Sistemi Operativi (Laboratorio)

Lorenzo Martignoni

Dipartimento di Informatica e Comunicazione
Università degli Studi di Milano, Italia
lorenzo@security.dico.unimi.it

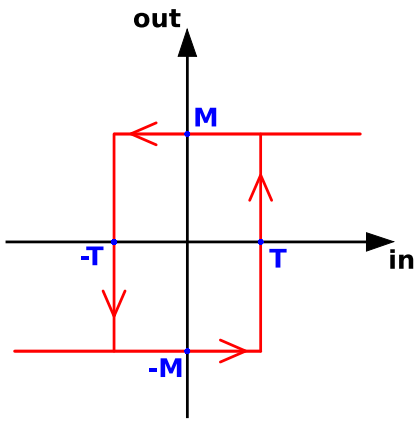
a.a. 2008/09

Lezione VIII: Memoria di massa e system call

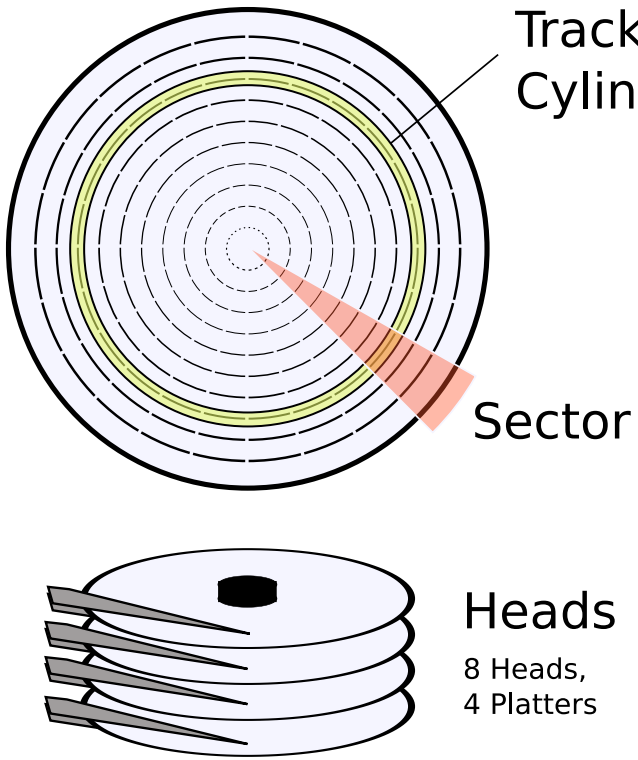
Memorie di massa

Il disco fisso (hard disk) è generalmente una memoria magnetica. Viene sfruttato il fenomeno del ciclo di isteresi di elementi magnetici (L'isteresi è la caratteristica di un sistema di reagire in ritardo alle sollecitazioni applicate e in dipendenza dello stato precedente).

Un ciclo di isteresi può essere ottenuto anche elettronicamente (Schmitt trigger). Le memorie USB, invece, sono basate su transistor NAND.



Hard disk



Gli elementi contenenti dati (blocchi fisici) sono definiti da tre coordinate:

1. **Cylinder**: Il cilindro definito dall'insieme delle tracce corrispondenti dei vari piatti
2. **Head**: La testina (per esempio, sopra e sotto)
3. **Sector**: Lo spicchio

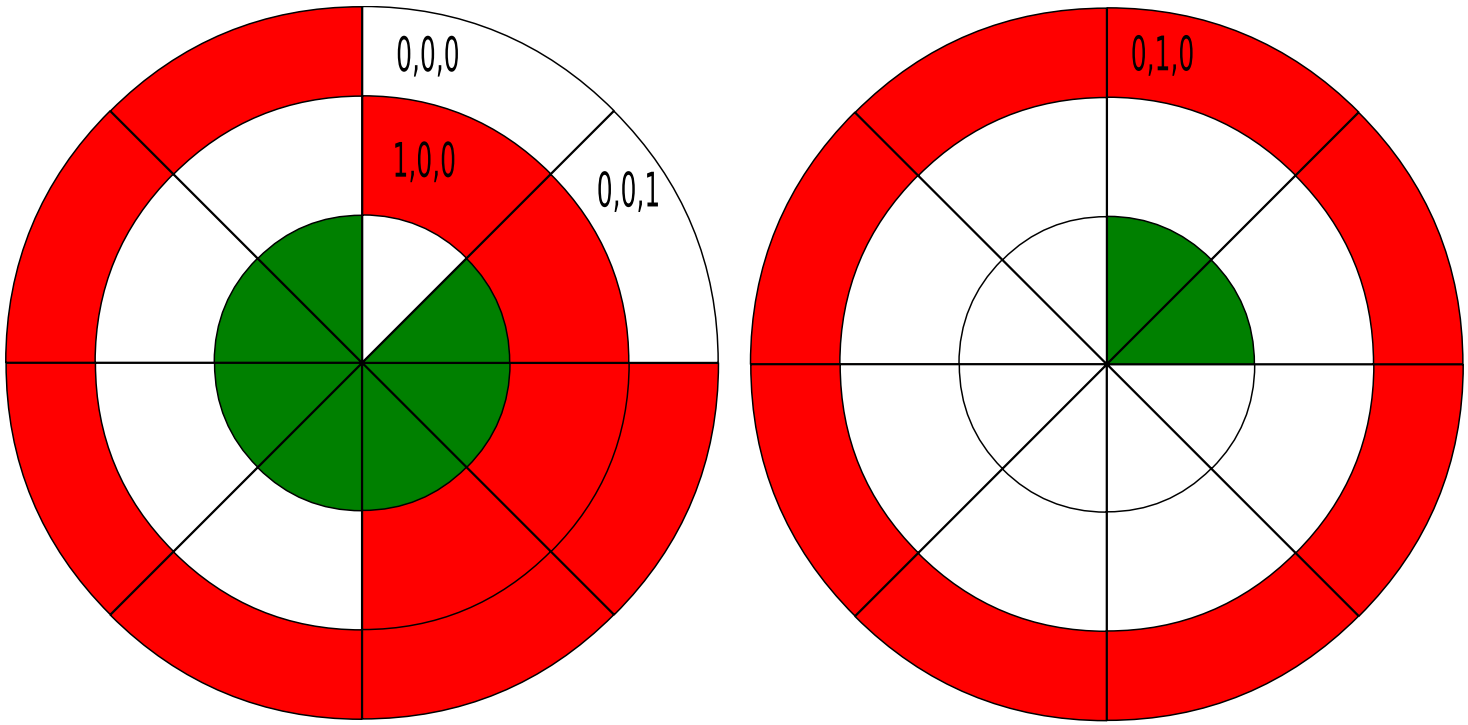
Calcolo dei blocchi

$$blocksPerPlatterSide = (cylindersPerPlatter) * (SectorsPerPlatter)$$
$$blocksPerPlatter = (blocksPerPlatterSide) * (HeadsPerPlatter)$$
$$blocksPerPlatter = (cylindersPerPlatter) * (SectorsPerPlatter) * (HeadsPerPlatter)$$
$$blocks = (Cylinders) * (Heads) * (Sectors)$$

Esempio

Un floppy disk con 80 cilindri, 2 testine, 18 settori $\rightsquigarrow$ 2880

CHS



- $C = 3$ $H = 2$ $S = 8$, totale blocchi 48
- Zona (partizione) rossa 0,0,2 $\rightsquigarrow$ 1,0,3

$$(1*(2*8)+0*8+3*1)-(0*(2*8)+0*8+2*1) = 19-2 = 17$$

(in realtà 18 perché contiamo da zero)

Tempo di lettura e scrittura

- $T = \mathit{TempoDiRotazione} + \mathit{TempoDiRicerca} + \mathit{TempoDiAccesso}$
- Il tempo di rotazione è detto anche latenza
- Il tempo di ricerca (seek time) può essere ottimizzato con algoritmi opportuni

Device logico

- L'astrazione fornita dal s.o. per il disco è quella del device a blocchi.
- Il blocco è un blocco logico, potenzialmente diverso dal blocco fisico.
- I device a blocchi sono file speciali, identificati da
  - **Major number**: identifica la categoria del device (disco IDE, floppy)
  - **Minor number**: numero d'ordine del device all'interno di una categoria

Esempio

- In MINIX il disco IDE primario è il device a blocchi: `/dev/c0d0` (3 0)
- Le partizioni sono a loro volta device a blocchi: `/dev/c0d0p0s0` (3 128), `/dev/c0d0p0s1` (3 129)

mknod

I file speciali si creano con `/usr/bin/mknod` generalmente in `/dev`

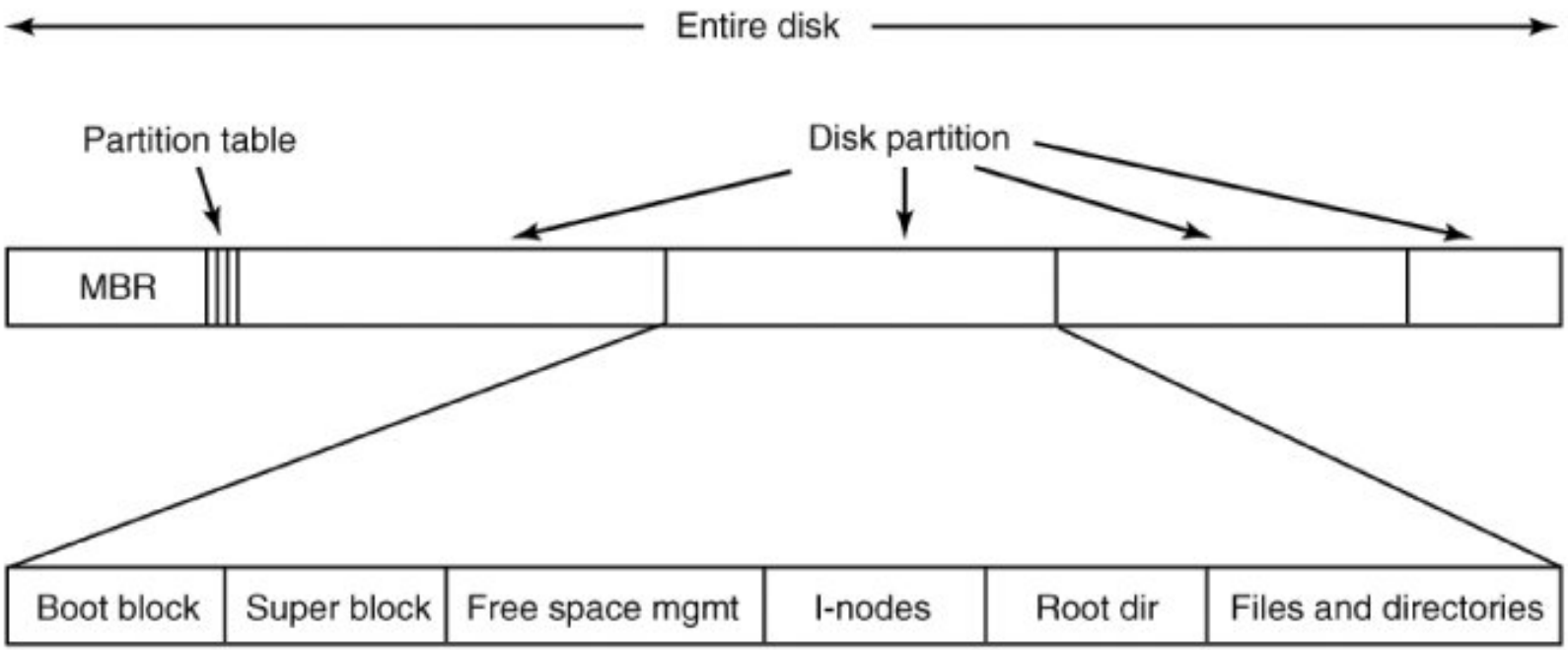
- Device a blocchi `b`
- Device a caratteri `c`
- Named pipe `p` (non ha major e minor)

Partizioni

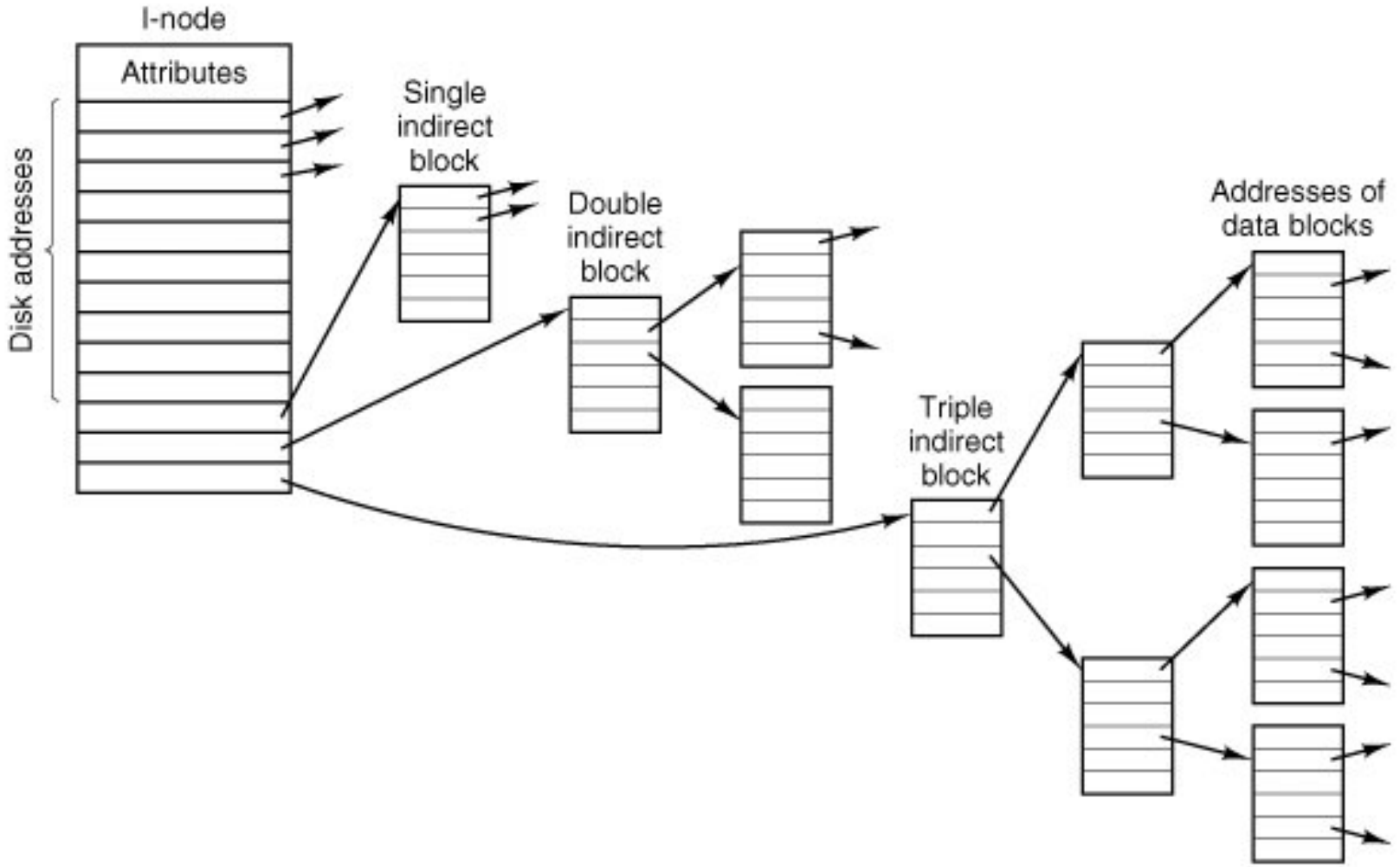
Lo spazio di memoria di uno hard-disk è ripartito in porzioni indipendenti (partizioni): in linea di principio possono contenere anche sistemi differenti. Generalmente contengono sotto-file-system il cui backup e/o aggiornamento è indipendente.

- Partition table sector: contiene la descrizione di 4 partizioni (primarie) agli offset 446, 462, 478, 494
- Partizione: una zona contigua del disco (CHS)
- Partizione estesa: una partizione che permette una nuova suddivisione (partizioni logiche) grazie ad un nuovo PTS

Disk layout


Entire disk
Partition table
Disk partition
MBR
Boot block
Super block
Free space mgmt
I-nodes
Root dir
Files and directories

I-node


I-node
Attributes
Disk addresses
Single indirect block
Double indirect block
Triple indirect block
Addresses of data blocks

Creare e usare un fs

- Un file system va creato (`mkfs`)
- Un file system va montato (`mount`)
- Corrispondentemente va smontato (`umount`)
- Ogni file è caratterizzato da un i-node e conosciuto tramite uno o piú link o nomi (`ln`)

i-node speciali

- Directory (`mkdir`)
- Link simbolici (`ln -s`)

Programmi utili per lavorare sui nomi o percorsi

- `dirname`
- `basename`

Programmi utili per lavorare sugli i-node

- `stat`
- `readlink`

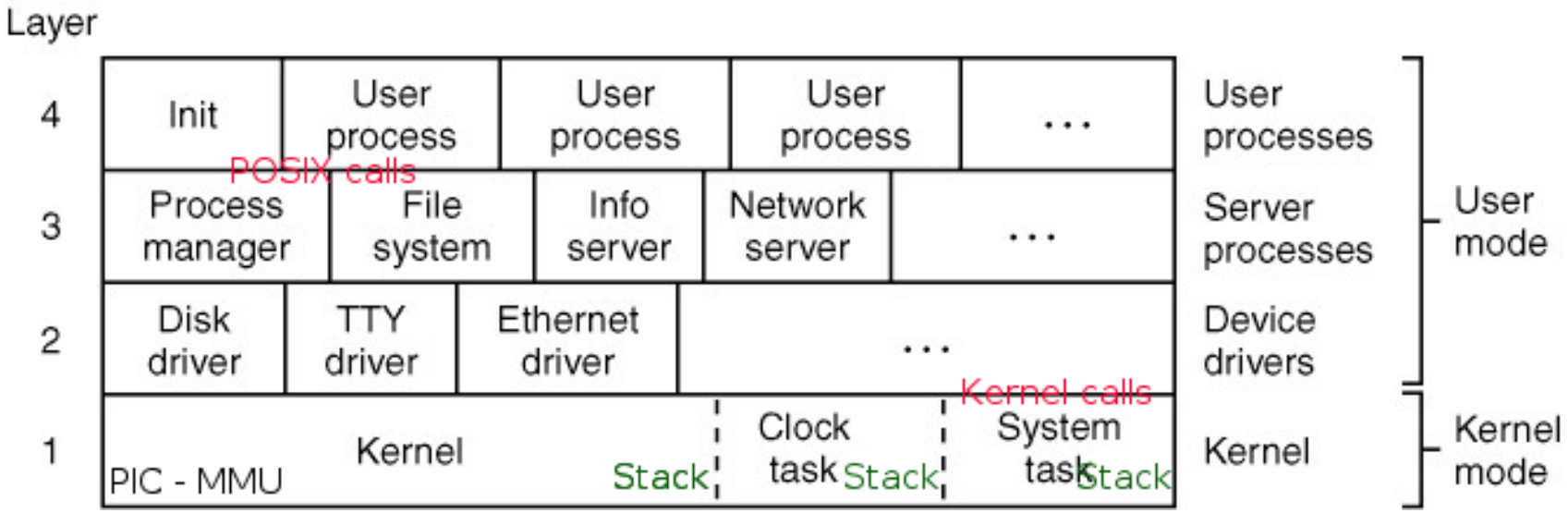
Esercizio

1. Creare una directory
2. Creare un link simbolico al file con il database degli utenti
3. Creare un link (hard) al file con il database dei gruppi
4. Analizzare e discutere l'output di `ls` e `stat` (con e senza il parametro `-s`)
5. Creare un archivio `tar`, analizzare e discutere il contenuto dell'archivio

Esercizio

1. Creare un disco virtuale (`qemu-img`) e connetterlo alla macchina virtuale
2. Partizionare il disco (`part`)
3. Creare il file system (`mkfs`)
4. Montare il file system (`mount`)
5. Copiare dei dati
6. Smontare il file system (`umount`)

MINIX


Layer
4
3
2
1
Init
User process
User process
User process
...
POSIX calls
Process manager
File system
Info server
Network server
...
Disk driver
TTY driver
Ethernet driver
...
Kernel calls
PIC - MMU
Kernel
Stack
Clock task
Stack
System task
Stack
User processes
Server processes
Device drivers
Kernel
User mode
Kernel mode

Come aggiungere una system call `foo()`

Supponendo che il servizio sia gestito gestito da PM

1. Aggiornare `/usr/src/include/minix/callnr.h`
2. Aggiungere la entry `do_foo` al `call_vec` di PM
3. Aggiungere la entry `no_sys` al `call_vec` di FS
4. Aggiungere `do_foo()` (p.es. in `pm/misc.c`)
5. Creare la libreria wrapper e aggiornare `unistd.h`

`http://www.cis.syr.edu/~wedu/seed/Documentation/Minix3/How_to_add_system_call.pdf`

Esercizio

- Aggiungere una syscall che stampa “ciao”
- Aggiungere una syscall che stampa il numero di processi nella coda ready
- Aggiungere una syscall che modifica il proprietario di un processo in esecuzione

Data server

`/usr/src/servers/ds`

Copyright