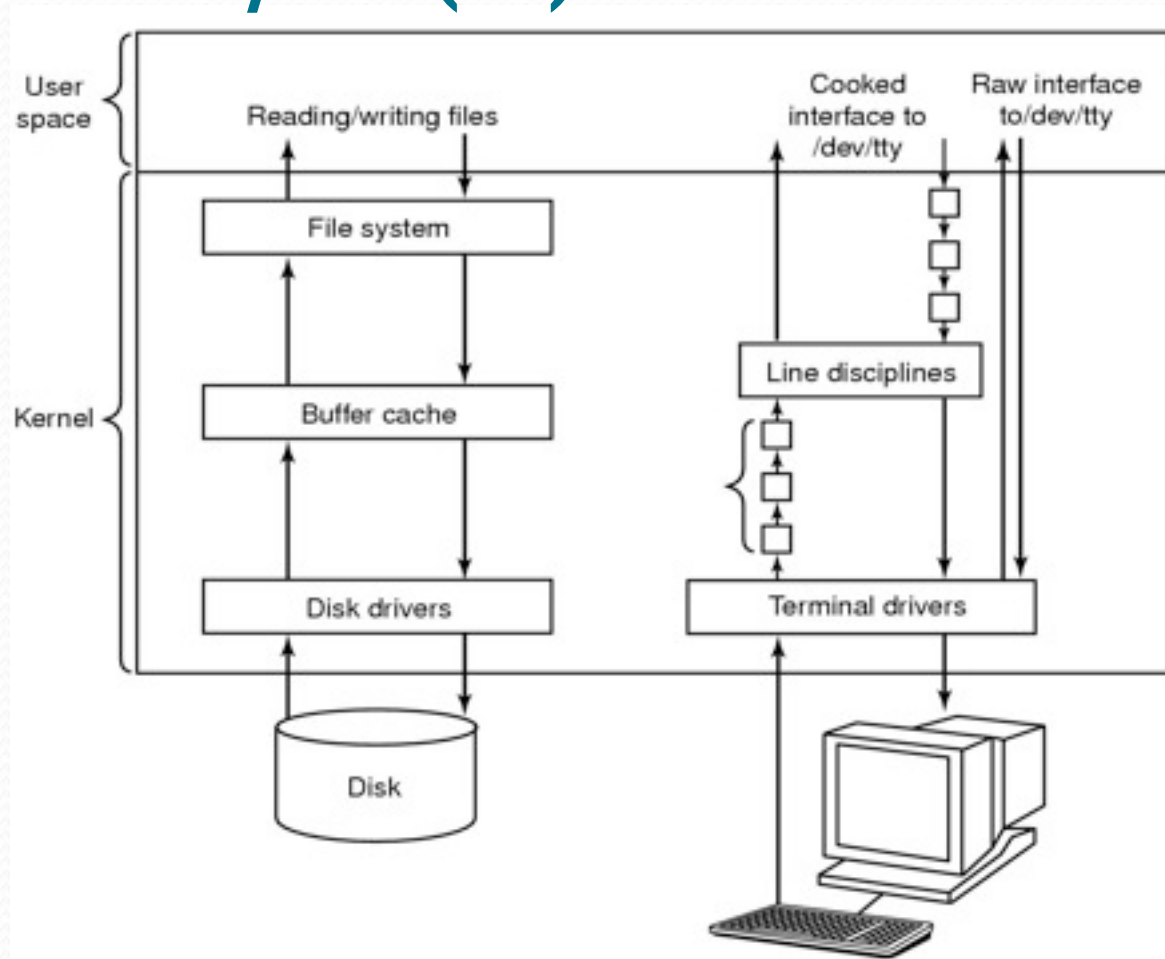
Llamadas POSIX

UNIX E/S (1)

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	...

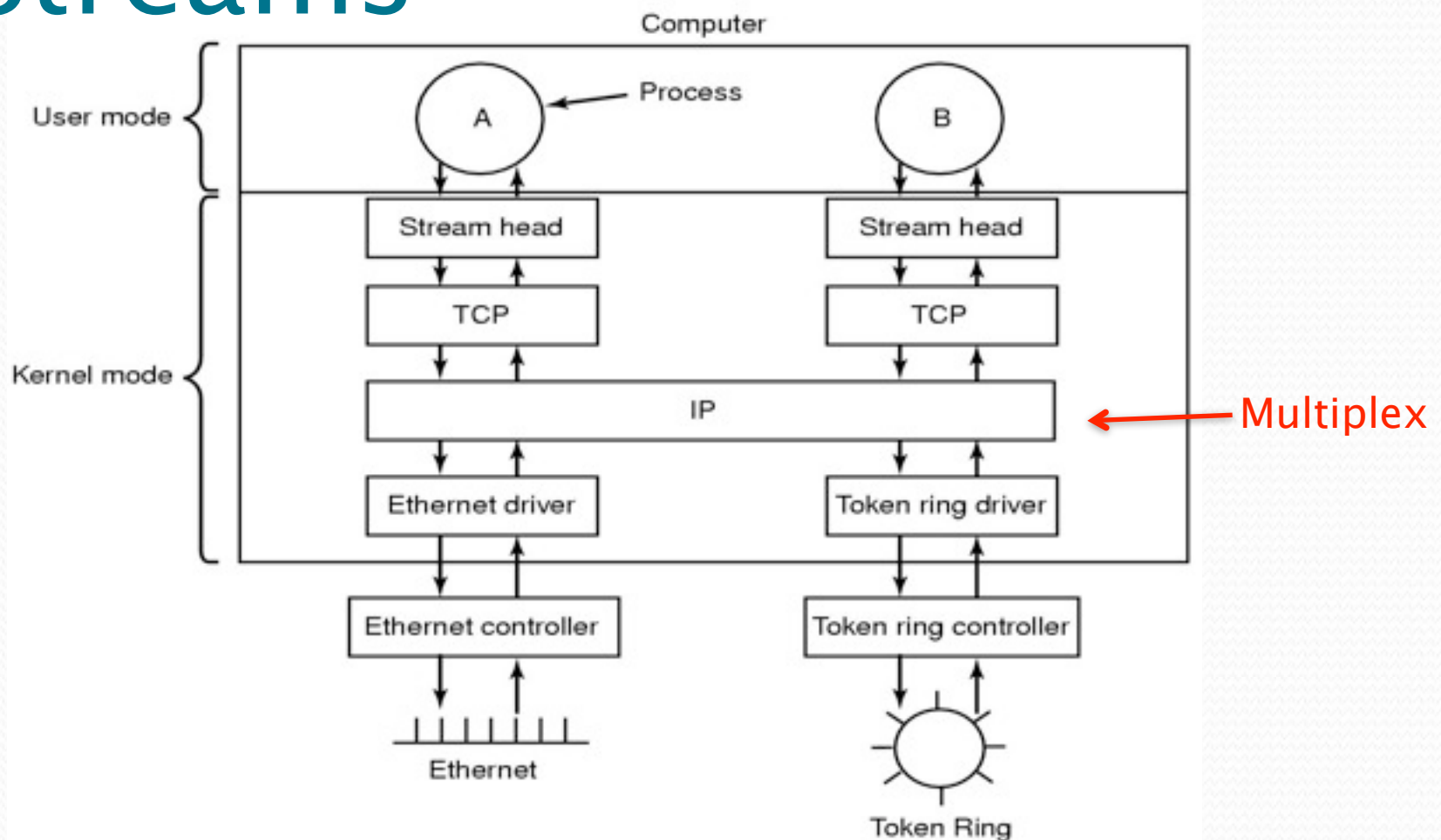
Algunos campos de la tabla cdevsw

UNIX E/S (2)



El sistema UNIX E/S en BSD

Streams



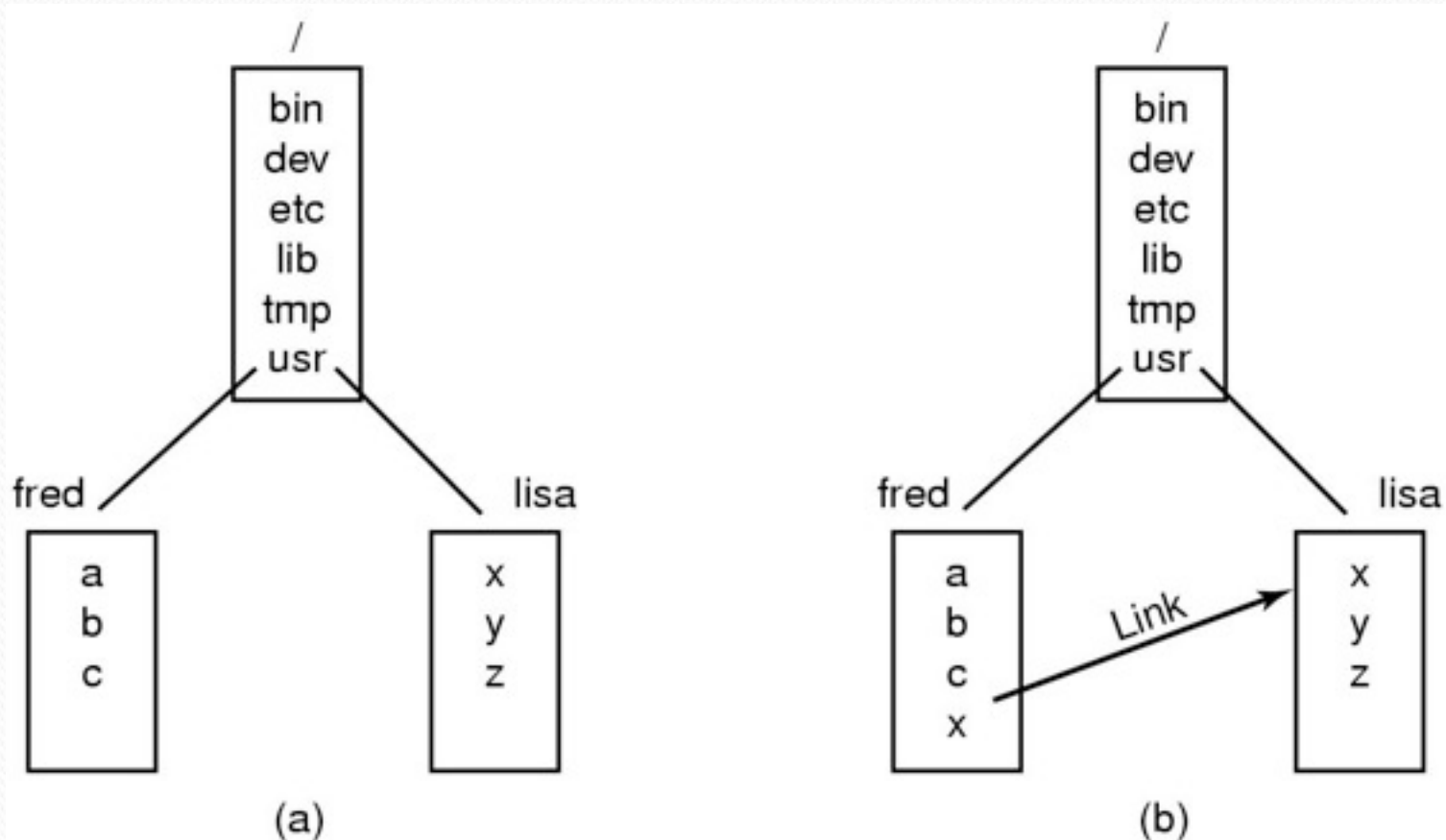
Un ejemplo de streams en System V

El sistema de archivos UNIX (1)

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

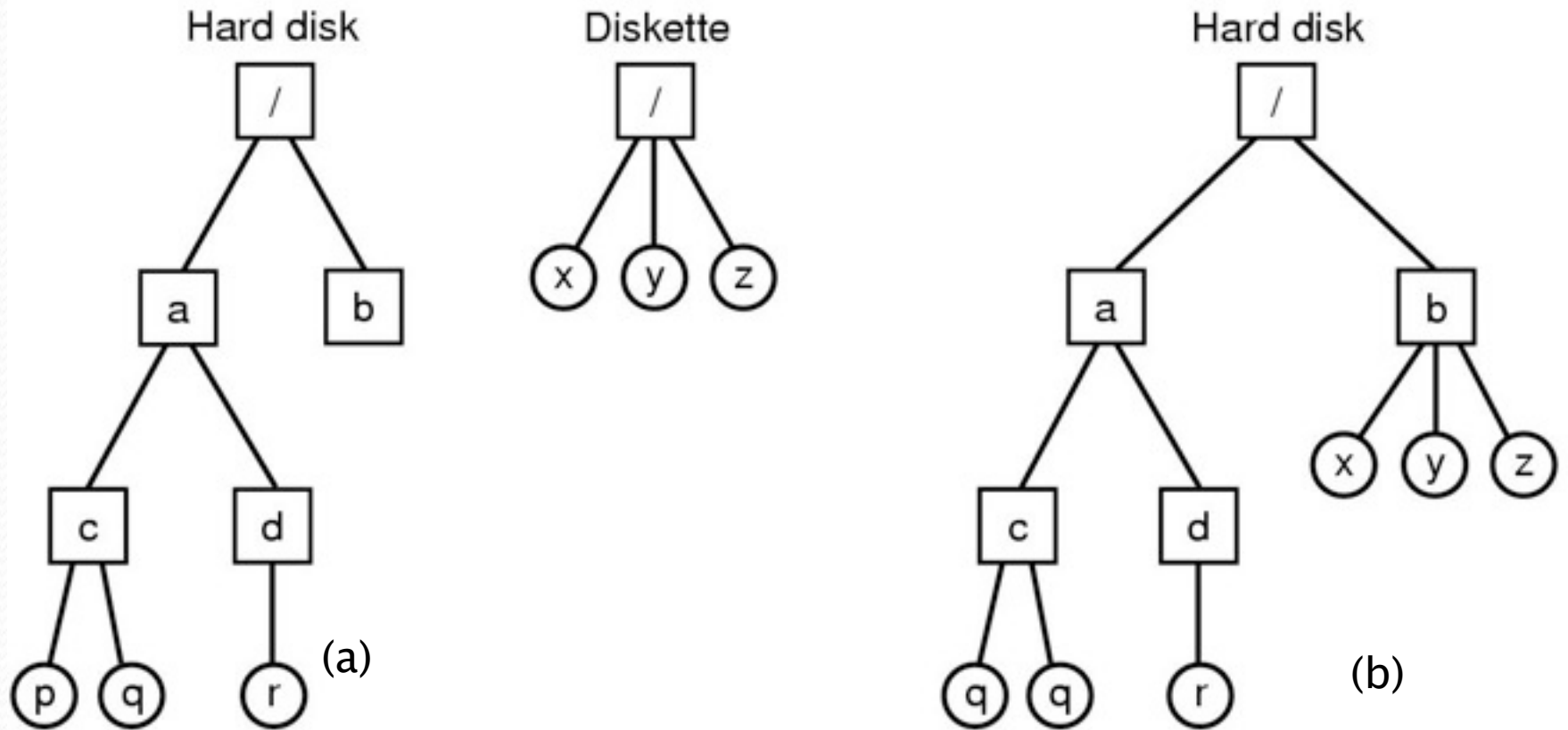
Algunos directorios importantes

El sistema de archivos UNIX (2)



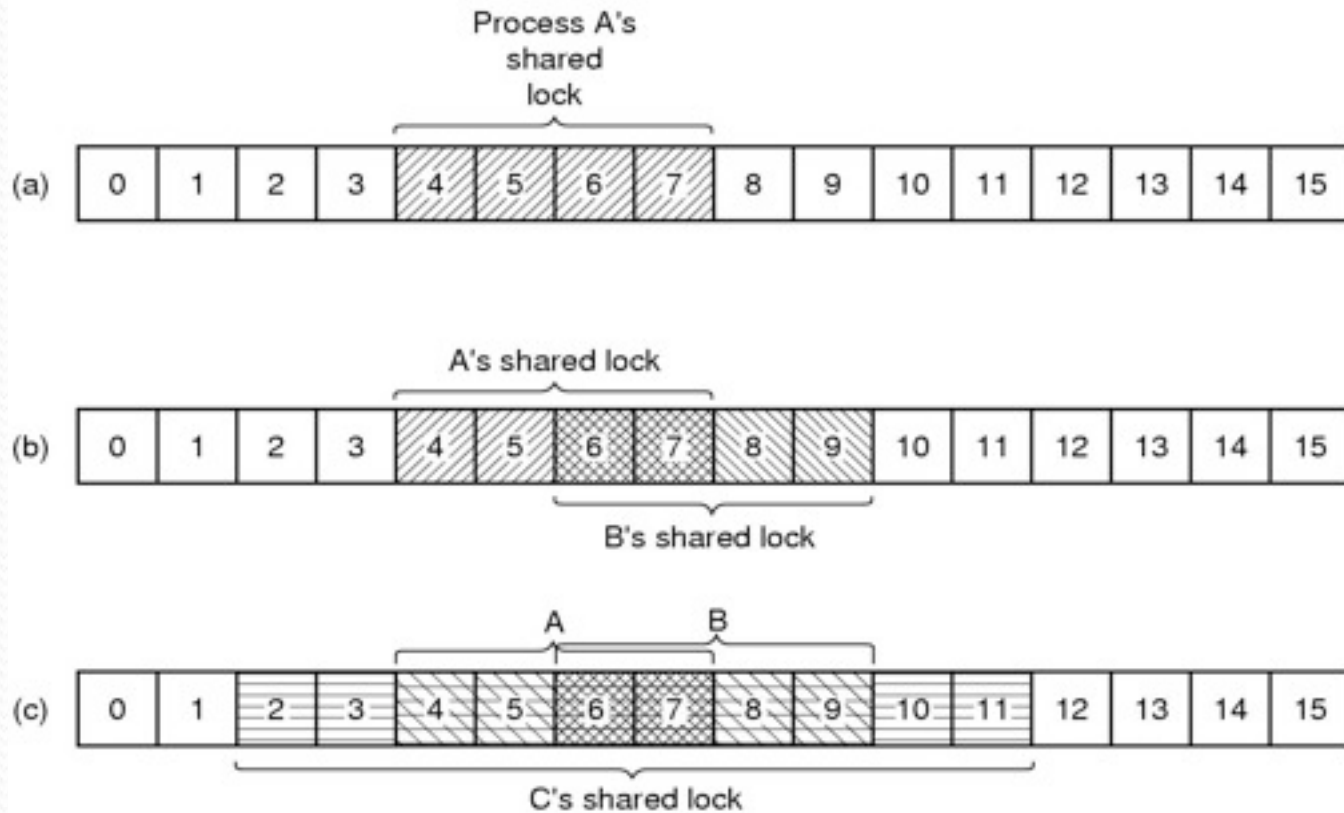
(a) Antes de enlazar. (b) Después de enlazar

El sistema de archivos UNIX (3)



(a) Antes de mount (b) Después de mount

Bloqueando archivos



(a) Archivo con un bloqueo

(b) Adición de un segundo bloqueo

(c) Un tercer bloqueo

Llamadas al sistema para administración de archivos

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

- **s** es un código de error
- **fd** es un descriptor de archivo
- **position** es un desplazamiento

La llamada 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

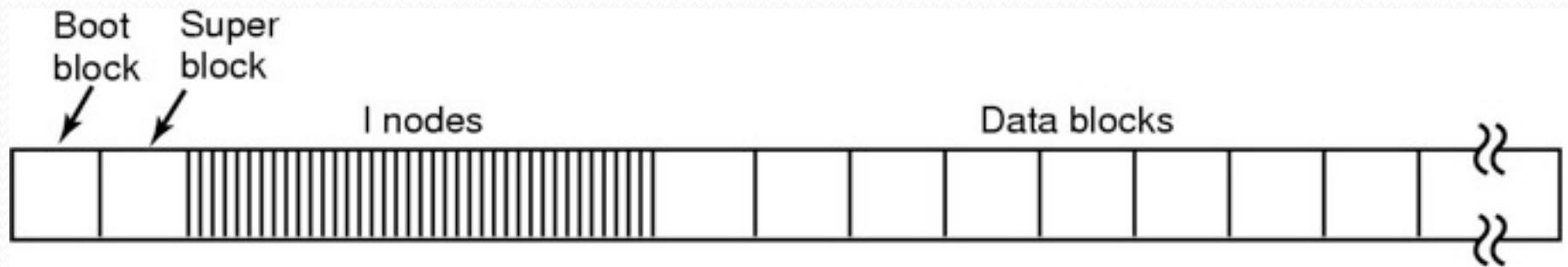
Campos retornados por la llamada stat

Llamadas al sistema para administración de directorios

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

- **s** es un código de error
- **dir** identifica el directorio
- **dirent** es una entrada de directorio

Implementación del sistema de archivos (1)



Organización del disco en sistemas UNIX

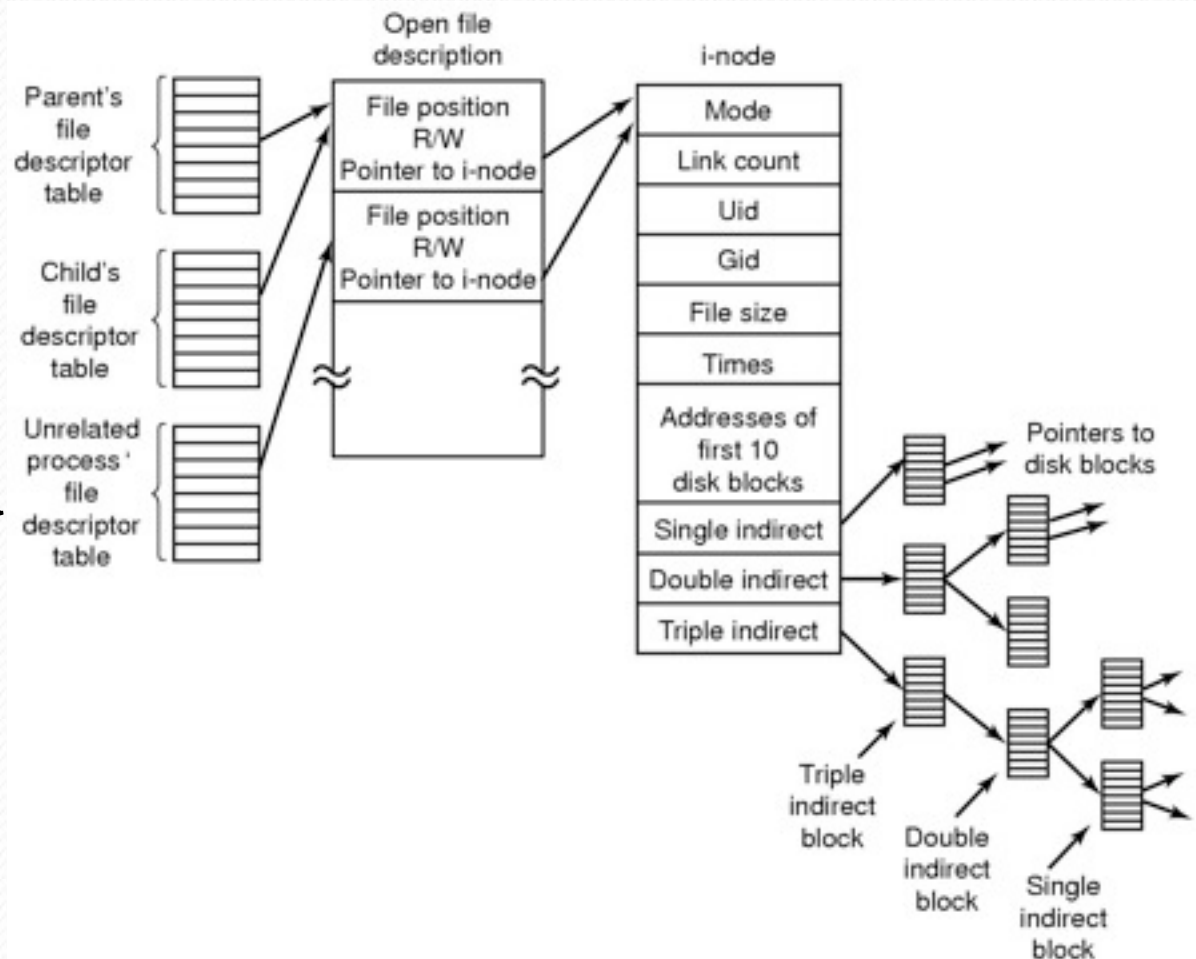
Implementación del sistema de archivos (2)

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)

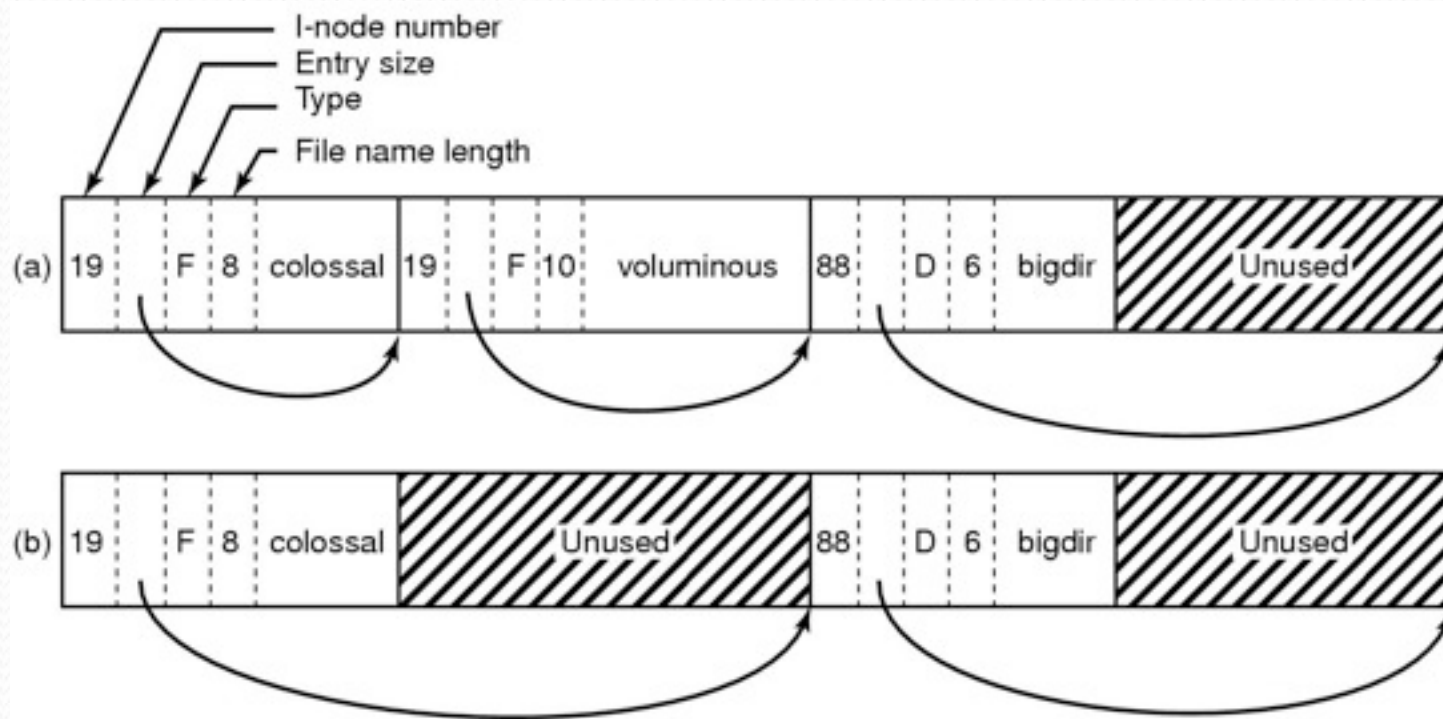
Estructura de un nodo-i

Implementación del sistema de archivos (3)

Relación entre la tabla de descriptores de archivos y la descripción de archivo abierto

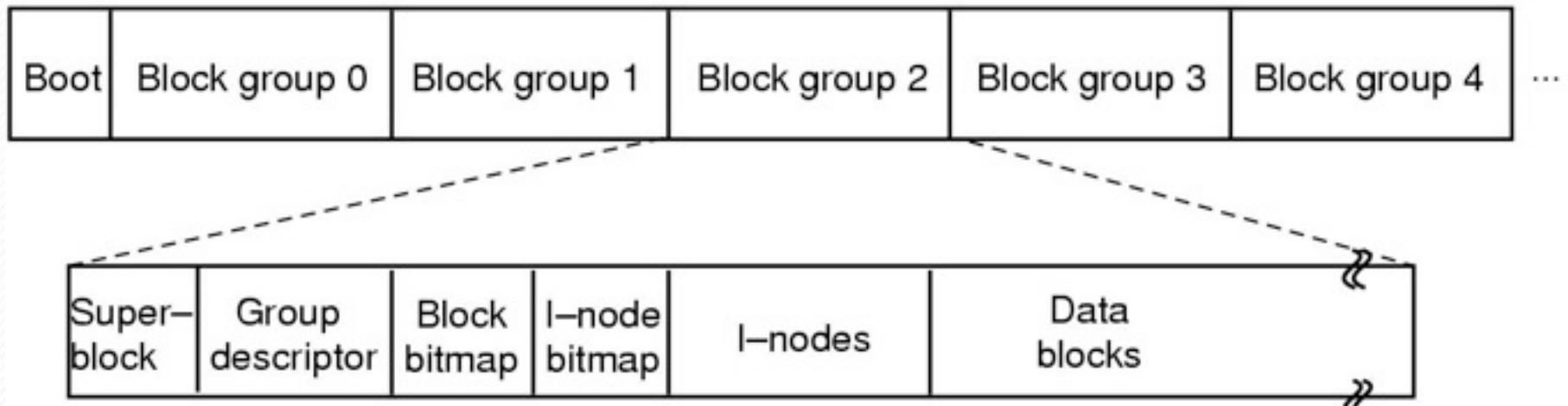


Implementación del sistema de archivos (4)



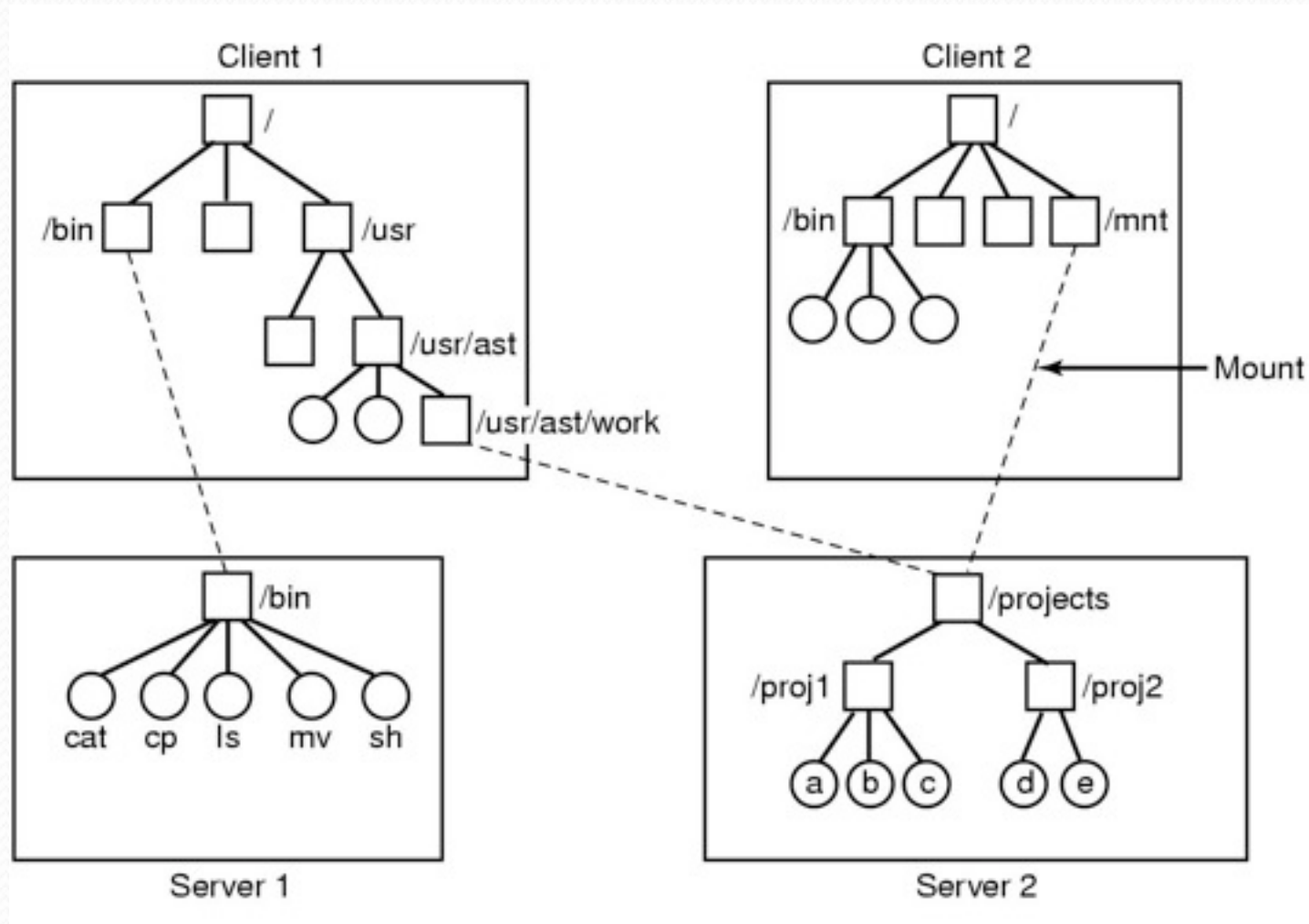
- Un directorio BSD con tres archivos
- El mismo directorio después que el archivo voluminous ha sido eliminado

El sistema de archivos Linux



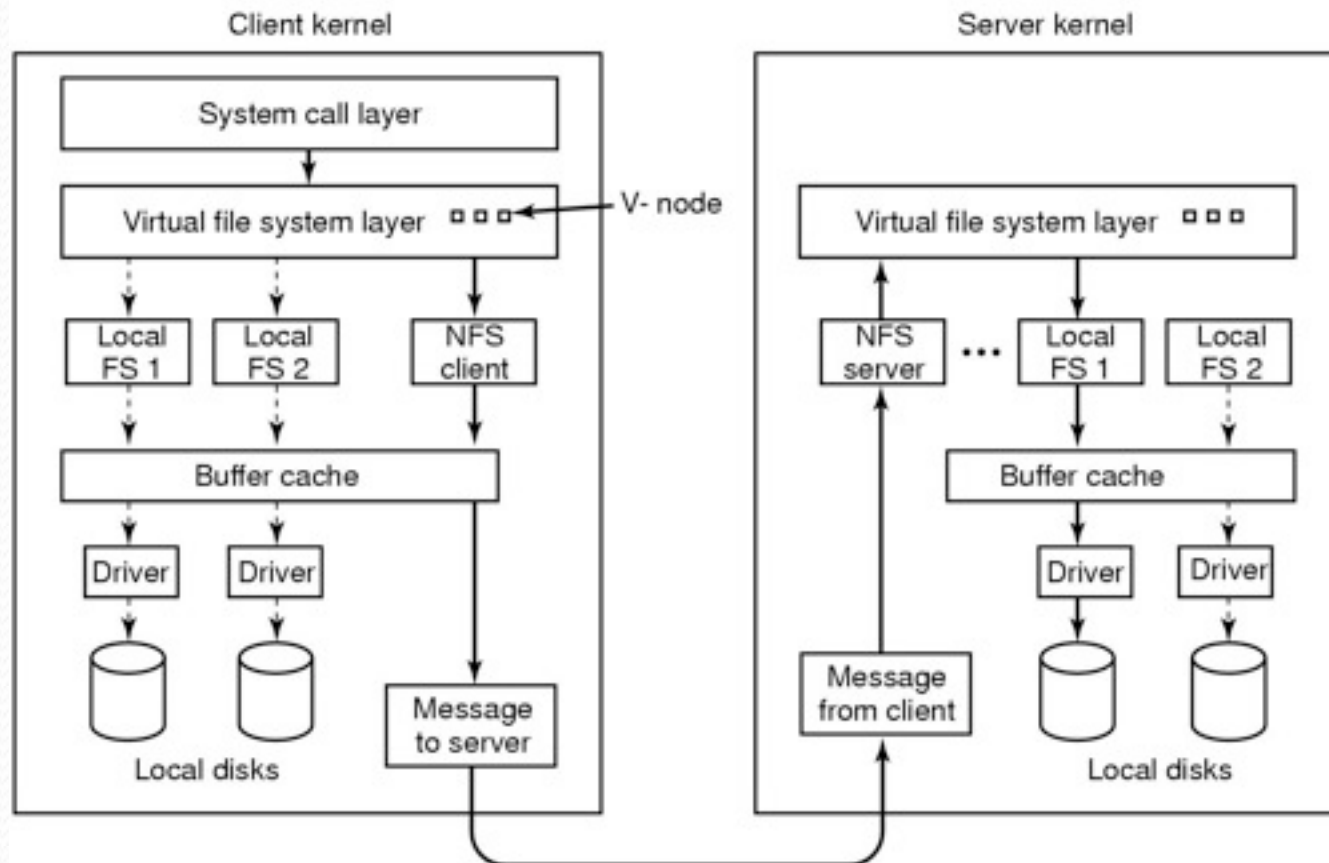
Linux Ex2 file system

Network File System (1)



Ejemplo de sistemas de archivos remoto

Network File System (2)



La estructura de NFS

Seguridad en 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)

Algunos ejemplos

Llamadas al sistema para la protección de archivos

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

- **s** es un código de error
- **uid** y **gid** son el UID y GID, respectivamente

Resumiendo

- El acceso a dispositivos se realiza utilizando archivos especiales
 - La E/S en dispositivos de bloques utiliza caché
 - La E/S por caracteres puede utilizarse en modo crudo o cocinado
- El sistema de archivos es jerárquico
 - Todos los discos se montan en un solo árbol de directorios
 - Se pueden establecer enlaces
- La protección se basa en tres grupos de bits