Sistemi Operativi

Lezione 12
Il File System

I Dischi

Alcune delle fotografie riportate sono riprese da:

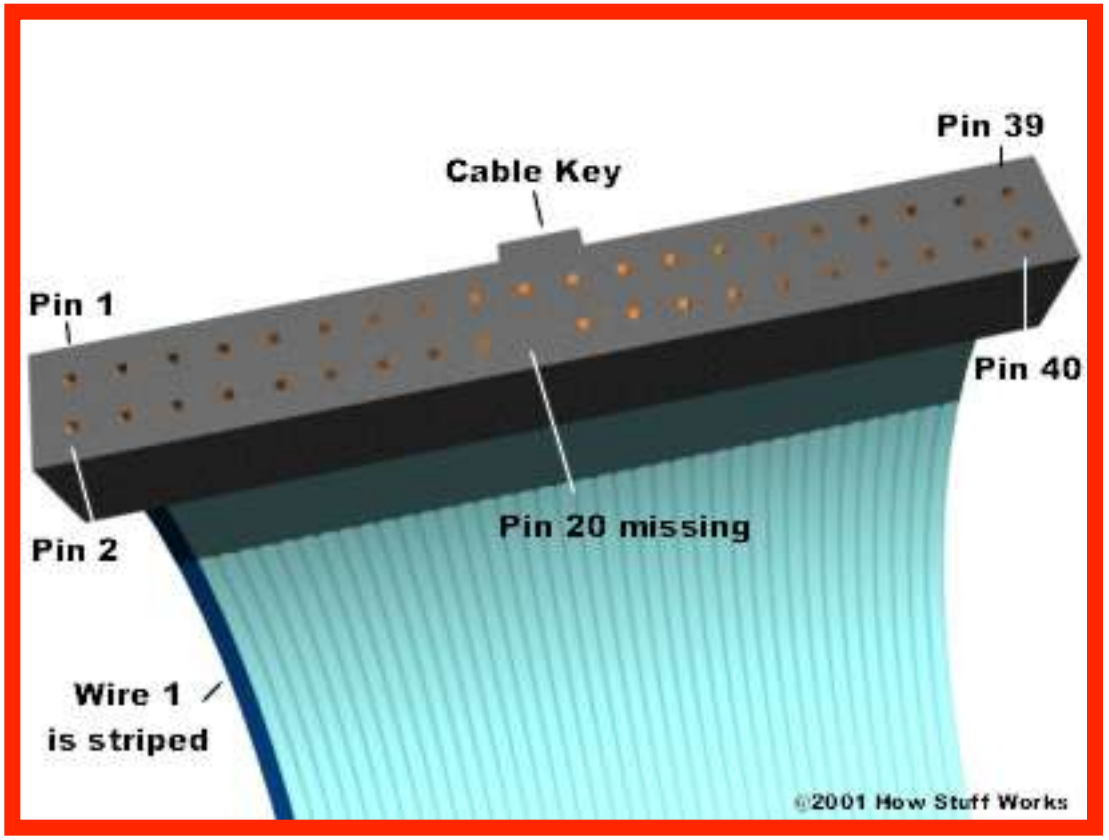
http://royal.pingdom.com/2010/02/18/amazing-facts-and-figures-about-the-evolution-of-hard-disk-drives/

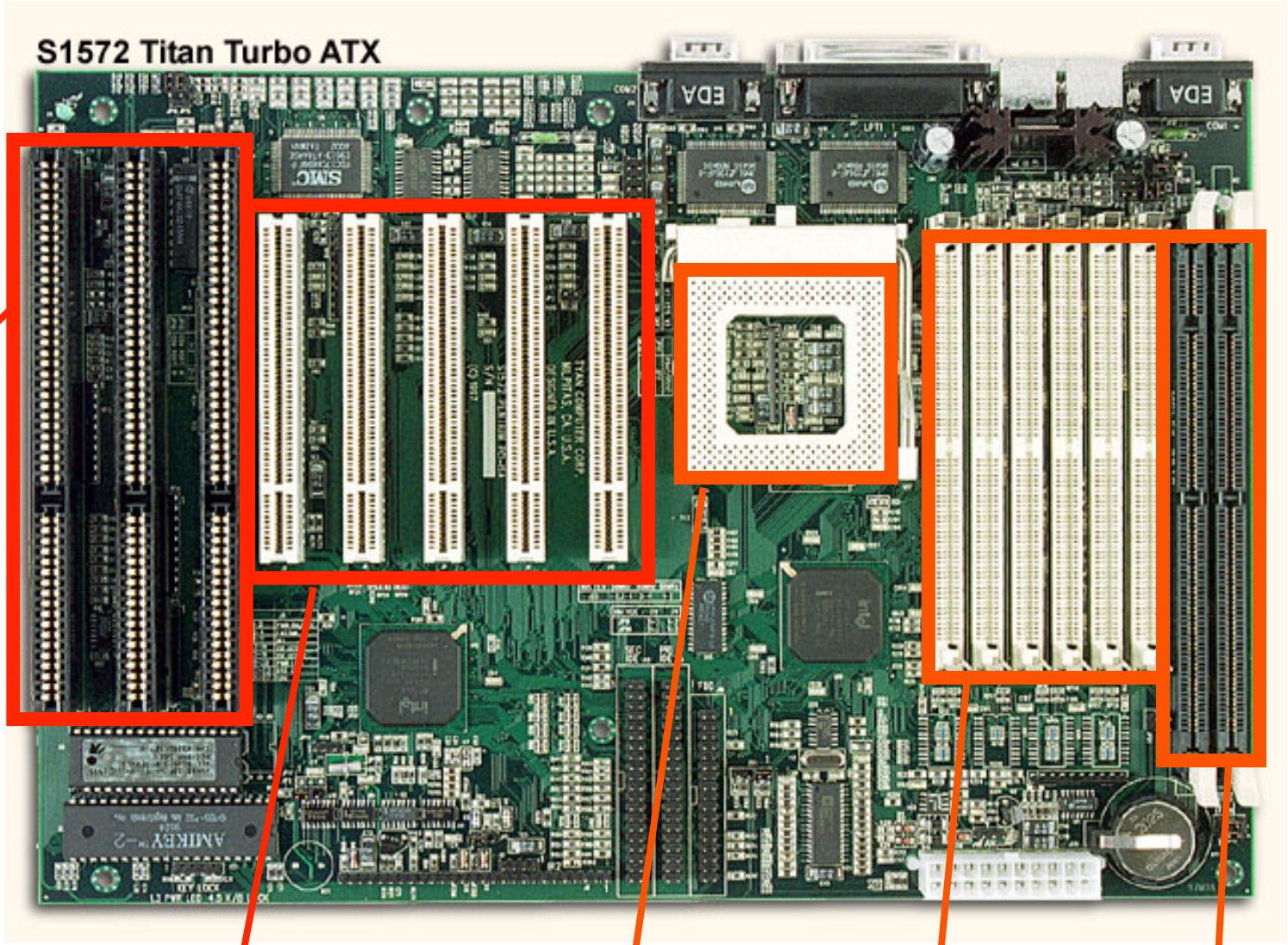
Hard Disk

Disk Head

Controller (IDE)

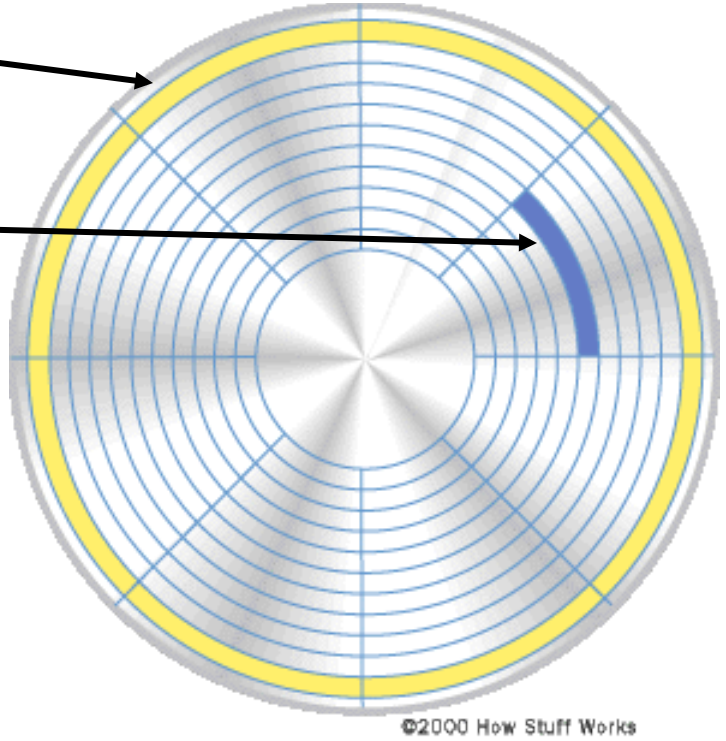
IDE Cable


Pin 39
Cable Key
Pin 1
Pin 40
Pin 20 missing
Pin 2
Wire 1
is striped
©2001 How Stuff Works

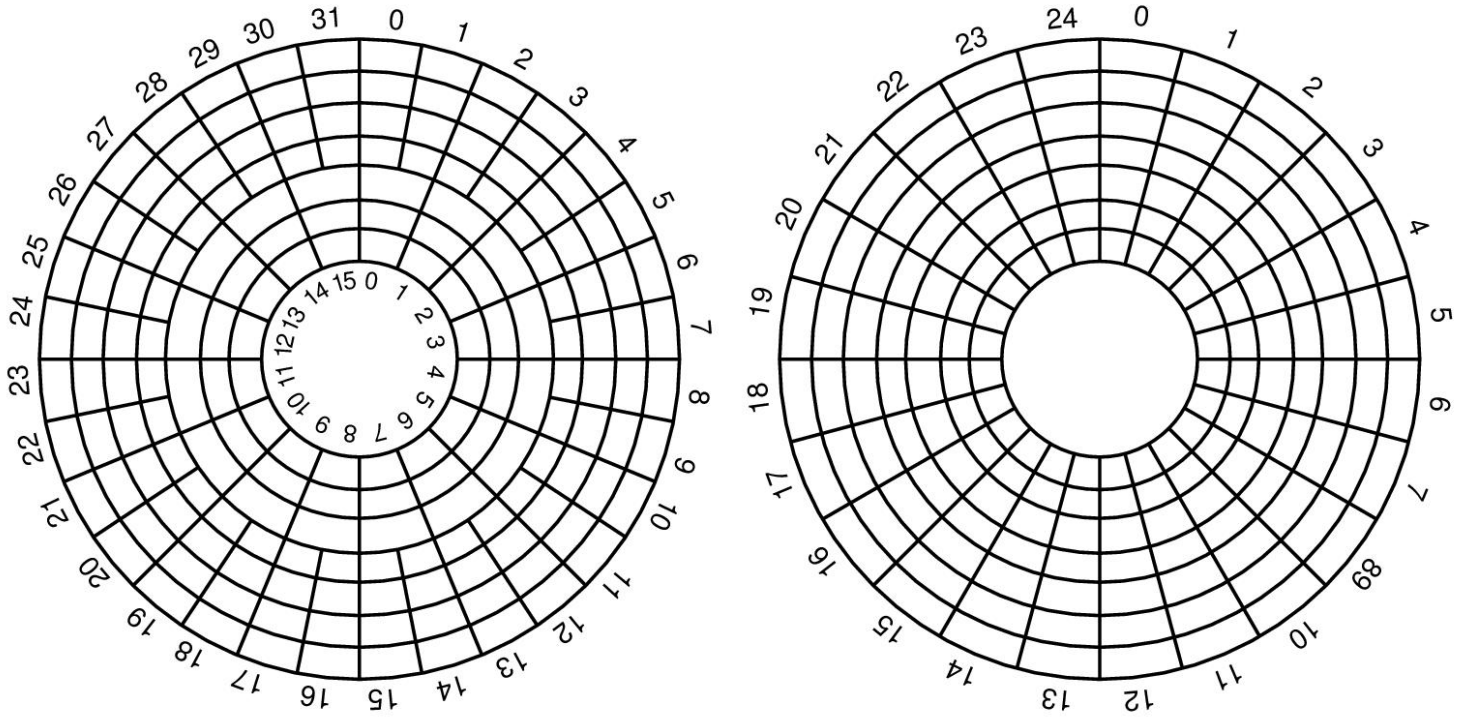

S1572 Titan Turbo ATX
3 ISA slots
5 PCI slots
Pentium CPU
6 SIMM slots
2 DIMM slots

Tracce e Settori

- Ogni superficie è suddivisa in un insieme di Tracce
- Ogni traccia è suddivisa in settore

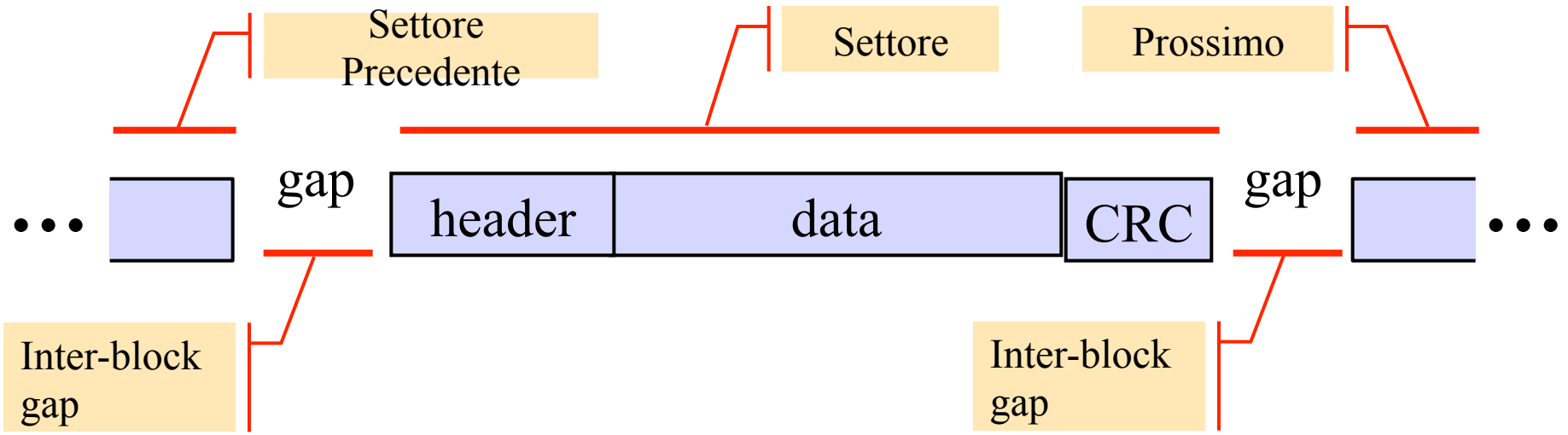


Tracce



- Struttura fisica di un disco con doppio settore sulle tracce esterne
- Struttura logica dello stesso dispositivo

Formato di una traccia

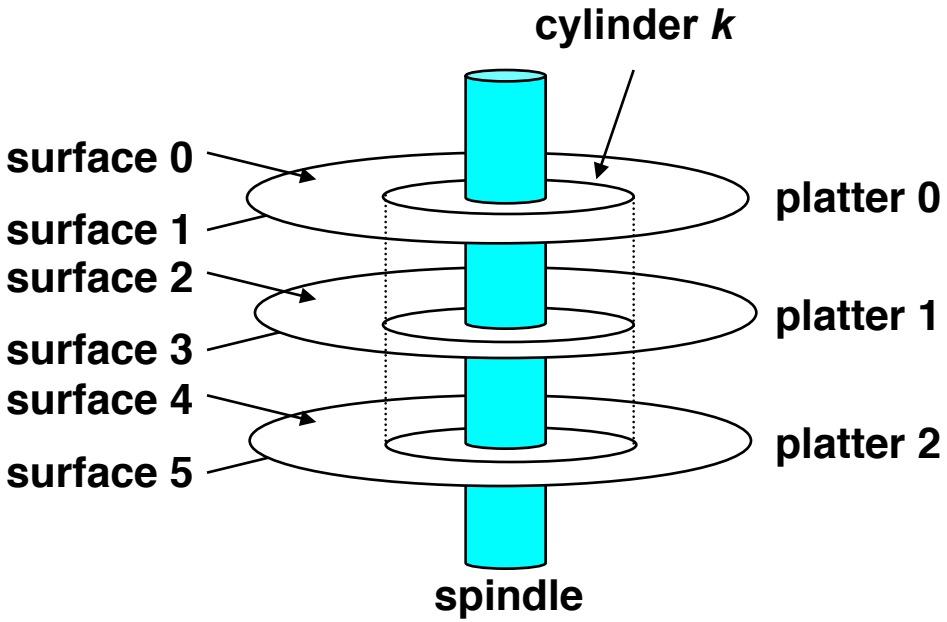


Note:

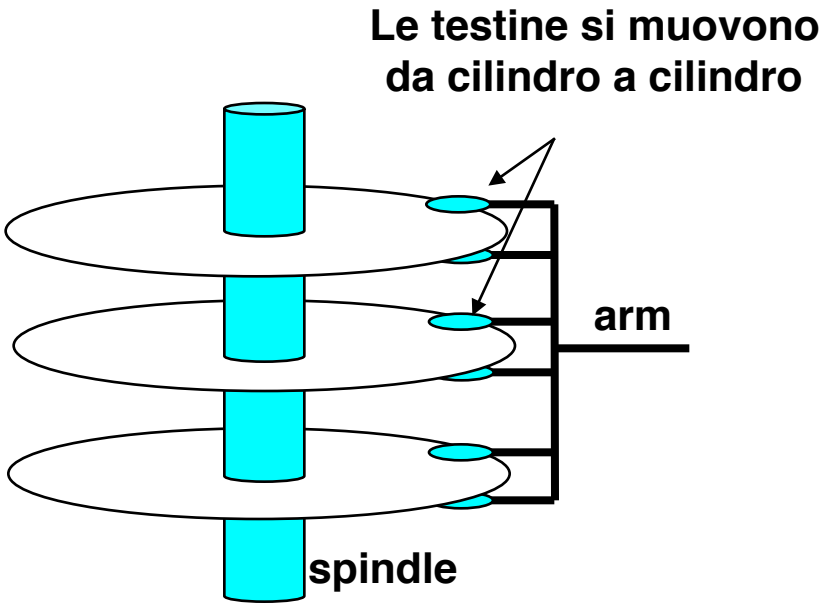
CRC è acronimo di "cyclic redundancy check". Un codice correttore relativo al campo data, situato alla fine di ogni settore.

Cilindri

- Tutte le tracce con lo stesso numero costituiscono un cilindro



Cilindri



Disk Hardware

Parameter	IBM 360-KB floppy disk	WD 18300 hard disk
Number of cylinders	40	10601
Tracks per cylinder	2	12
Sectors per track	9	281 (avg)
Sectors per disk	720	35742000
Bytes per sector	512	512
Disk capacity	360 KB	18.3 GB
Seek time (adjacent cylinders)	6 msec	0.8 msec
Seek time (average case)	77 msec	6.9 msec
Rotation time	200 msec	8.33 msec
Motor stop/start time	250 msec	20 sec
Time to transfer 1 sector	22 msec	17 µsec

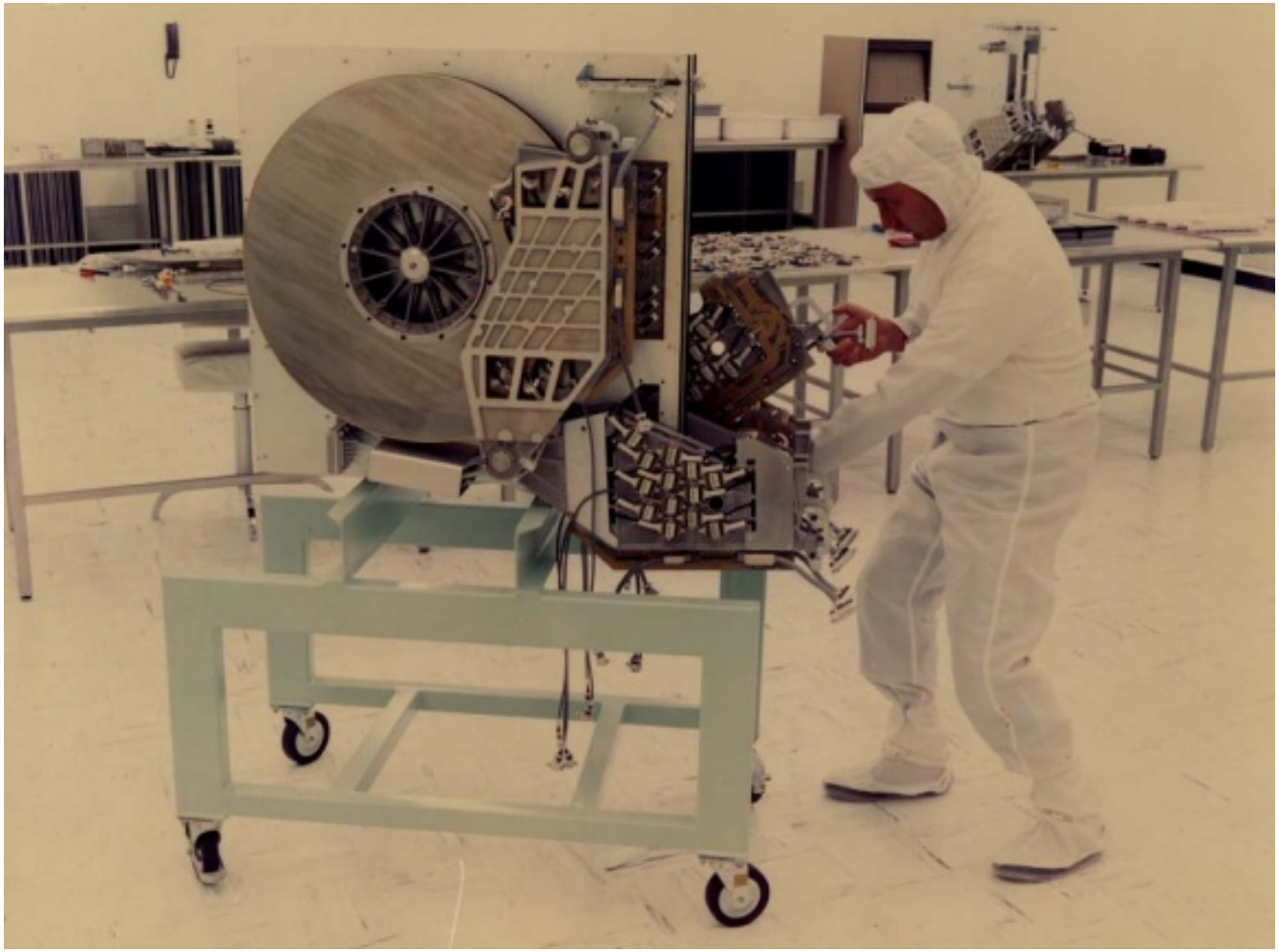
Evoluzione dei dischi a partire dal dispositivo originale del PC IBM sino al modello Western Digital WD 18300 (~2000)

Prezzi per GB (Ottobre 2009)

- 8.2¢ per GB – 1 TB Western Digital Caviar (internal)
  - 7200 rpm, 32 MB drive cache
  - 3 Gb/sec SATA-II cache to host; 145 MB/sec buffer to disk
- 8.4¢ per gigabyte – 1.5 TB Barracuda (internal)
  - 7200 rpm, 8.5 ms. avg. seek time, 32 MB drive cache
  - 3 Gb/sec SATA-II cache to host
- 63.3¢ per GB – 1 TB HP Hot Swap (internal)
  - 7200 rpm, 8,5 ms. avg. seek time, 32 MB drive cache
  - 3 Gb/sec SATA-II cache to host; 145 MB/sec buffer to disk
- $2.66 per GB – 146 GB IBM (hot swap)
  - 15,000 rpm, avg. seek time not specified
  - Ultra SCSI

1956 – Primo HD (IBM 350)- 5MB

1979 – 250 MB

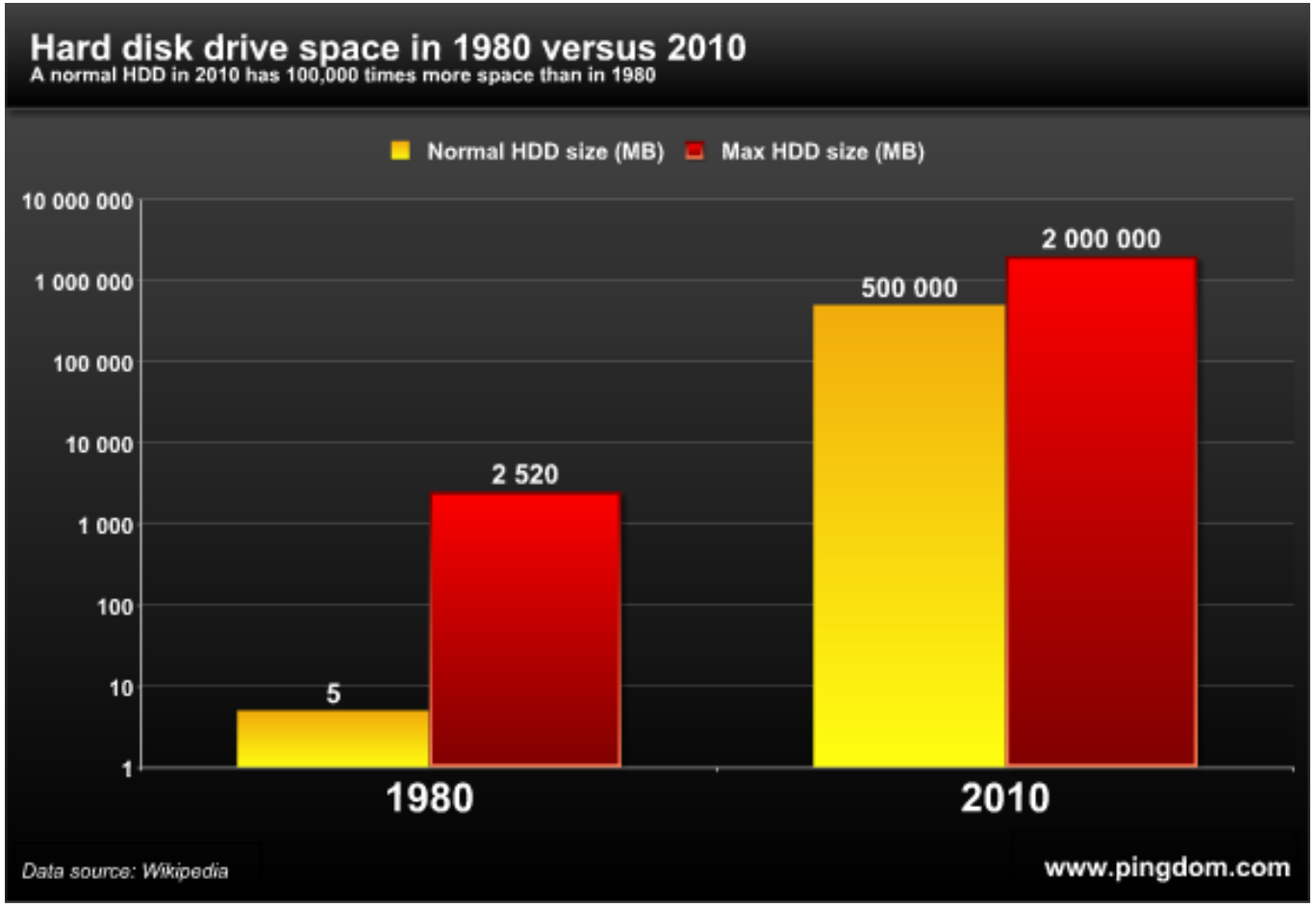


L’evoluzione

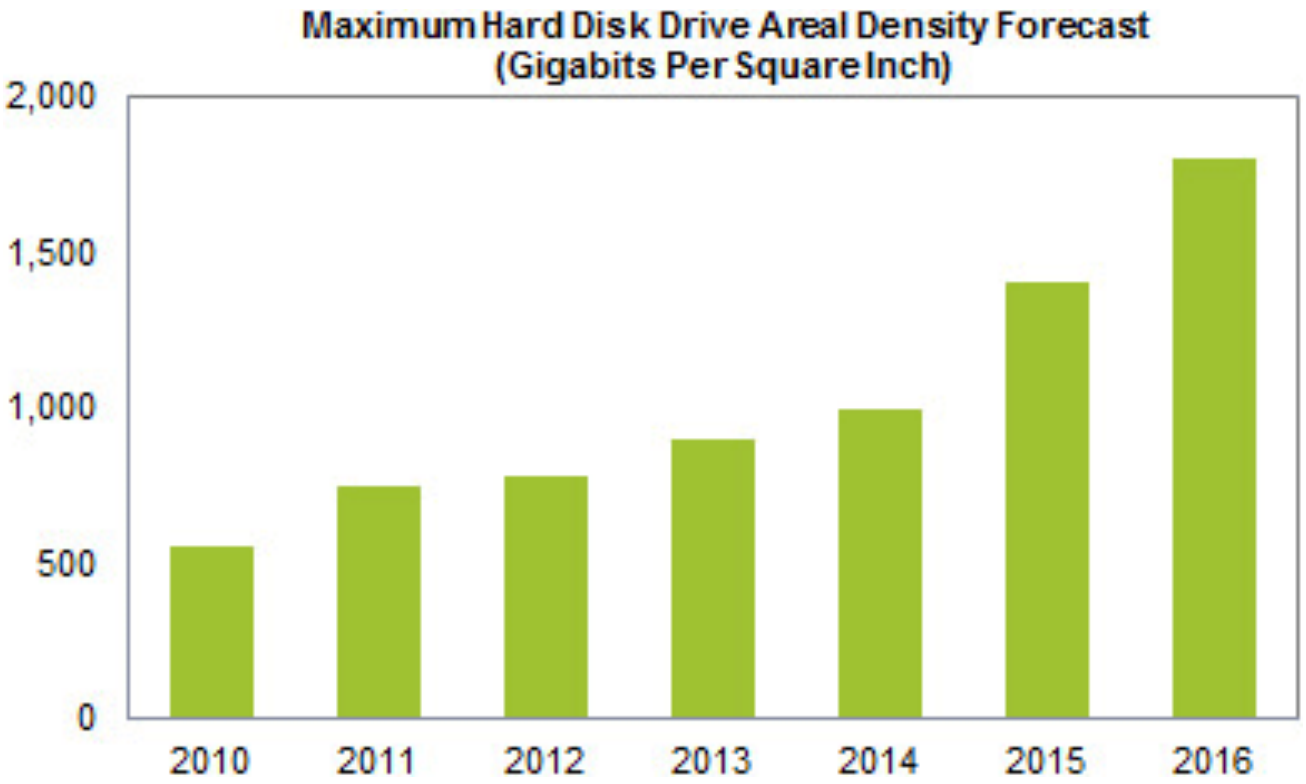
Ultima generazione (2008)

1GB, 16 mm in altezza, 16 grammi

L'evoluzione



Areal Density



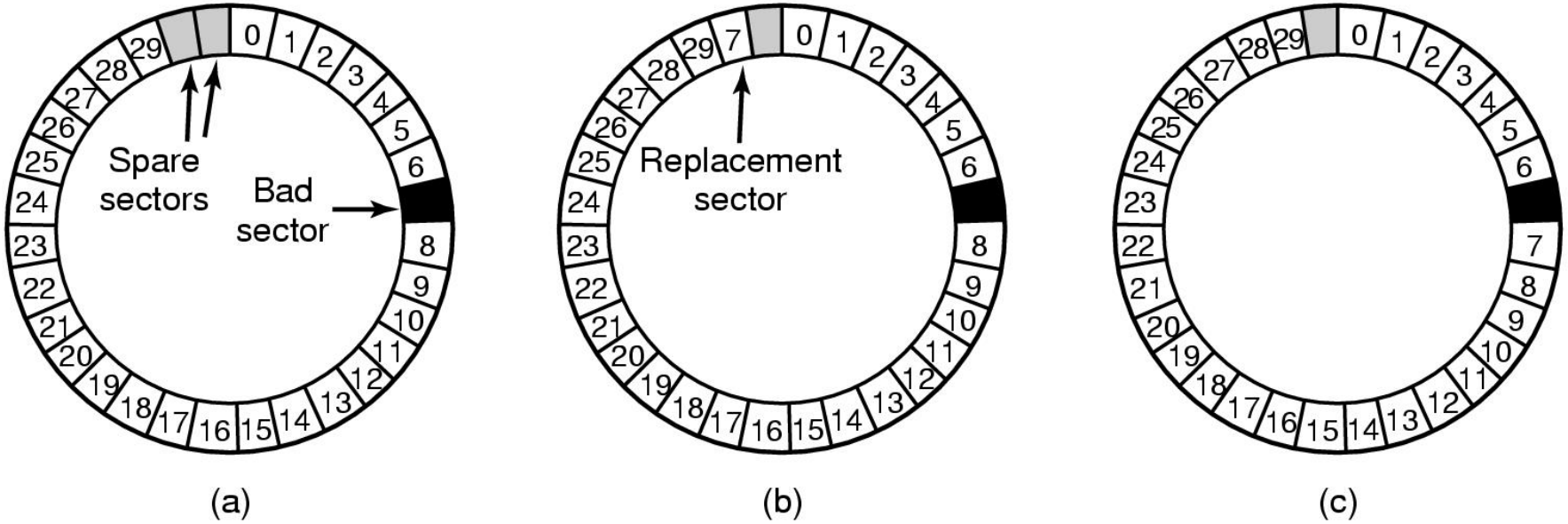
Errori

- Nella gestione di un disco si possono verificare diverse tipologie di errore, alcune di queste possono essere risolte dal controller, in altri casi è il driver che deve trovare una soluzione
  - **Errori di programmazione:** invio di parametri errati (n.ro cilindro, n.ro settore ecc.) al controller. In questi casi l'operazione va interrotta sperando che non si verifichi troppo frequentemente
  - **Transient checksum error:** errori di lettura/scrittura a causa di tracce di polvere. In questi casi si tratta di riprovare un certo numero di volte
  - **Permanent checksum error:** il settore viene marcato come cattivo (bad). Solo l'hw può porre rimedio a questa situazione

Errori

- **Seek errors:** il braccio non è ben calibrato e non si posiziona correttamente sulle tracce. È necessario avviare un'operazione per ricalibrare il braccio, quando il controller lo rende possibile
- **Controller error:** anche in questo caso va avviata un'operazione di reset del controller, che può comunque risultare non risolutiva

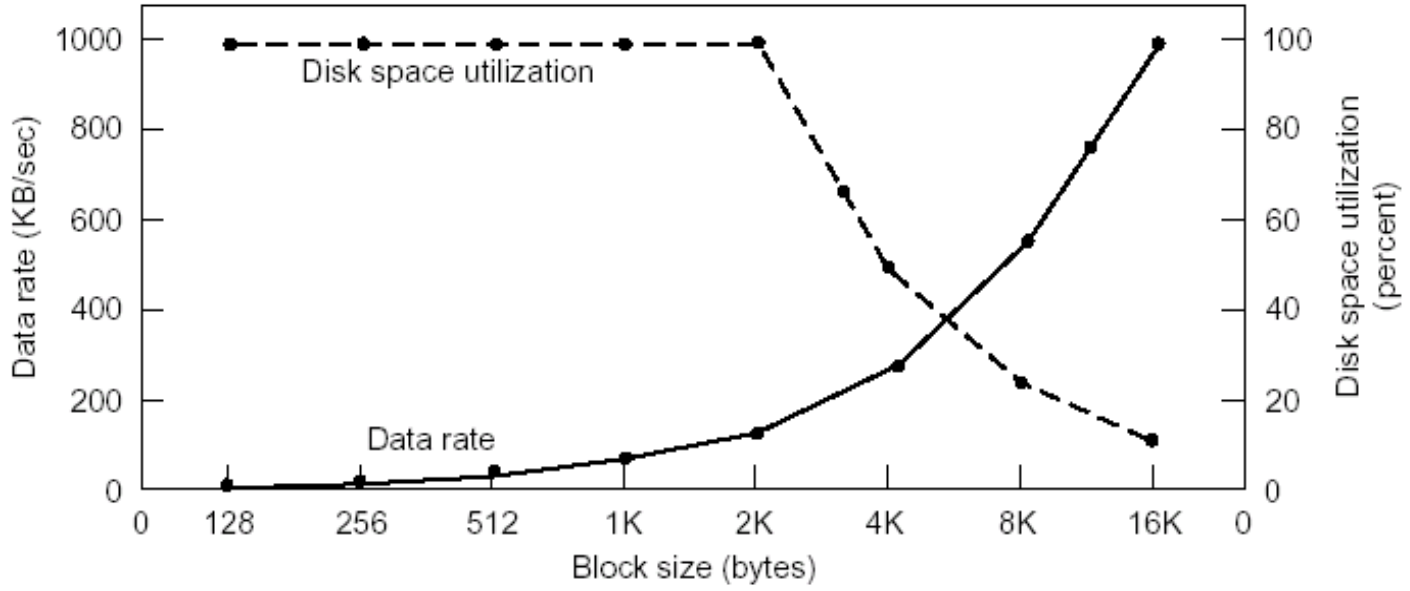
Bad Block



a. Una traccia con settore rovinato
b. Sostituzione di un settore disponibile con il contenuto del settore rovinato
c. Shifting di tutti i settori per bypassare quello rovinato

Dimensione ottimale blocco

- Esiste un trade-off, tra spreco dello spazio e velocità di trasferimento in base alla dimensione del blocco fisico



R.A.I.D.
Redundant array of inexpensive disks

- Un tipo di unità dischi che utilizza una combinazione di due o più due drive per migliorare la resistenza ai guasti e le prestazioni
- Usata frequentemente nei sistemi server, molto meno sui PC
- Si distinguono diversi livelli di R.A.I.D.

Redundant Arrays of Disks

- I file sono suddivisi in più “stripe” di una certa dimensione prefissata
- Ogni stripe è memorizzata su un diverso disco
- L’incremento del numero di dischi diminuisce l’affidabilità dell’intero sistema
- La disponiblità viene aumentata attraverso la ridondanza:
  - Quando un disco si rompe il suo contenuto viene costruito usando i dati ridondati

Affidabilità (reliability)

- Affidabilità (Reliability) si misura come Mean Time To Failure (MTTF) ed indica il grado di resistenza ai guasti di un dispositivo
- Affidabilità di N dischi =
  - Affidabilità di 1 disco / N

    50,000 ore / 70 dischi = 700 ore
- MTTF per l'intero sistema passa da 6 anni ad 1 mese
- Disk array troppo poco affidabili per poter essere usati, senza particolari accorgimenti

Disponibilità (Availability)

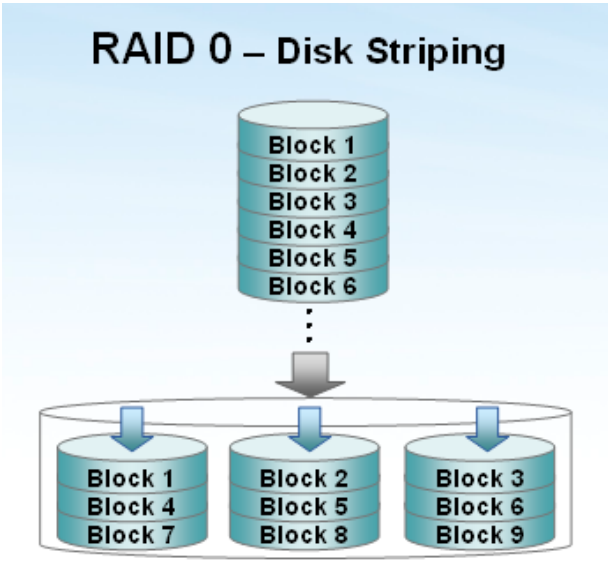
- Disponibilità: misura il livello con cui un servizio viene erogato ad un utente anche in caso di compromissione di qualche componente
- NEL RAID è garantita attraverso la ridondanza
  - Capacity penalty: per memorizzare i dati ridondati
  - Bandwidth penalty: per aggiornare i dati ridondati

R.A.I.D. (0 - 1)

- **Livello 0**
  - Data striping su più dischi
  - Nessuna ridondanza
  - Migliora le prestazioni ma non ha effetto sulla fault-tolerance
- **Livello 1**
  - Data mirroring: (a.k.a.: “shadowing”)
  - I dati sono scritti contemporaneamente su due dischi diversi
  - Se un disco si guasta, il sistema “si sposta” automaticamente sul secondo senza alcuna perdita di dati o di qualità del servizio
  - Grosso impatto sulla fault tolerance, poco impatto sulle prestazioni

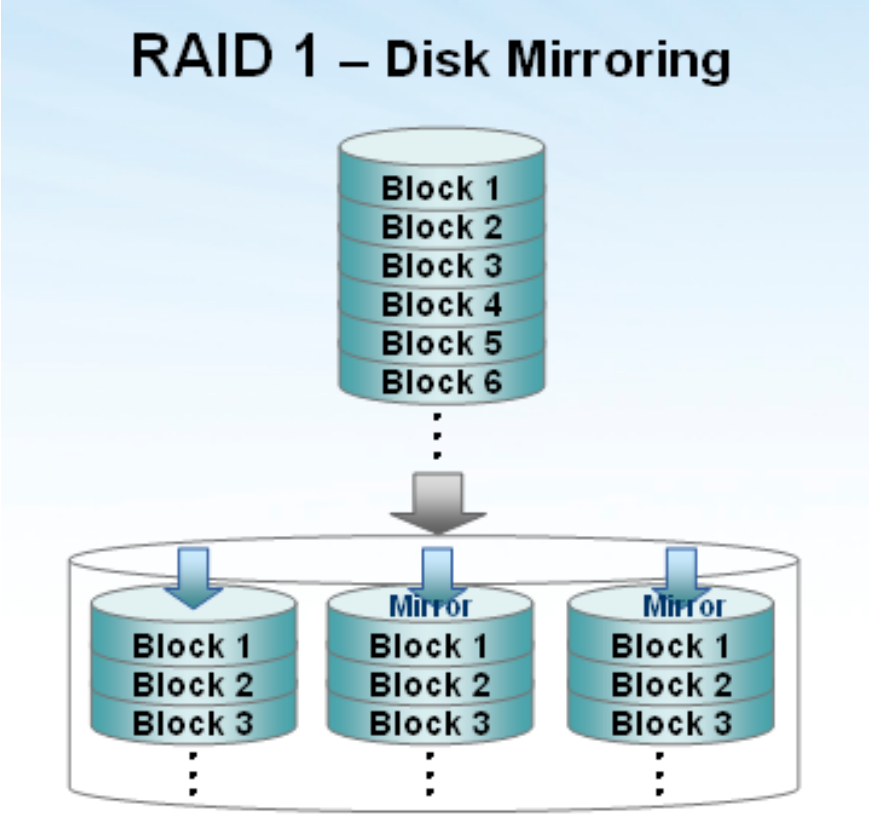
“RAID 0”: No redundancy

- Esempio: 3 dischi organizzati in 3 stripe
- Gli accessi a grosse moli di dati sono più veloci perchè si accede a 3 dischi contemporaneamente



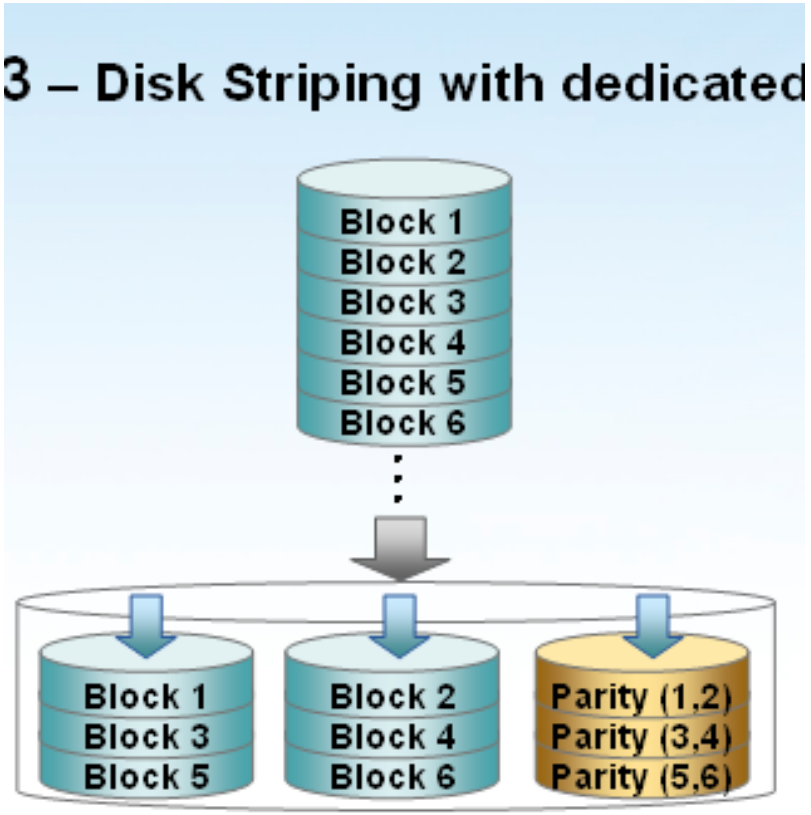
RAID 1: Mirror data

- ogni disco è duplicato sul suo mirror
  - Soluzione a disponibilità molto alta
- Banda ridotta in scrittura:
  - 1 write logica = 2 write fisiche
- Soluzione costosa: 100% di overhead sulla capacità



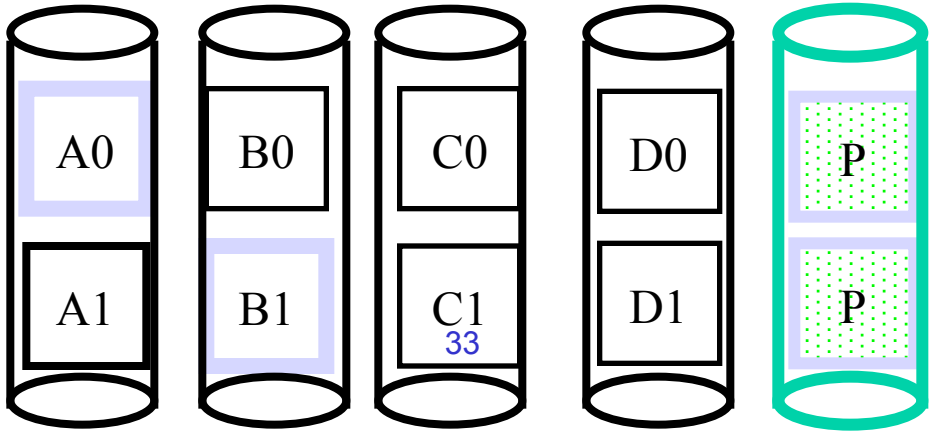
RAID 3/4: Parity

- Raid3 stripe = Byte
- Raid4 stripe = sequenza byte
- Per ogni stripe presente sui dischi dell'array si calcola la rispettiva parità che viene memorizzata sul disco P
- Logicamente si dispone di un singlo disco ad alta capacità L'overhead della capacità è 25%
- RAID4 consente di parallelizzare small read



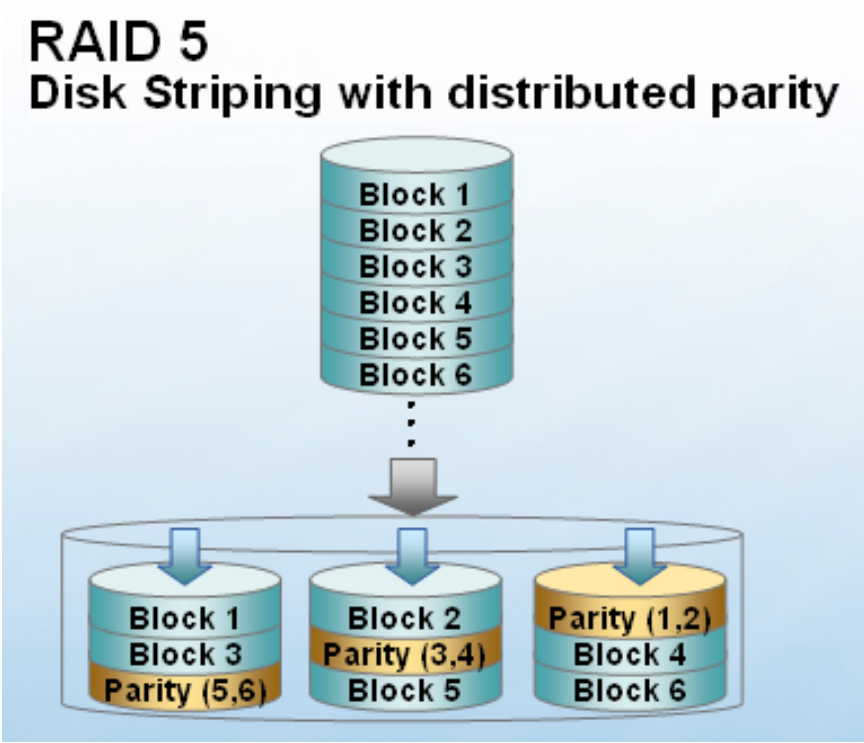
Inspiration for RAID 5

- Parity disk è un collo di bottiglia, tutte le write (e facoltativamente le read) devono poter accedere al Parity disk



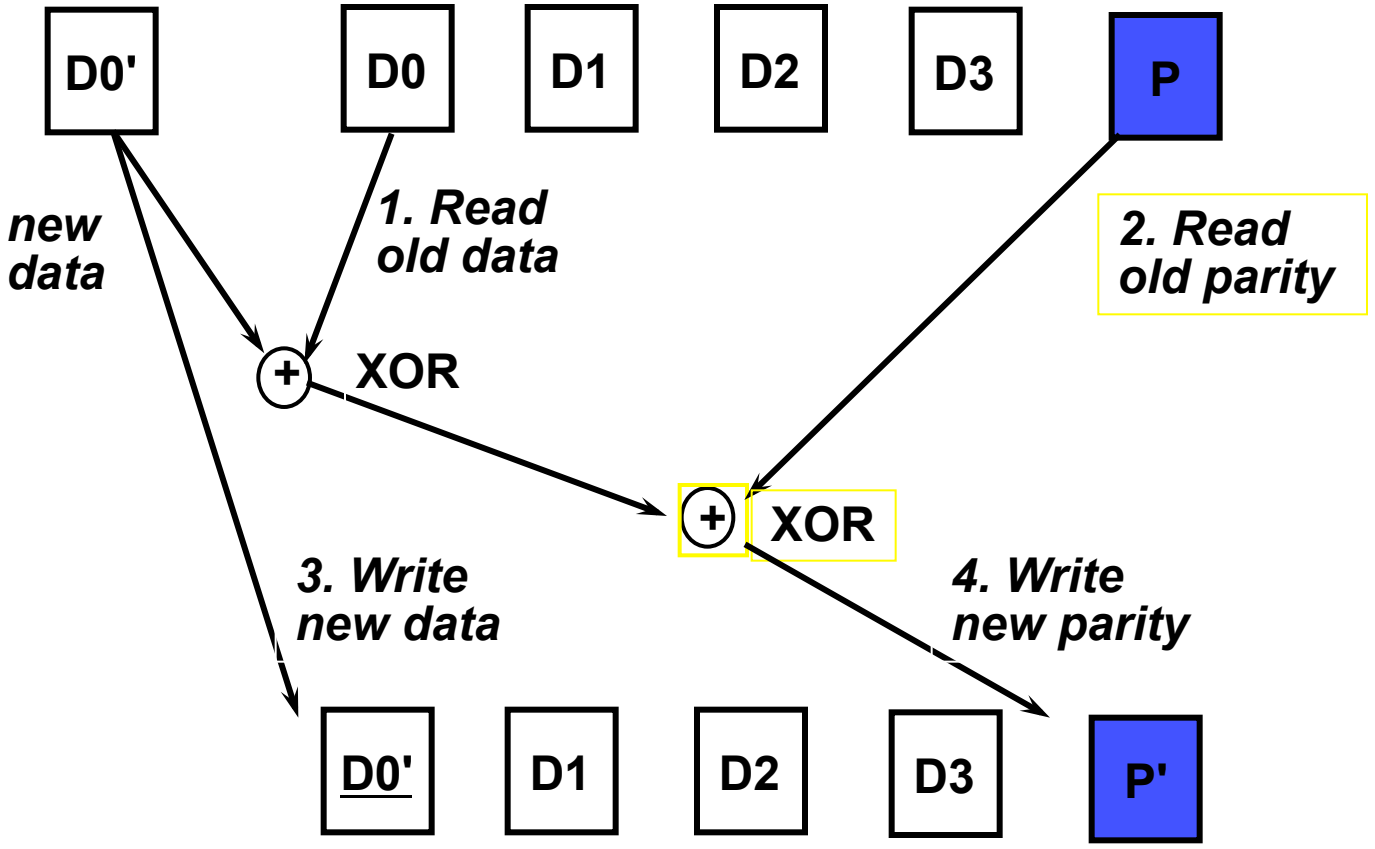
RAID 5

- Poichè i dati di parità sono distribuiti è possibile effettuare più write (small) parallele
  - Esempio: write di A0, B1 usano dischi 0, 1, 4, 5, e posson oessere eseguiti in parallelo
  - Permane il problema delle small write



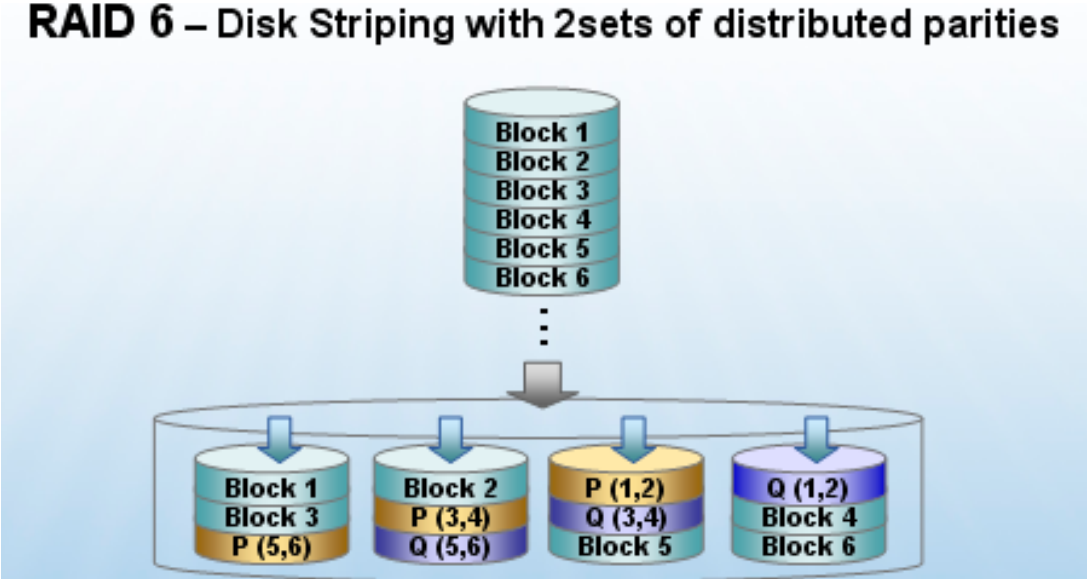
Problema: Small Write

1 Logical Write = 2 Physical Reads + 2 Physical Writes



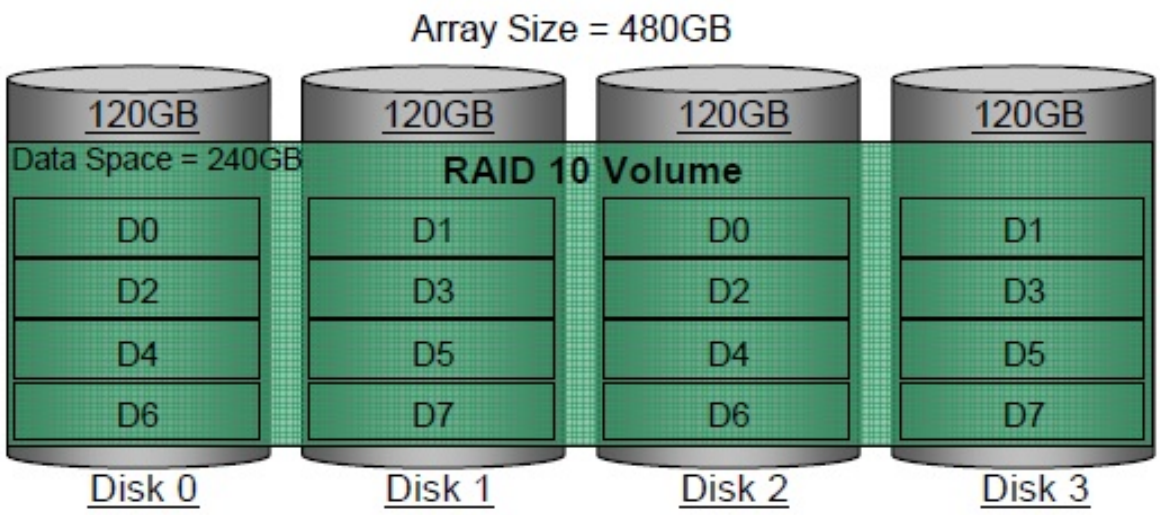
RAID 6

- In grado di correggere sino a 2 errori
- Usa due insiemi di informazioni di parità
- Small write richiede 6 accessi



RAID 1+0

- Ottenuto dalla composizione dei principi sottostanti RAID 1 e RAID 0
- Tra i sistemi RAID oggi più diffusi

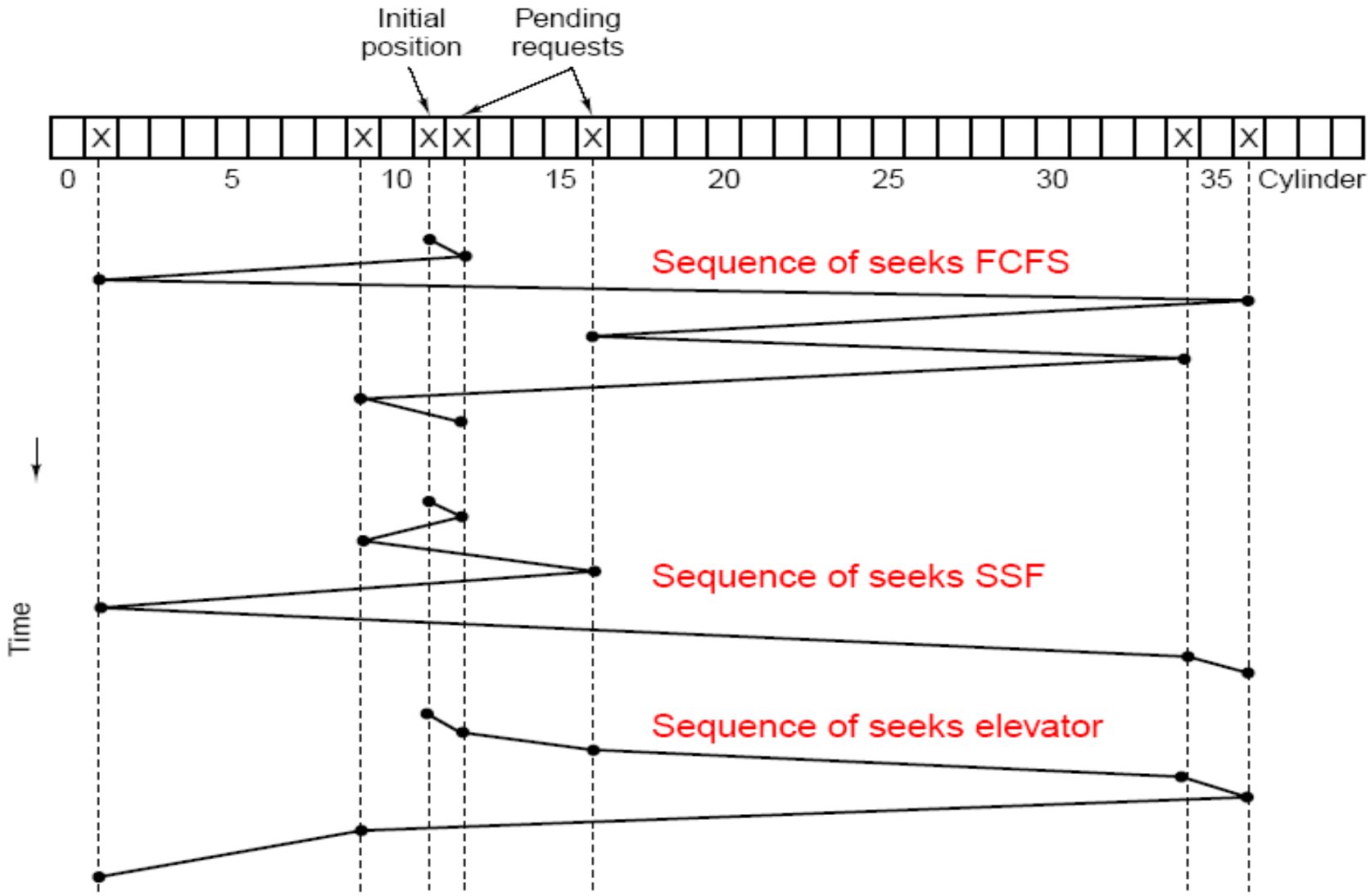


Disk Arm Scheduling

- Il tempo richiesto per leggere o scrivere un blocco del disco è determinato da tre fattori
  - Seek time
  - Rotational delay
  - Actual transfer time
- Il termine preponderante è il tempo di seek
- Il controllo degli errori è effettuato dal device controller

Disk scheduling

- L'operazione di seek è molto costosa (ms), il SO cerca quindi di ottimizzare le richieste al disco al fine di ridurre l'impatto di questa operazione:
  - FCFS
    - Si usa quando il numero medio di richieste è basso
  - SSTF
    - Minimizza il movimento del braccio
    - Favorisce i blocchi mediani
  - SCAN (elevator)
    - Serve le richieste in una direzione sino all'ultima richiesta
  - C-SCAN
    - Come scan ma solo in una direzione



L’interfaccia verso l’utente

Termini

- Campo
  - Elemento base
  - Contiene un singolo valore
  - Caratterizzato dal tipo di dato e dalla dimensione
- Record
  - Insieme di campi opportunamente correlati
  - Trattato come una singola entità logica
    - Esempi: record studente, impiegato, cittadino

File

- File
  - Insieme di record omogenei opportunamente correlati
  - Identificato da un nome logico
- Database
  - Collezione di record non necessariamente omogenei e di relazioni tra loro esistenti

File naming

- Il primo problema che il file system deve risolvere è proporre agli utenti uno schema attraverso il quale identificare i propri file all'interno di un sistema
  - I nomi solitamente possono contenere un qualunque carattere alfanumerico fatta eccezione per alcuni caratteri speciali
  - I nomi variano in lunghezza 8-255 caratteri
  - Alcuni sistemi non distinguono tra maiuscole e minuscole
  - È oramai prassi utilizzare una notazione postfissa

File Naming

Extension	Meaning
file.bak	Backup file
file.c	C source program
file.gif	Compuserve Graphical Interchange Format image
file.hlp	Help file
file.html	World Wide Web HyperText Markup Language document
file.jpg	Still picture encoded with the JPEG standard
file.mp3	Music encoded in MPEG layer 3 audio format
file.mpg	Movie encoded with the MPEG standard
file.o	Object file (compiler output, not yet linked)
file.pdf	Portable Document Format file
file.ps	PostScript file
file.tex	Input for the TEX formatting program
file.txt	General text file
file.zip	Compressed archive

Attributi o metadata

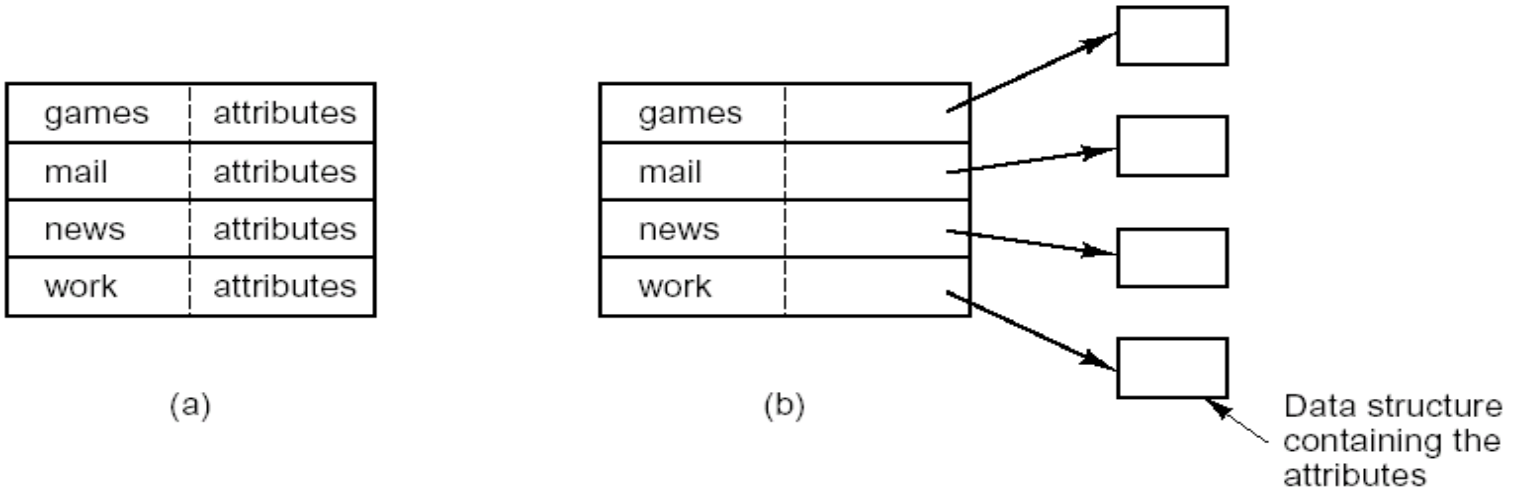
- Oltre che dal nome un file è caratterizzato dagli attributi: informazioni associate al file ed usate dal sistema per svolgere attività di gestione e manutenzione del file:
  - Controllo degli accessi
  - Back-up
  - Gestione degli spazi
  - accounting
- Sono dati molto dipendenti dal sistema operativo

Alcuni attributi di un file

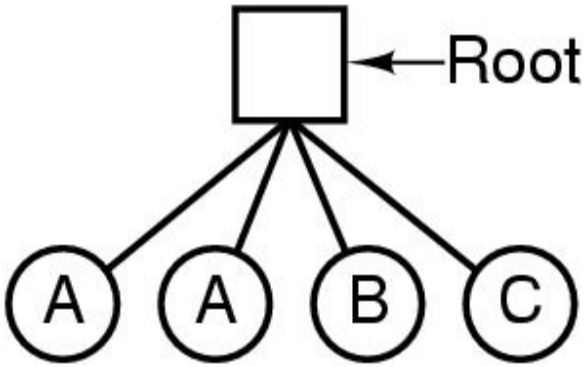
Attribute	Meaning
Protection	Who can access the file and in what way
Password	Password needed to access the file
Creator	ID of the person who created the file
Owner	Current owner
Read-only flag	0 for read/write; 1 for read only
Hidden flag	0 for normal; 1 for do not display in listings
System flag	0 for normal files; 1 for system file
Archive flag	0 for has been backed up; 1 for needs to be backed up
ASCII/binary flag	0 for ASCII file; 1 for binary file
Random access flag	0 for sequential access only; 1 for random access
Temporary flag	0 for normal; 1 for delete file on process exit
Lock flags	0 for unlocked; nonzero for locked
Record length	Number of bytes in a record
Key position	Offset of the key within each record
Key length	Number of bytes in the key field
Creation time	Date and time the file was created
Time of last access	Date and time the file was last accessed
Time of last change	Date and time the file has last changed
Current size	Number of bytes in the file
Maximum size	Number of bytes the file may grow to

File Directory

- L'elenco dei file contenuti in un device è memorizzato in apposti file di sistema chiamati directory
- File di proprietà del sistema operativo
- Oltre ai nomi dei file contengono anche i loro attributi
- Esistono diversi modi per organizzare le directory all'interno di un file system



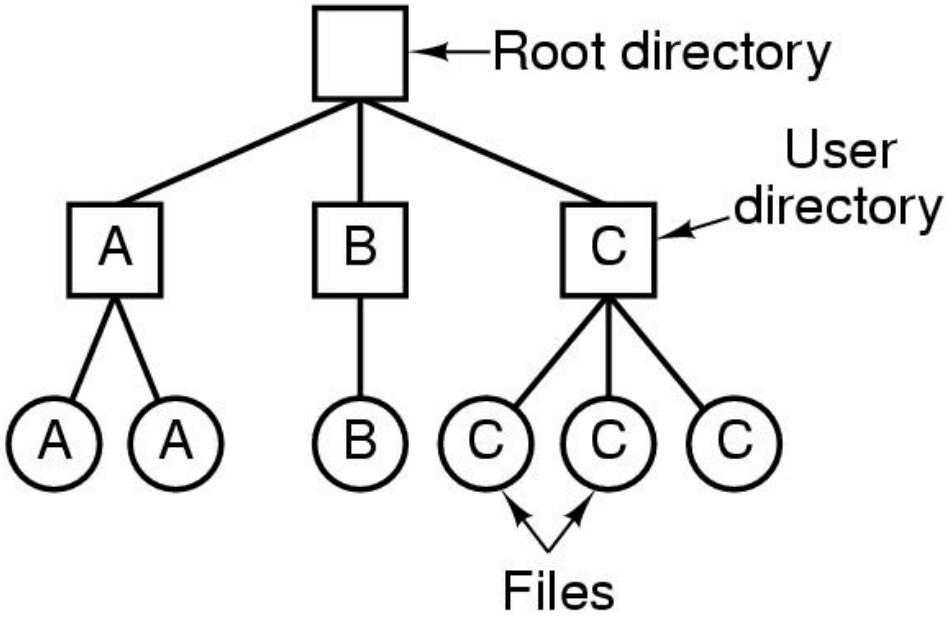
Single-Level Directory



Le lettere indicano i diversi proprietari dei file

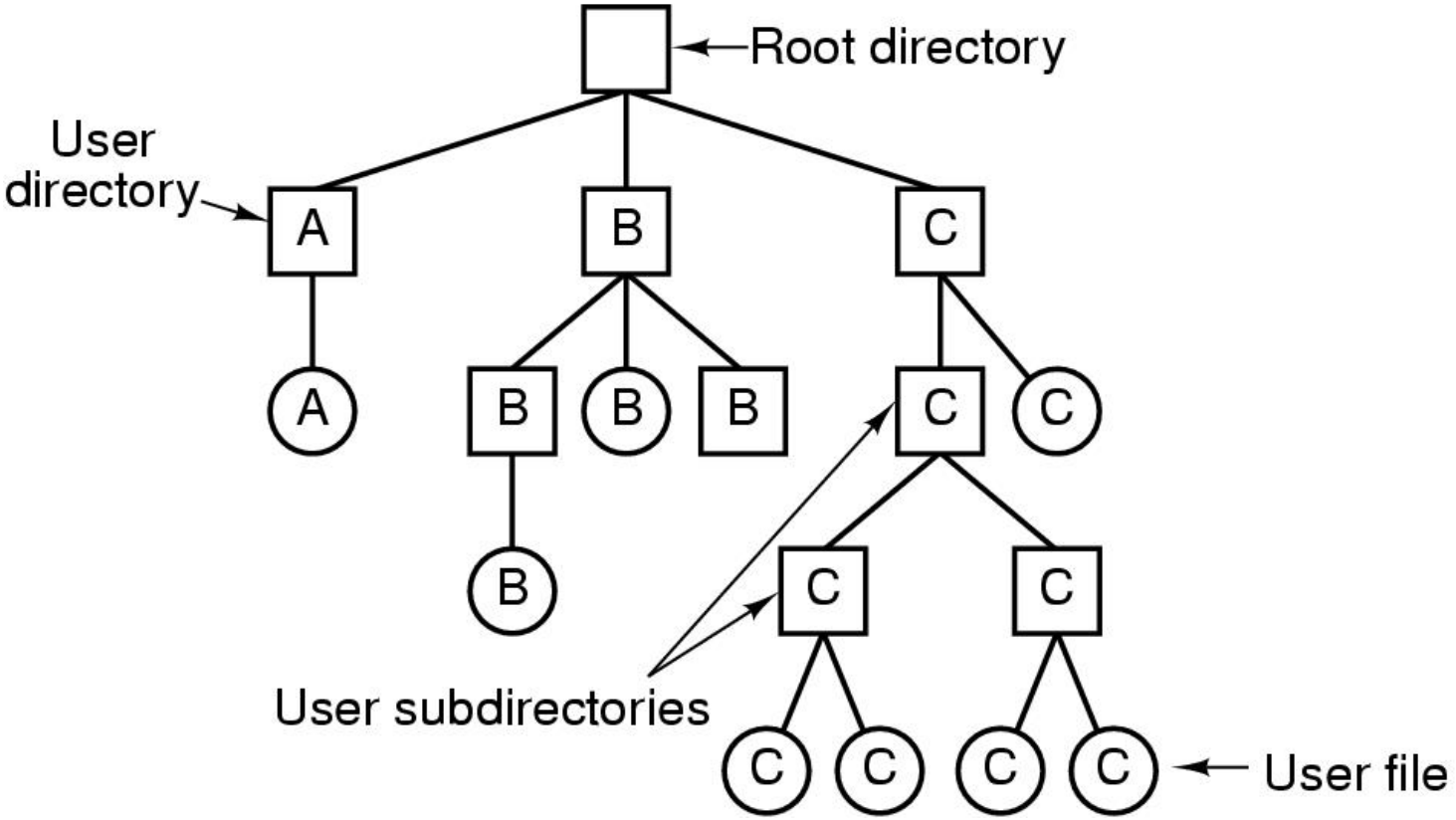
- Un sistema con directory ad un livello
  - contiene 4 file
  - Di proprietà di 3 persone, A, B, and C
  - Ovviamente non possono coesistere due file con lo stesso nome

Two-level Directory



- Quando un utente accede ad un file il SO deve sapere chi è al fine di individuare il corretto file

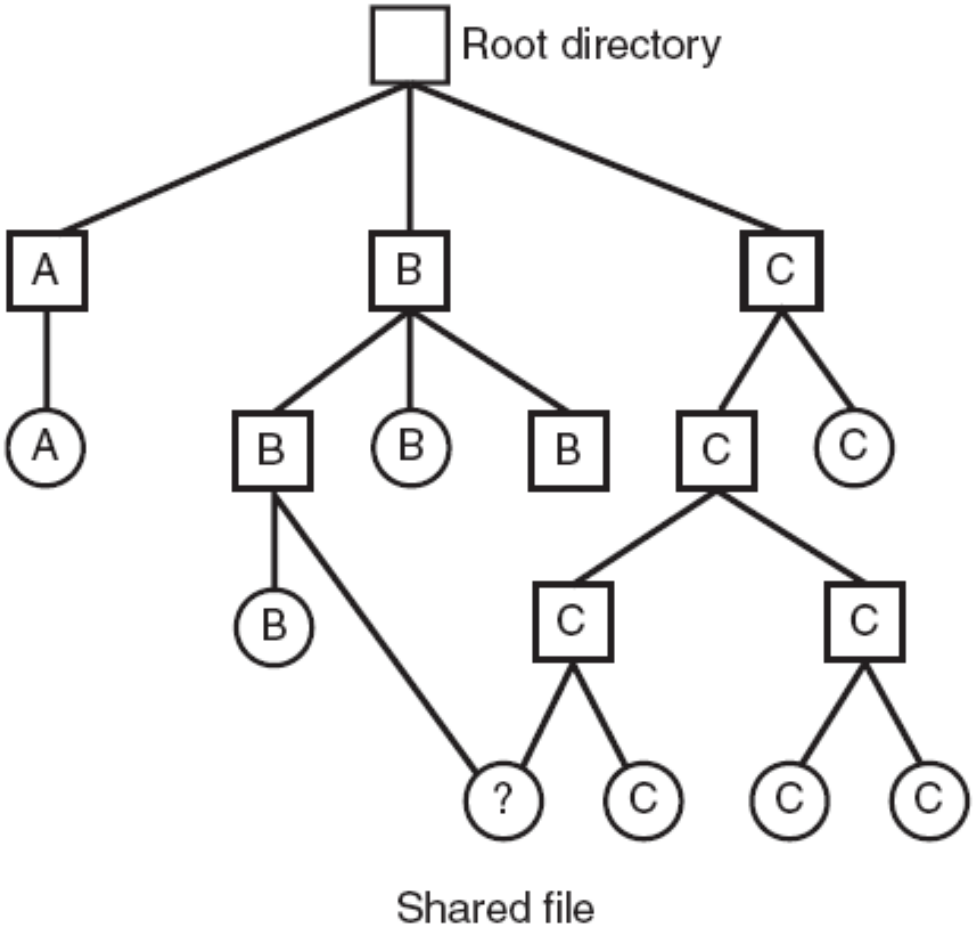
Hierarchical Directory



Path name

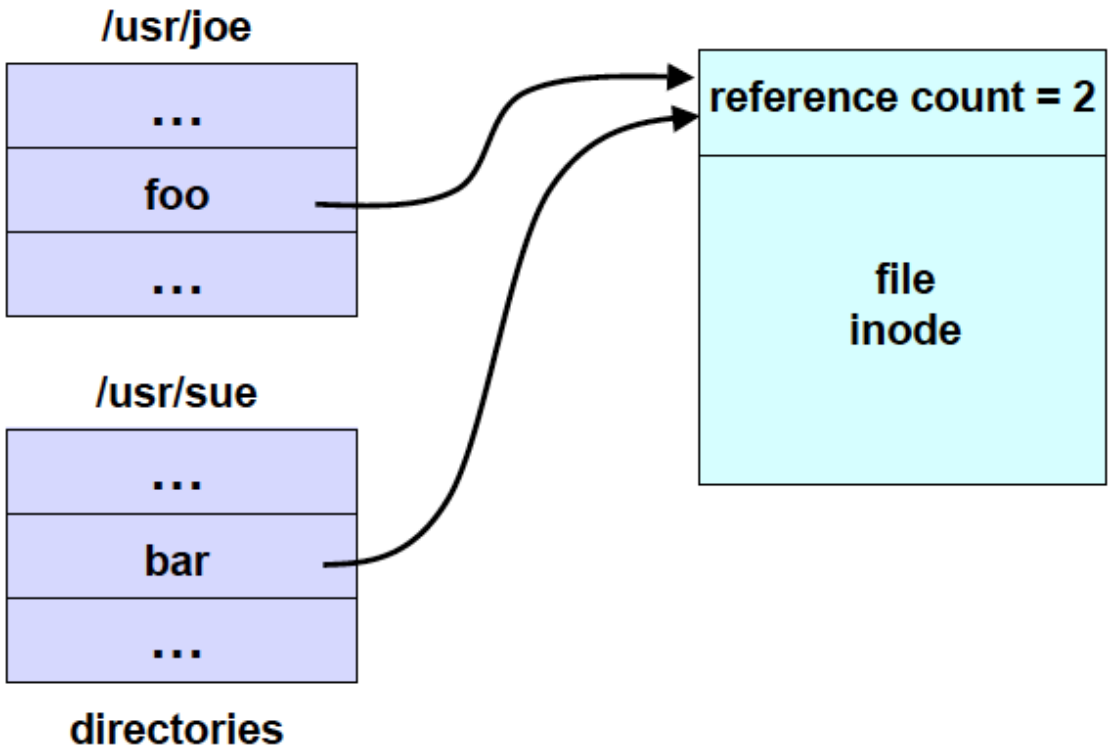
- Nei casi di directory con strutture ad albero il nome dell'utente e del file non sono più sufficienti per individuare univocamente un file
- Si usano i path name:
  - Assoluti: che descrivono il commani a partire dalla radice (root) iniziano sempre con / (\)
  - Relativi: sono computati a partire da una working directory (ogni processo ne possiede una)

File condivisi

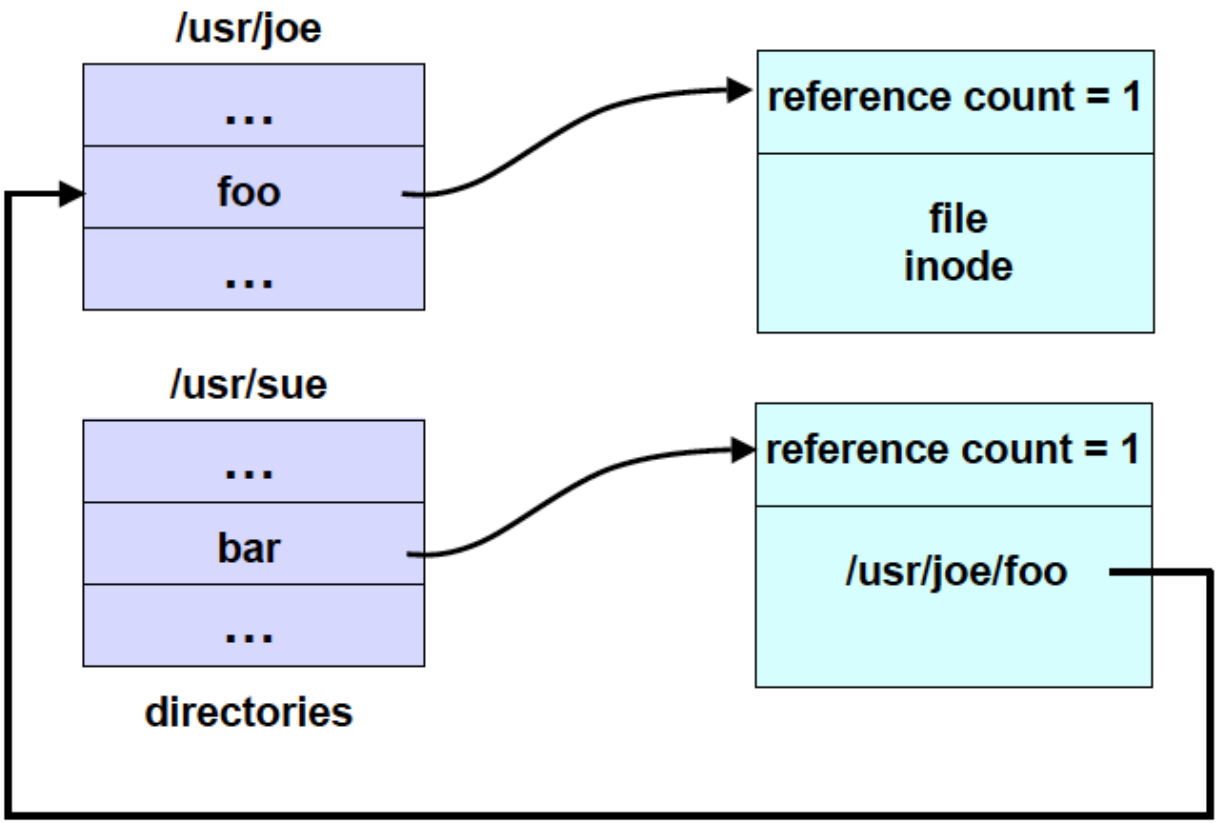


Shared file

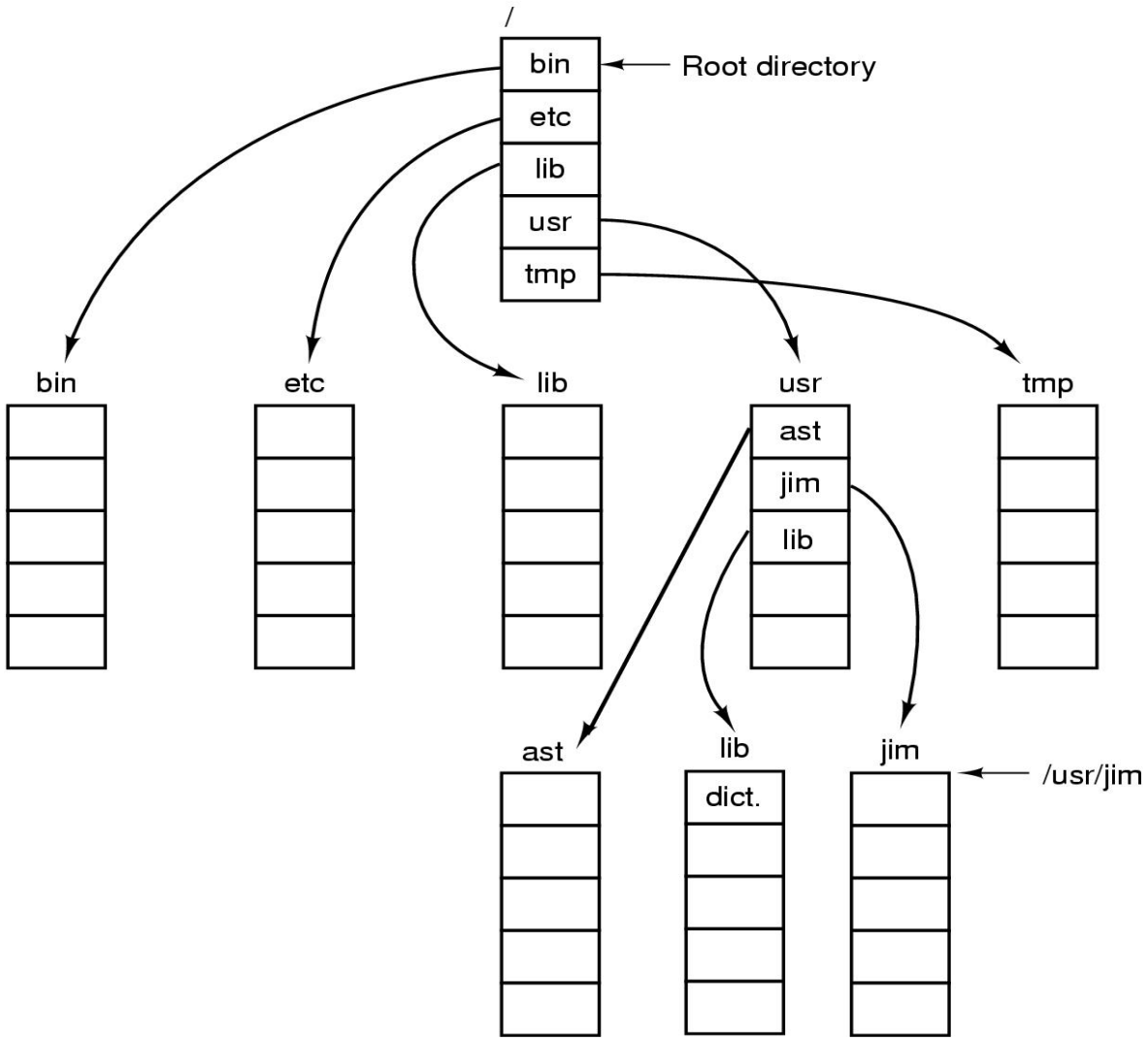
File condivisi: hard link



File condivisi : soft link



Path Name

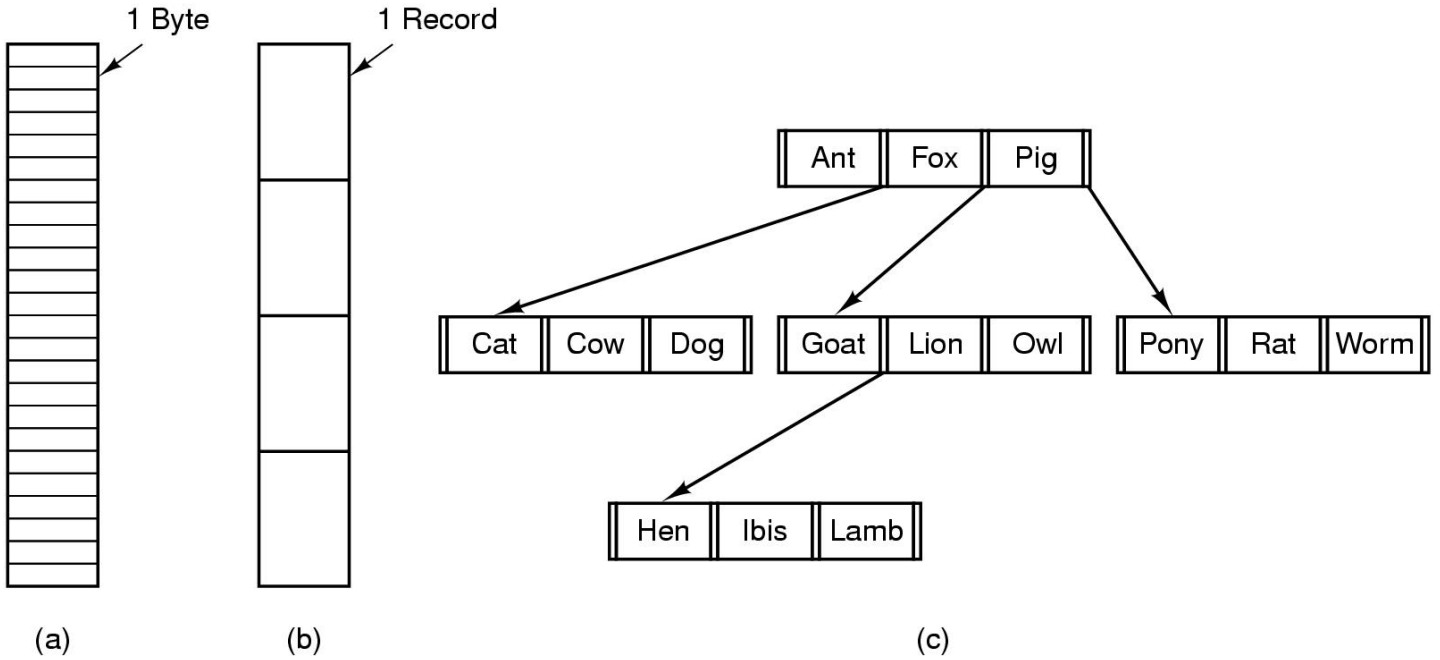

/
bin
etc
lib
usr
tmp
Root directory
bin
etc
lib
usr
tmp
ast
jim
lib
ast
lib
jim
dict.
/usr/jim

Sulle directory un utente può …

1. Create
2. Delete
3. Opendir
4. Closedir
5. Readdir
6. Rename
7. Link
8. Unlink

Struttura dei file

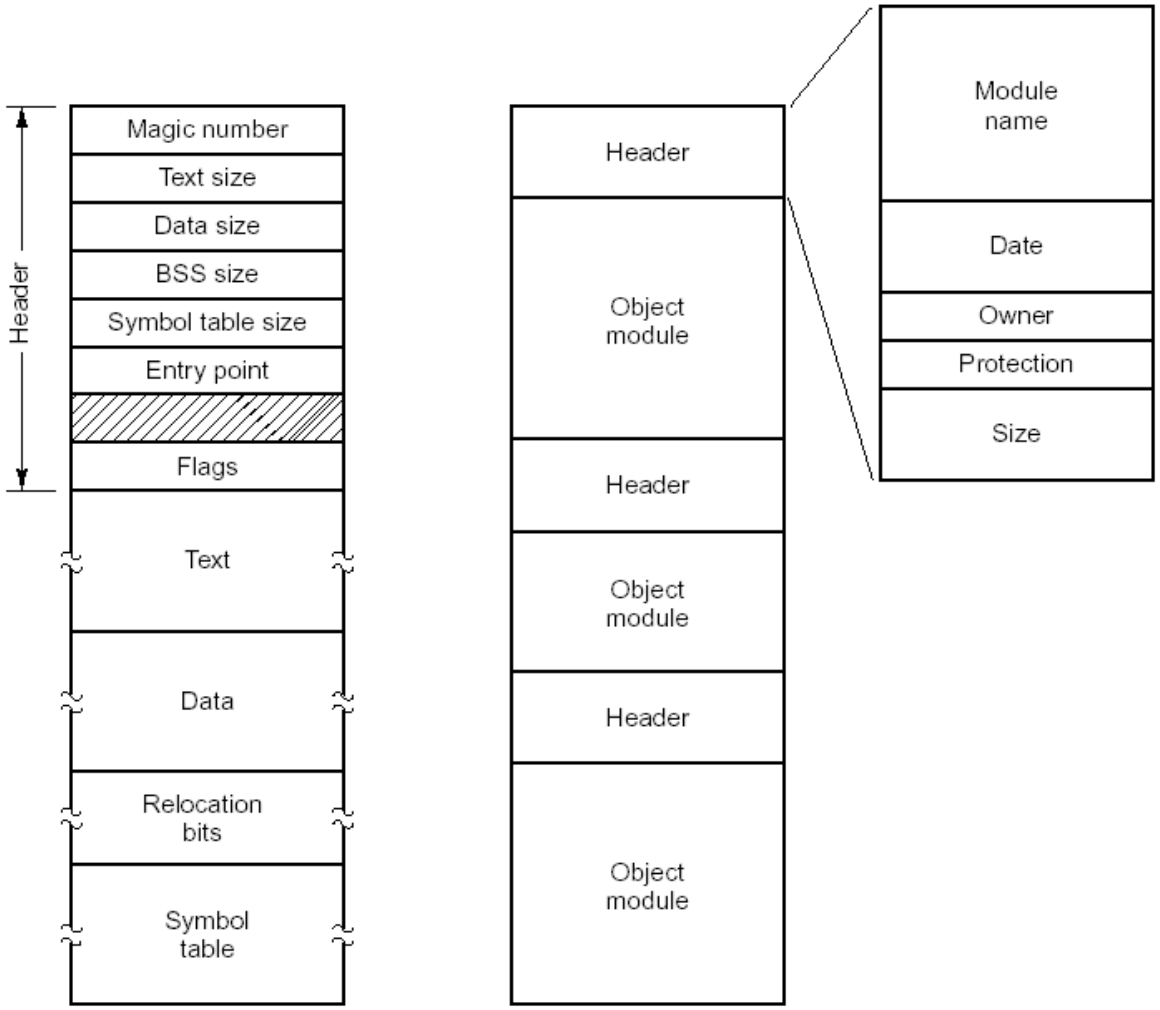
- Ulteriore elemento che caratterizza un file system sono le modalità con cui consente ad un utente di organizzare le proprie informazioni all'interno di un file



Tipi di file

- Ogni file system può supportare diversi tipi di file che differiscono tra loro per l'uso che il sistema ne fa
  - Regular file: contengono informazioni degli utenti
    - ASCII
    - Binary
  - Directory: usati dal file system
  - Character special file: presenti sul sistema Unix, dove sono usati per la gestione di dispositivi seriali
  - Block special file: presenti sul sistema Unix, dove sono usati per la gestione di dispostivi a blocchi

Binary file



File Access

- Le modalità di accesso determinano come un'applicazione può accedere ai dati memorizzati in un file

- Accesso Sequenziale
  - Legge tutti i bytes/records dall'inizio
  - Non è possibile effettuare salti all'interno del file, è possibile ritornare all'inizio o ripartire dall'ultima operazione di lettura

File access

- Random access
  - bytes/records possono essere letti in qualunque ordine
  - Fondamentale per supportare data base
  - La read può avere due diversi formati:
    - Move file marker (seek), then read (file)
    - Read (file, n.ro rec.), Read (file, key)

Su di un file un utente può …

1. Create
2. Delete
3. Open
4. Close
5. Read
6. Write
7. Append
8. Seek
9. Get attributes
10. Set Attributes
11. Rename