Il File System

Le applicazioni su un calcolatore hanno bisogno di memorizzare e rintracciare informazioni.

Le informazioni memorizzate nello spazio degli indirizzi di un processo vengono perse al termine dell'esecuzione del processo.

Un processo può utilizzare il suo spazio degli indirizzi per memorizzare un quantitativo limitato di informazioni.

Può essere necessario che le informazioni siano memorizzate per lungo tempo.

Più processi possono aver bisogno delle stesse informazioni contemporaneamente per cui è necessario che queste e la loro allocazione siano indipendenti dal processo.

Per memorizzare grandi quantità di informazioni, rendere la memoria permanente e dare la possibilità a più processi di accedere alle stesse informazioni è necessario registrare tali informazioni in dischi o altri supporti in unità dette file.

I File…

Il sistema operativo consente la gestione dei file mediante il modulo di gestione del *File System*.

Oltre al contenuto proprio del file, il file system gestisce informazioni complementari relative al file come:

- struttura;
- nome;
- tipo di accesso;
- protezione, ecc.

Il nome del file si suddivide generalmente in:

Nome.Estensione

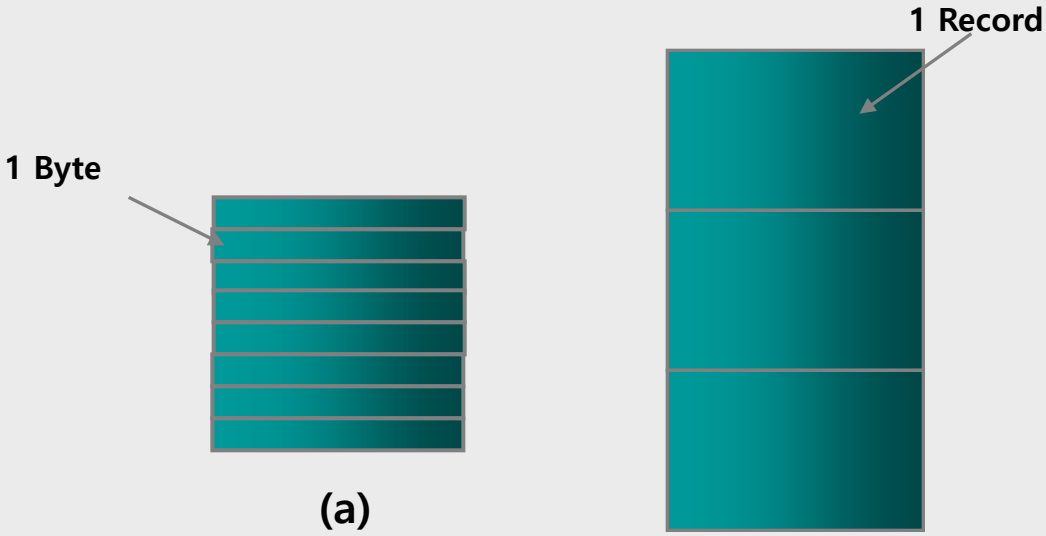
entrambi con un numero variabile di caratteri e varia utilizzazione del minuscolo/maiuscolo.

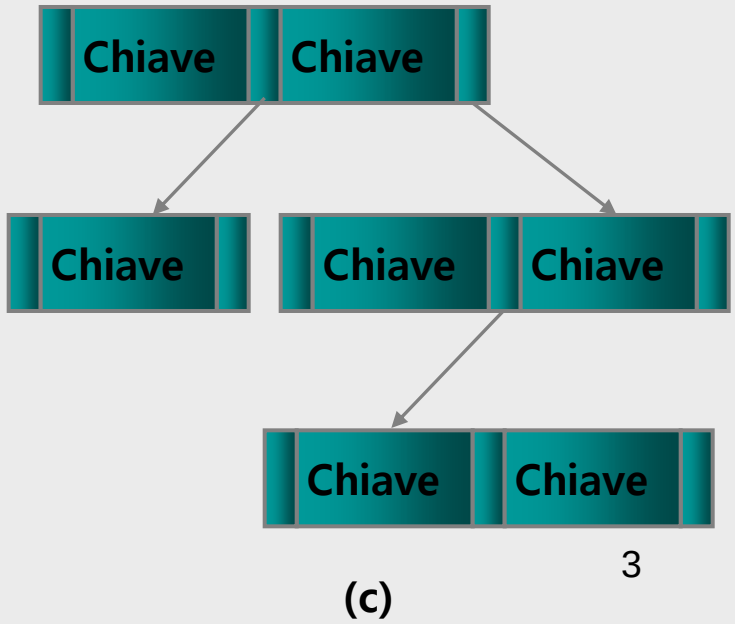
Spesso l'estensione indica un particolare tipo di file…

I File (struttura)…

I file, in genere, contengono le informazioni in tre tipi di strutture:

- *sequenza non strutturata di byte:* consente la massima flessibilità al programma utente. Dos e Unix utilizzano questa struttura (a); Accesso diretto ai singoli byte
- *sequenza di record a lunghezza fissa:* ha perso popolarità con il tempo (b);
- *albero di record con campo chiave:* ancora in uso sui grossi mainframe usati per l'elaborazione di dati commerciali (c).





I File (tipi e formati)…

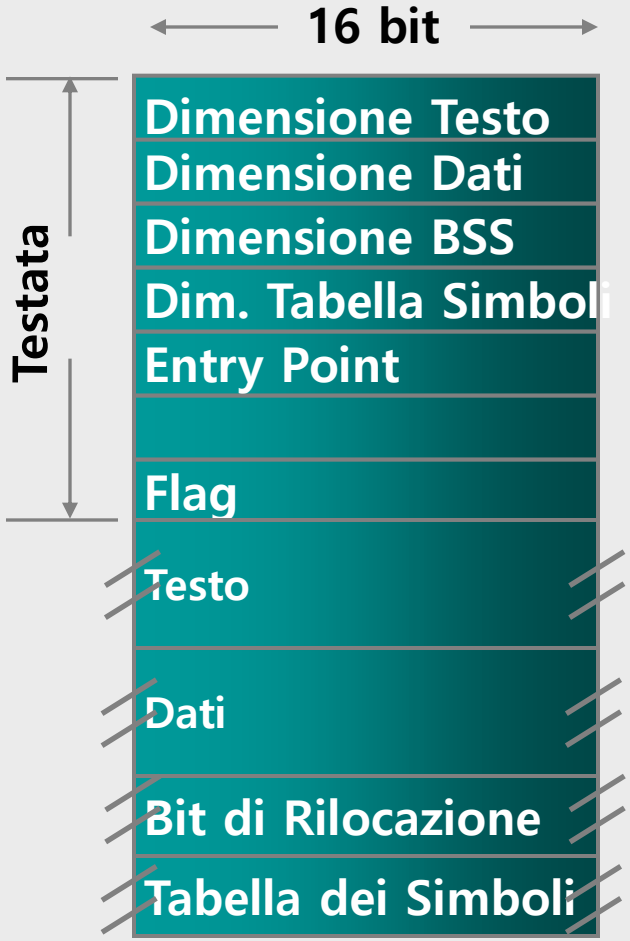
Tipi di file:

- ***file regolari:*** contengono le informazioni dell'utente, un programma eseguibile, ecc.
- ***directory:*** conservano la struttura del file system ed informazioni sui file regolari;
- ***file speciali a caratteri:*** usati per modellare unità di input/output seriali come video, stampanti, reti, ecc;
- ***file speciali a blocchi:*** usati per modellare unità di input/output a blocchi come i dischi.

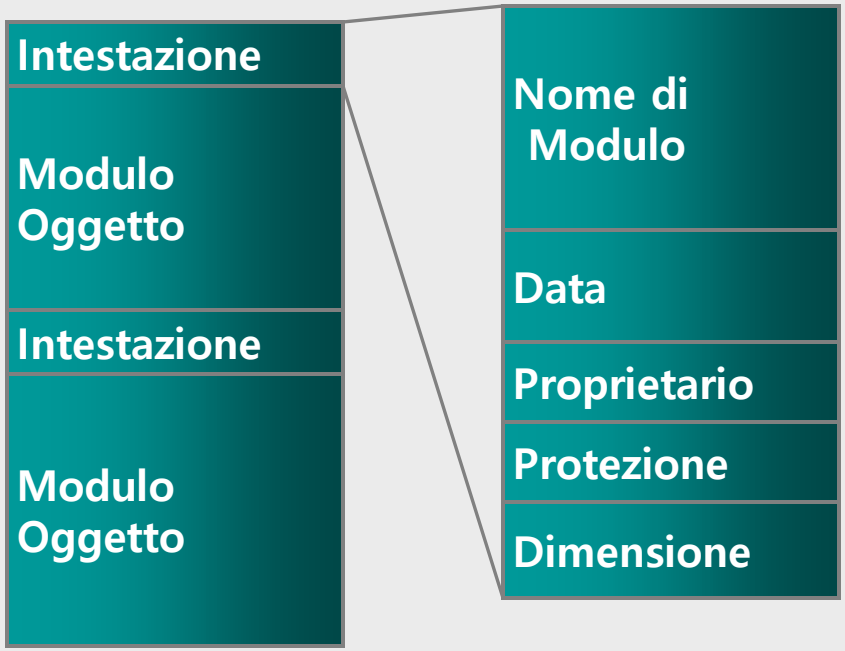
Le informazioni nei file sono generalmente contenute in due formati:

- ***codice ASCII (o UNICODE, ecc.)***
  - ✘ utilizza 8bit di cui uno di controllo: 7bit=128 configurazioni
  - ✘UNICODE: Codice a 16 bit (codifica i caratteri usati in quasi tutte le lingue vive e in alcune lingue morte, nonché simboli matematici e chimici, cartografici, l'alfabeto Braille, ideogrammi etc)
  - ✘ i file programma sorgente ed i documenti sono generalmente registrati in questo formato.
- ***codice Binario:***
  - ✘ le sequenze di codici sono "illeggibili";
  - ✘ di solito i file memorizzati in questa modalità hanno una struttura interna.

I File (formati)...



file eseguibile Unix



archivio contenente procedure

I File (accesso)…

- ***Accesso Sequenziale:***
  - i byte o i record che costituiscono il file sono letti uno dopo l'altro, dall'inizio alla fine;
- ***Accesso Casuale (diretto).***
  - l'accesso al blocco di informazioni avviene in maniera diretta;
  - l'avvento dei dischi ha consentito l'introduzione di questo metodo di accesso;
  - Meccanismo utilizzato dalle basi di dati
  - generalmente l'accesso è diretto al blocco e sequenziale all'interno del blocco.

I File (attributi)…

I sistemi operativi associano ai file altre informazioni come:

- data e ora della creazione;
- proprietario;
- dimensione, ecc.

Campo	***Significato***
Protezione	Chi può accedere al file e in che modo
Password	Parola d'ordine necessaria per accedere al file
Flag di sola lettura	0 per lettura/scrittura, 1 per sola lettura
Flag di sistema	0 per file normale, 1 per file di sistema
Flag ASCII/binario	0 per file ASCII, 1 per file binario
Lunghezza record	Numero di byte in un record
Tempo di creazione	Data e ora del momento in cui è stato creato il file
Dimensione attuale	Numero di byte nel file
Dimensione massima	Dimensione massima che il file può raggiungere

I File (operazioni)…

Il sistema operativo pone a disposizione dell'utente ***comandi di alto livello*** e ***chiamate di sistema*** per la gestione dei file. Alcune chiamate di sistema più comuni sono:

- ***create:*** creazione del file;
- ***delete:*** cancellazione del file;
- ***open:*** apertura del file con varie modalità (read-only, append, ecc.);
- ***close:*** chiusura del file;
- ***read:*** lettura di blocchi di dati;
- ***write:*** scrittura di blocchi di dati;
- ***append:*** aggiunta, in coda al file, di nuovi dati;
- ***seek:*** posizionamento del puntatore all'interno del file.

File mappati in memoria…

Un file viene caricarlo tutto o in parte in memoria e gli vengono assegnati indirizzi virtuali.

• I processi possono accedere in maniera trasparente ai file (possono essere quindi contenuti in memoria centrale per interi o per blocchi)

• Si velocizzano le operazioni compiute sul file, soprattutto quando si riferiscono a parti limitate e non all'intero file.

La virtualizzazione dell'accesso comporta alcuni problemi :

- In caso di apertura contemporanea del file da parte di più processi alcune informazioni possono risultare inconsistenti;
- Quando il processo che sta usando il file termina, quest'ultimo viene riscritto sul disco.
- In caso di caduta del sistema alcune informazioni possono essere perse;

Le directory

I vecchi sistemi operativi prevedevano un'***unica directory*** in cui confluivano tutti i file.

Questo concetto è ormai superato dai ***sistemi gerarchici*** di directory.

Tipicamente un *elemento della directory* contiene:

- il nome di un file ed eventualmente include anche gli ***attributi*** del file;
- in alternativa, il nome di un file ed un ***puntatore*** ad un'altra struttura in cui si trovano gli attributi e gli indirizzi del disco.

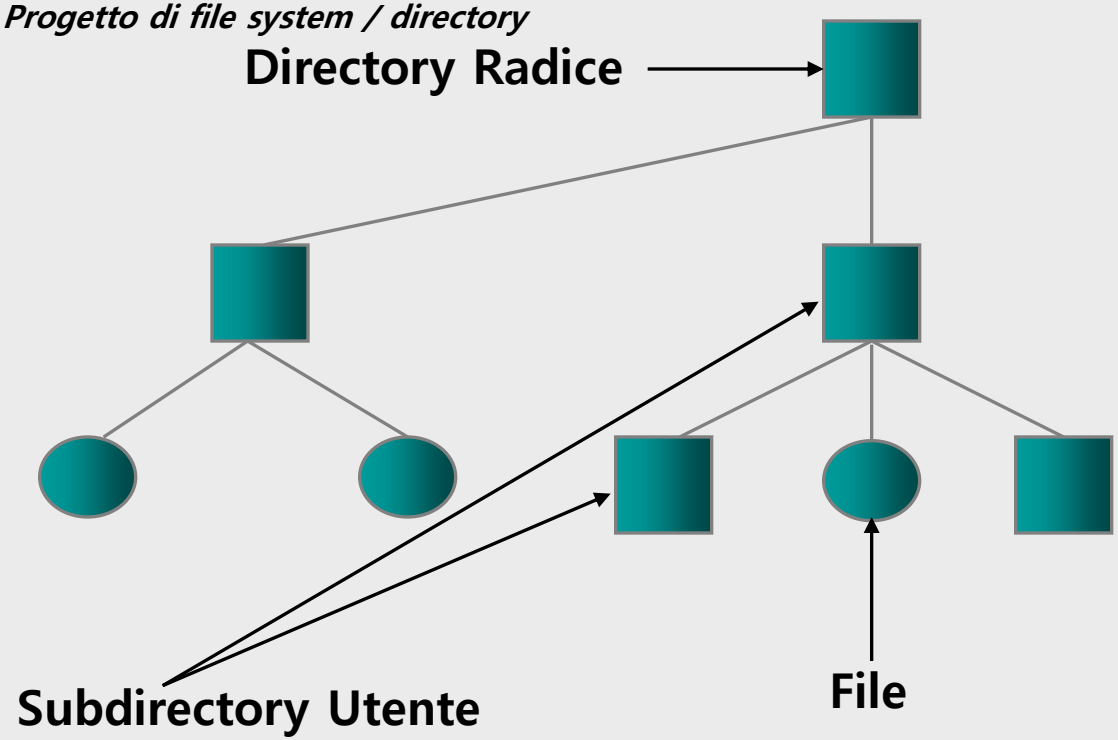
Ogni directory può contenere:

- ***file regolari***;
- altre ***directory***.

In questo modo, a partire da una directory ***radice*** (***root*** directory) è possibile generare un albero di directory e ***subdirectory***.

Le directory

Progetto di file system / directory



Il tipo di organizzazione dell'*albero delle directory* dipende dalle scelte dell'amministratore di sistema. Un tipico esempio prevede:

- la root directory include alcune directory di sistema come

  etc, bin, lib, tmp, ecc.;

- la directory *usr* contiene una subdirectory per ogni utente ed ogni utente organizza il proprio sottoalbero nel modo che ritiene più efficiente e comodo per il tipo di attività che svolge.

- ***path name assoluto:*** cammino dalla directory radice al file. I componenti del cammino sono separati da opportuni simboli di separazione come " / " in Unix

- ***path name relativo:*** si usa congiuntamente al concetto di directory di lavoro. Un utente può definire una directory come *directory di lavoro corrente*. In questo caso tutti i path name che non iniziano con la directory radice sono considerati relativi alla directory di lavoro.

Le directory (operazioni)

Alcune tipiche chiamate di sistema per la gestione delle directory sono:

- ***create:*** crea una directory vuota;

  mkdir nome
  Crea directory nidificate: mkdir -p dir1/dir2

- ***delete:*** cancella una directory;

  cancellare directory vuota: *rmdir nome*
  cancellare directory non vuota: *rm –rf nome*

- Spostarsi tra le directory:

  cd newdir
  cd .. (directory precedente)
  cd (directory home)
  cd ~bill (directory home dell'utente bill)

- ***rename:*** rinomina una directory;

  mv source dest

- *Elencazione file:*

  ls
  ls -l elenco dettagliato

Implementazione del file system

Un fattore chiave nell'implementazione della memorizzazione dei file è tener traccia di quali blocchi del disco associare a ciascun file.

Le modalità di allocazione dei blocchi e del relativo reperimento sono:

- ***Allocazione contigua***
- ***Allocazione a lista concatenata***
- ***Allocazione a lista concatenata con indice***
- ***Allocazione mediante uso di tabelle i-node***

Allocazione contigua

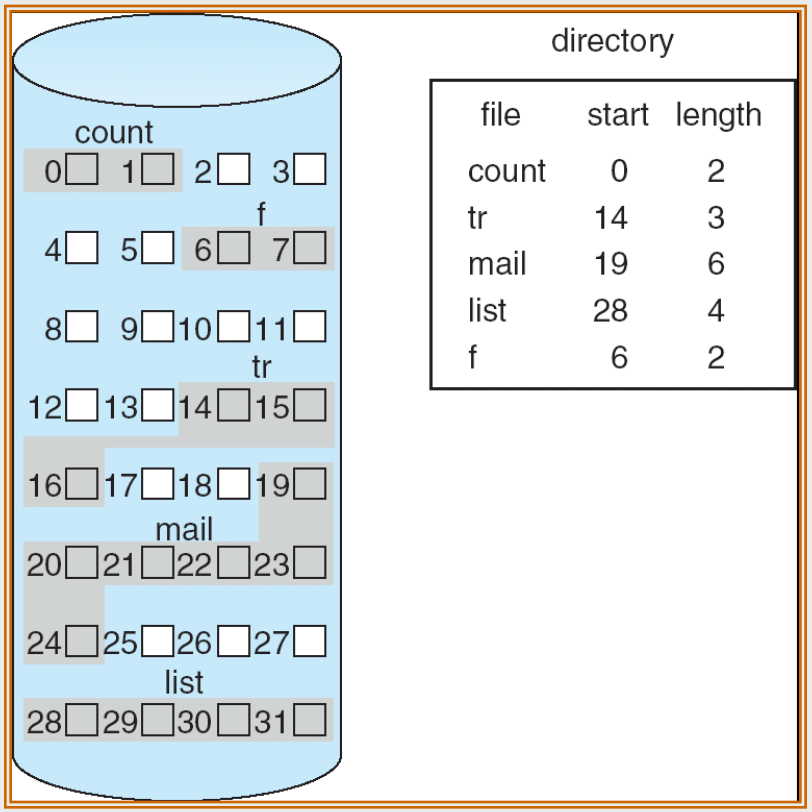
Memorizza il file in blocchi di disco consecutivi.

Pregi:

- semplice da implementare: occorre tenere traccia dell' indirizzo di inizio del file e della sua lunghezza;
- efficiente perché permette di leggere l'intero file con una sola operazione.

Difetti:

- è necessario conoscere la dimensione massima del file;
- il disco risulta frammentato (frammentazione esterna) e molto spazio viene sprecato.
- Difficoltà di reperire spazio libero
  - *Best fit*
  - ***First fit***
  - Worst fit



Allocazione a lista concatenata

Ogni blocco associato al file contiene:

- dati veri e propri;
- il numero del blocco successivo assegnato al file.

Pregi:

- non implica spreco di spazio perché i blocchi vengono allocati dinamicamente;
- no frammentazione esterna;
- nell'elemento della directory viene memorizzato l'indirizzo del primo blocco e dell'ultimo blocco.

Difetti:

- La lettura sequenziale del file è semplice, ma l'accesso diretto ai blocchi è estremamente lento;
- Ogni blocco deve contenere il puntatore al blocco successivo. Tale area deve essere non accessibile ai programmi utenti… (protezione)

Allocazione a lista concatenata con indice

- I blocchi di disco assegnati al file non sono necessariamente contigui;
- L'elenco dei blocchi assegnati al file è mantenuto mediante una tabella (sezione su disco all'inizio di ciascuna partizione).
- L'elemento della tabella indicizzato dal numero di blocco contiene il numero di blocco successivo del file. L'ultimo blocco punta a NULL
- Blocchi vuoti sono indicati da NULL
- L'elemento directory contiene il numero del primo blocco del file

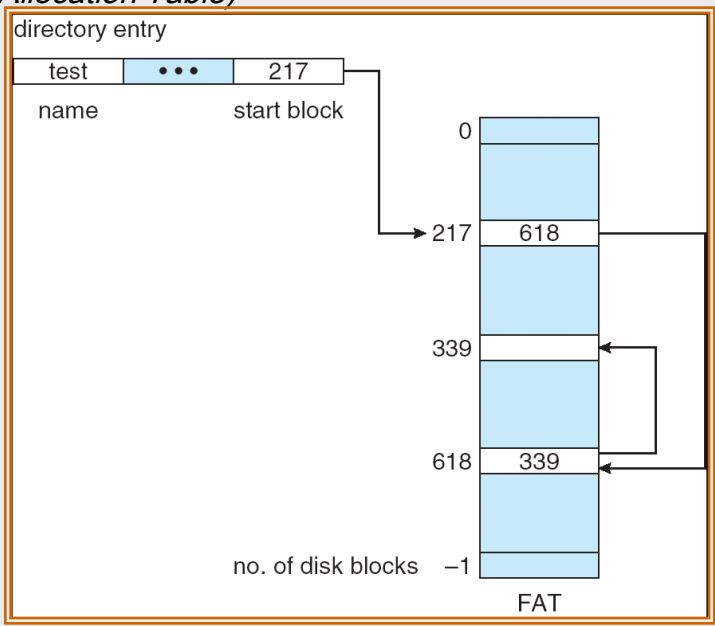
Questo tipo di allocazione è utilizzata nel sistema operativo MS-DOS: FAT (File Allocation Table)

Pregi:

- l'intero blocco è disponibile per i dati;
- l'accesso diretto ad un blocco è rapido;

Difetti:

- Ad un gran numero di blocchi di disco corrisponde una tabella grande che occupa molto spazio in memoria centrale.
- La lettura della FAT può causare numerosi spostamenti della testina. Soluzione: FAT in $



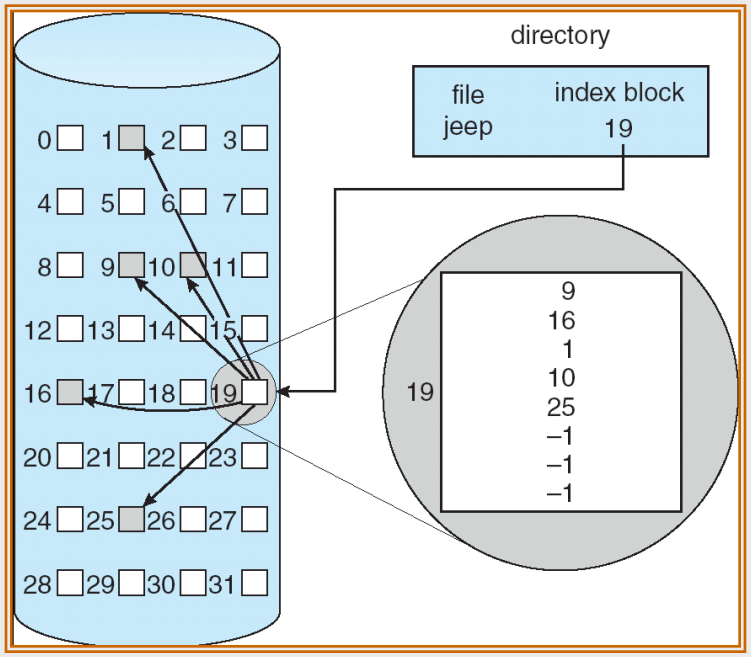
Allocazione mediante uso di tabelle i-node

Allocazione indicizzata: tutti i puntatori ai blocchi di un file sono raggruppati in una sola locazione: blocco indice.

Ogni file ha il proprio blocco indice

L'elemento directory contiene l'indirizzo del blocco indice

Allocazione utilizzata in Unix.



Gli i-node sono piccole tabelle che contengono:

- gli attributi del file;
- gli indirizzi dei primi quindici blocchi assegnati al file:

-I primi 12 puntano a blocchi diretti: contengono direttamente l'indirizzo dei blocchi del file

-Gli altri 3 puntano a blocchi indiretti.

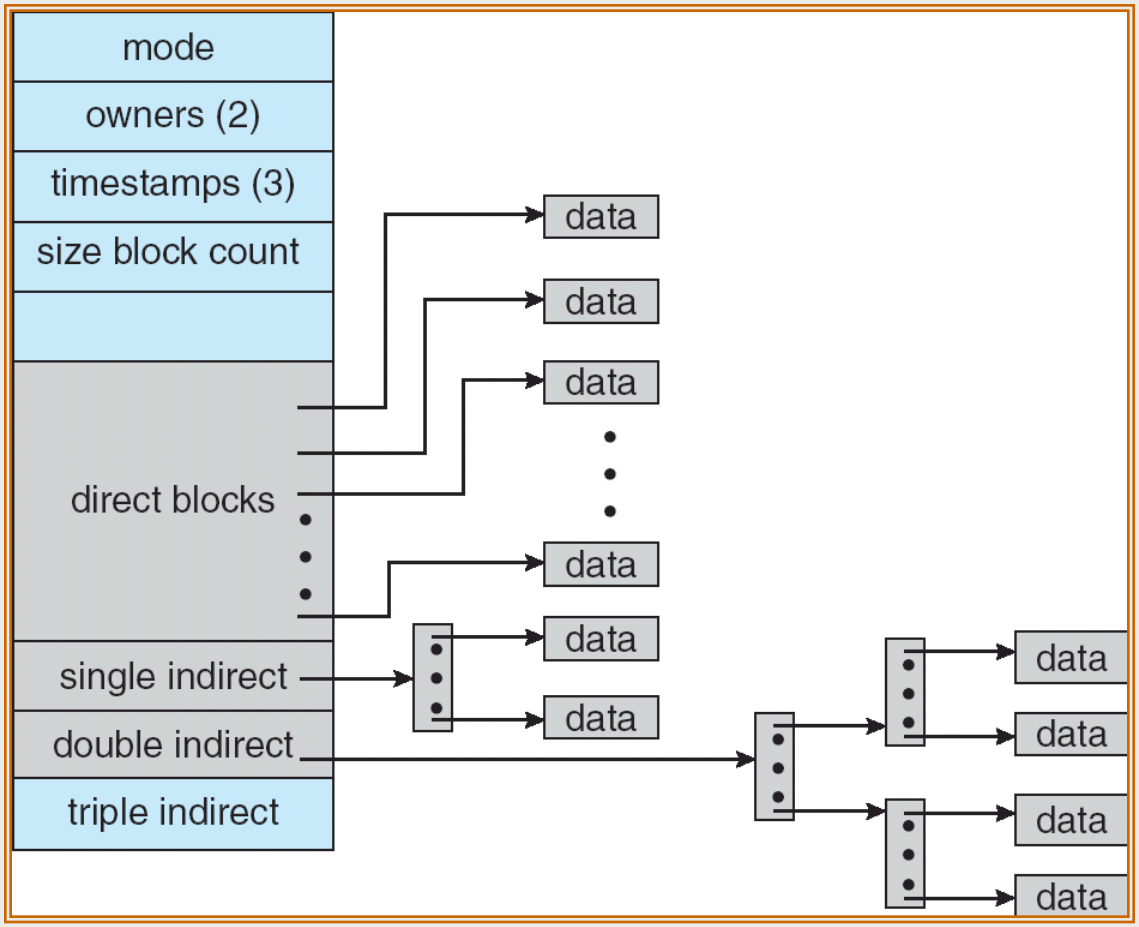
Blocco indiretto singolo: il blocco indirizzato non contiene dati ma indirizzi di altri blocchi i quali contengono dati

Blocco indiretto doppio

Blocco indiretto triplo

Allocazione mediante uso di tabelle i-node

Combined Scheme: UNIX (4K bytes per block)



Pregi

- **l'elemento di directory contiene solo il nome del file ed il numero di i-node;**
- consente di mantenere in memoria gli i-node dei file aperti sprecando poco spazio;
- consente di indirizzare un numero enorme di blocchi di disco;
- per i file di piccole dimensioni consente di reperire rapidamente l'indirizzo dei primi blocchi di disco.

Gestione dello spazio su disco: Dimensione dei blocchi

Blocco molto grande:

- rende più veloce il tempo di lettura (seek+rotational+transfert)
- provoca maggior spreco di spazio (frammentazione interna).

Blocco molto piccolo:

- causa numerosi accessi ai blocchi per leggere i dati
- consente una gestione più efficiente dello spazio.

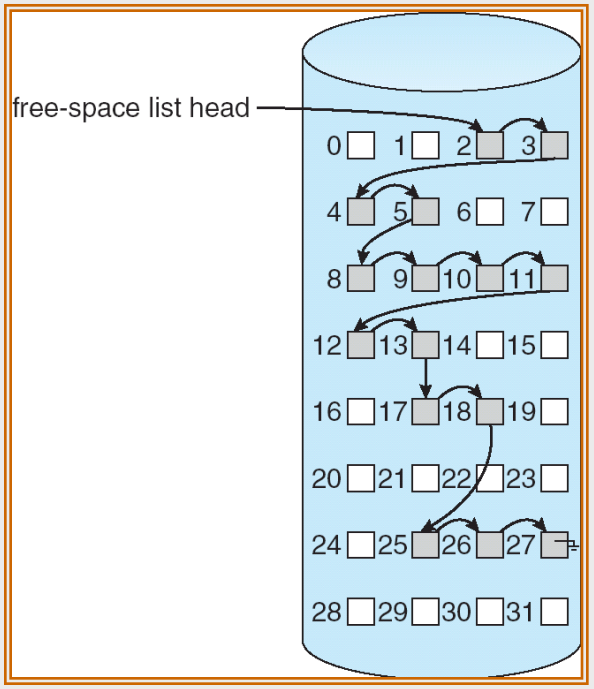
L'efficienza nel tempo e nello spazio sono intrinsecamente in conflitto.
Confrontando la velocità di trasferimento dei dati e l'efficienza dello spazio su disco risulta che a una buona utilizzazione dello spazio corrisponde una bassa percentuale di dati e viceversa.

COMPROMESSO:
Dimensione blocco di allocazione: 512 byte, 1 KB, 2 KB, 4KB, 8KB
NB: il *settore* è la minima unità fisica leggibile dal disco => la dimensione del settore sul disco è generalmente di 512 byte e la lettura di un blocco di allocazione di 1 K causerà sempre la lettura di due settori di disco.

Gestione dello spazio libero su disco

I metodi usati comunemente sono due:

- *lista concatenata:* tutti i blocchi liberi sono concatenati ed un puntatore indica il primo blocco libero. Necessità di proteggere il puntatore al primo blocco libero.
- *mappa di bit*: un disco di n blocchi richiede una mappa di n bit. I blocchi liberi si rappresentano con 1 sulla mappa, i blocchi allocati con 0 (o viceversa).



Bit map richiede extra space

Es:

block size = 2^{12} bytes (4KB)
disk size = $2^9 2^{30}$ bytes (500 gigabyte)
n = $2^{39}/2^{12} = 2^{27}$ bits (or 16M bytes)

Easy to get contiguous files

Il metodo della lista concatenata è più efficiente se il disco è quasi pieno, poiché il numero di blocchi è molto limitato. Inoltre in queste condizioni, la mappa di bit potrebbe risultare spesso piena e richiedere continui accessi ad altre parti di mappa per assegnare blocchi liberi.

Se c'è abbastanza memoria principale per contenere la mappa di bit, è generalmente preferibile quest'ultimo metodo.

Affidabilità del Fyle System

Il file system può contenere dati che non possono essere persi poiché di difficile o costoso recupero.

L'affidabilità del file system va perseguita spesso anche a scapito dell'efficienza o di economie.

I dispositivi di memorizzazione presentano spesso blocchi inutilizzabili:

- difetti di fabbricazione;
- usura;
- urti violenti.

I blocchi inutilizzabili non devono essere assegnati a nessun file, necessità una gestione appropriata.

Affidabilità del Fyle System

Gestione dei blocchi danneggiati:

- soluzione hardware
  - una parte di disco è riservata per tenere traccia dei blocchi rovinati e contiene dei blocchi di riserva;
  - in fase di inizializzazione il controllore del dispositivo legge questi blocchi e li sostituisce con blocchi di riserva, memorizzandone l'avvenuta sostituzione;
  - soluzione trasparente al file system.

- soluzione software
  - il file system tiene traccia dei blocchi rovinati assegnandoli tutti ad un file speciale da non utilizzare

Affidabilità del Fyle System BackUp

Nella pianificazione delle operazioni di backup vanno valutati principalmente:

- la frequenza con cui effettuare le copie di backup;
- i file che presentano la maggior necessità di essere copiati;
- la tecnica migliore per effettuare il backup.

Affidabilità del Fyle System BackUp

Frequenza: la scelta va pianificata in base alla quantità di lavoro svolta nel tempo ed al costo di ripristino. Generalmente la frequenza è di tipo:

- giornaliero;
- settimanale;
- mensile;
- annuale (solo per archivi storici).

Affidabilità del Fyle System BackUp

Scelta dei file: la scelta va pianificata in base al contenuto dei file ed al costo di ripristino. Ad esempio:

- i file *critici* potrebbero essere raggruppati in directory dedicate delle quali si effettua copia con maggiore frequenza;

- la duplicazione di alcuni file *critici* può essere opportuna se il quantitativo di memoria sprecato è giustificabile.

Affidabilità del Fyle System BackUp

Tecnica di backup: la scelta va pianificata in base al tempo disponibile ed al costo sopportabile per i supporti. Le tecniche di backup possono essere:

- backup su nastri dell'intero file system;
- backup incrementale (copia solo dei file nuovi o modificati dall'ultimo backup);
- duplicazione incrociata dei dati su due dispositivi: metà di ogni disco contiene i propri dati e l'altra metà contiene i dati dell'altro disco.

Affidabilità del Fyle System Prestazioni

Accesso a disco operazione lenta.

Molti file system prevedono tecniche per ridurre il numero di accessi al disco:

- caching;
- riduzione dei movimenti del braccio del disco (soluzione soft);
- riduzione dei movimenti del braccio del disco (soluzione hard).

Affidabilità del Fyle System
Prestazioni

***block cache**:* un certo numero di blocchi in memoria vengono riservati per contenere blocchi di disco.

- ad ogni richiesta di un blocco viene scandita la tabella dei blocchi in memoria per verificarne la presenza;
- se non è presente il blocco viene caricato dal disco, copiato nella chache e poi reso disponibile alle richieste;
- la lista può essere gestita con normali algoritmi di rimpiazzamento delle pagine come FIFO e LRU;

UNIX ogni 30 secondi riscrive tutti i blocchi modificati su disco (oppure a richiesta dell'utente),

WINDOWS riscrive immediatamente ogni blocco modificato.

Affidabilità del Fyle System Prestazioni

Riduzione dei movimenti del braccio del disco (**soluzione soft**):

- *raggruppamento di blocchi in uso.* Andando a stimare i blocchi a cui si accede con maggiore probabilità in sequenza al fine di ridurre i movimenti della testina.
- *posizione dell'indice dei blocchi* . La lettura di un file richiede comunque almeno 2 accessi (uno all'i-node ed uno al blocco). Il posizionamento degli i-node o della tabella di allocazione dei file al centro del disco per ridurre la distanza tra questi ed i blocchi a cui questi fanno riferimento.
- *partizioni aperte*. Si tende ad avvicinare il più possibile gli i-node ai blocchi cui fanno riferimento nel seguente modo:
  - divisione del disco in gruppi di cilindri, ognuno con i propri i-node e la propria lista dei blocchi liberi;
  - quando una partizione esaurisce i blocchi, un blocco può essere allocato in una partizione diversa.

Affidabilità del Fyle System
Prestazioni

***Interleave*.**

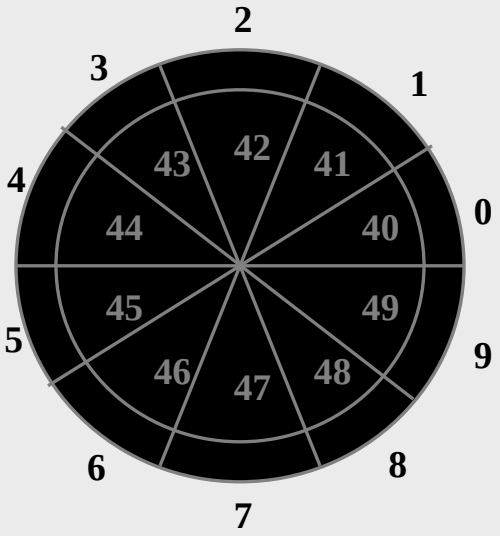
un processo richiede un tempo t_b per richiedere e recuperare un blocco. In situazioni (frequenti) di richieste consecutive di blocchi sequenziali può accadere che:

- il blocco richiesto può essere già passato sotto la testina;
- bisogna attendere una rotazione intera.

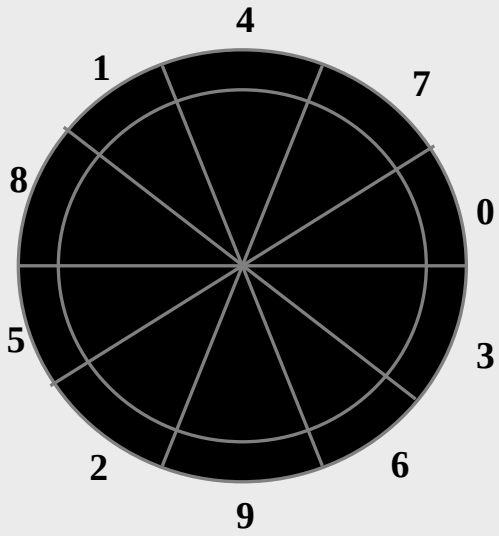
La tecnica di interleaving alloca i blocchi fisici del disco in ordine non sequenziale stretto, ma sequenziale con salti (fattore di interleaving).

Affidabilità del Fyle System
Prestazioni

fattore di interleaving : ¼ di giro



(a) Senza interallacciamento



(b) Con interallacciamento

Affidabilità del Fyle System
Prestazioni

La riduzione dei movimenti del braccio del disco (**soluzione hard**) avviene mediante la *schedulazione del braccio del disco*. Il tempo necessario per leggere un blocco di disco dipende da:

- seek time: spostamento della testina sul cilindro;
- latency time: tempo di rotazione per raggiungere il settore richiesto;
- transfer time: tempo di trasferimento.

Affidabilità del Fyle System Prestazioni

Il ritardo predominante è dato dal seek time.

Algoritmi per ottimizzare il movimento del braccio:

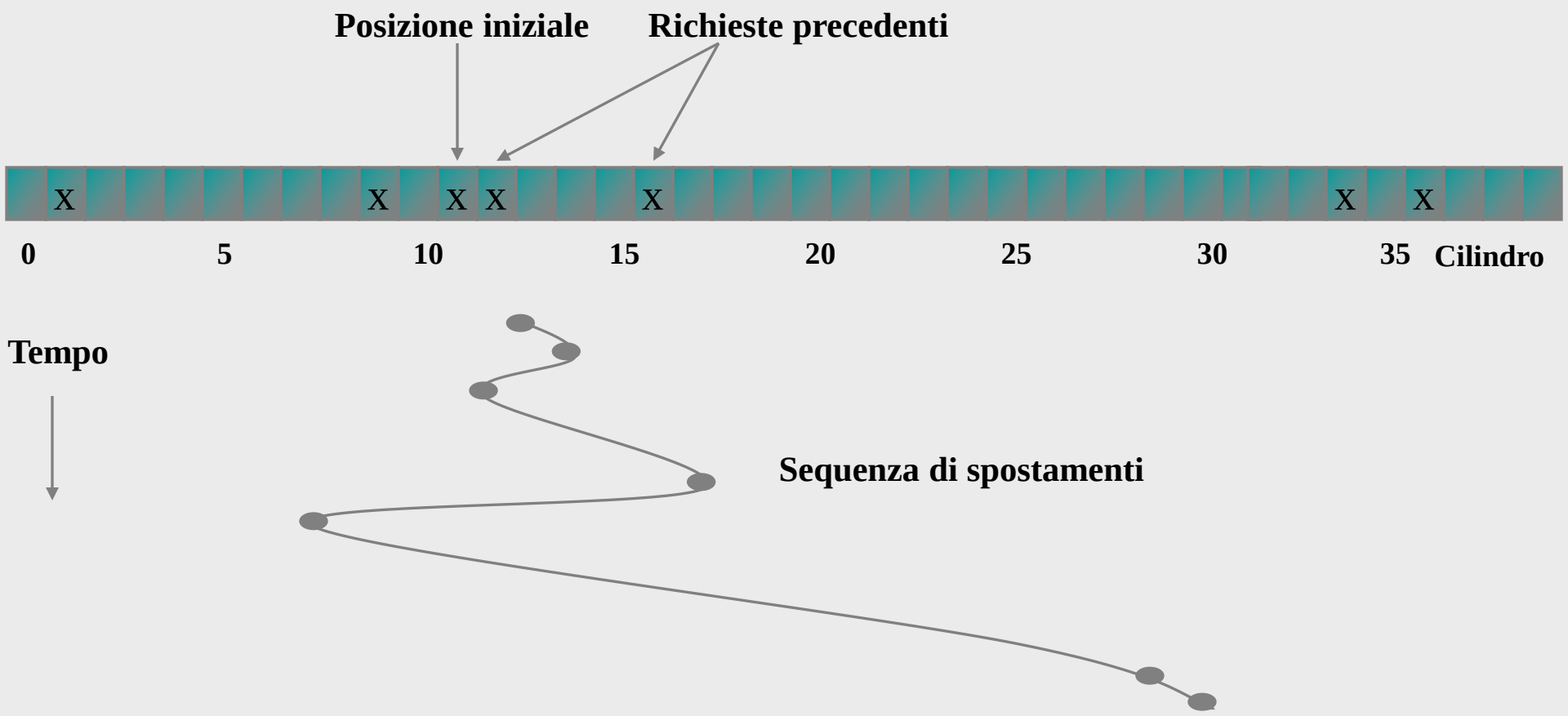
- FCFS (First Come First Served):
  - serve le richieste nell'ordine in cui giungono;
  - non richiede nessun specifico supporto hardware o software;
  - è l'algoritmo più corretto nei riguardi delle richieste
  - non ottimale;

Affidabilità del Fyle System
Prestazioni

- SSF (Shortest Seek First):
  - mantenendo in tabella tutte le richieste pendenti, l'SSF si sposta sempre sulla richiesta più vicina a quella appena servita;
  - riduce di circa la metà gli spostamenti sul disco rispetto al FCFS;
  - presenta un problema di localizzazione delle richieste.

Affidabilità del Fyle System
Prestazioni

Algoritmo SSF per la schedulazione dei movimenti del braccio

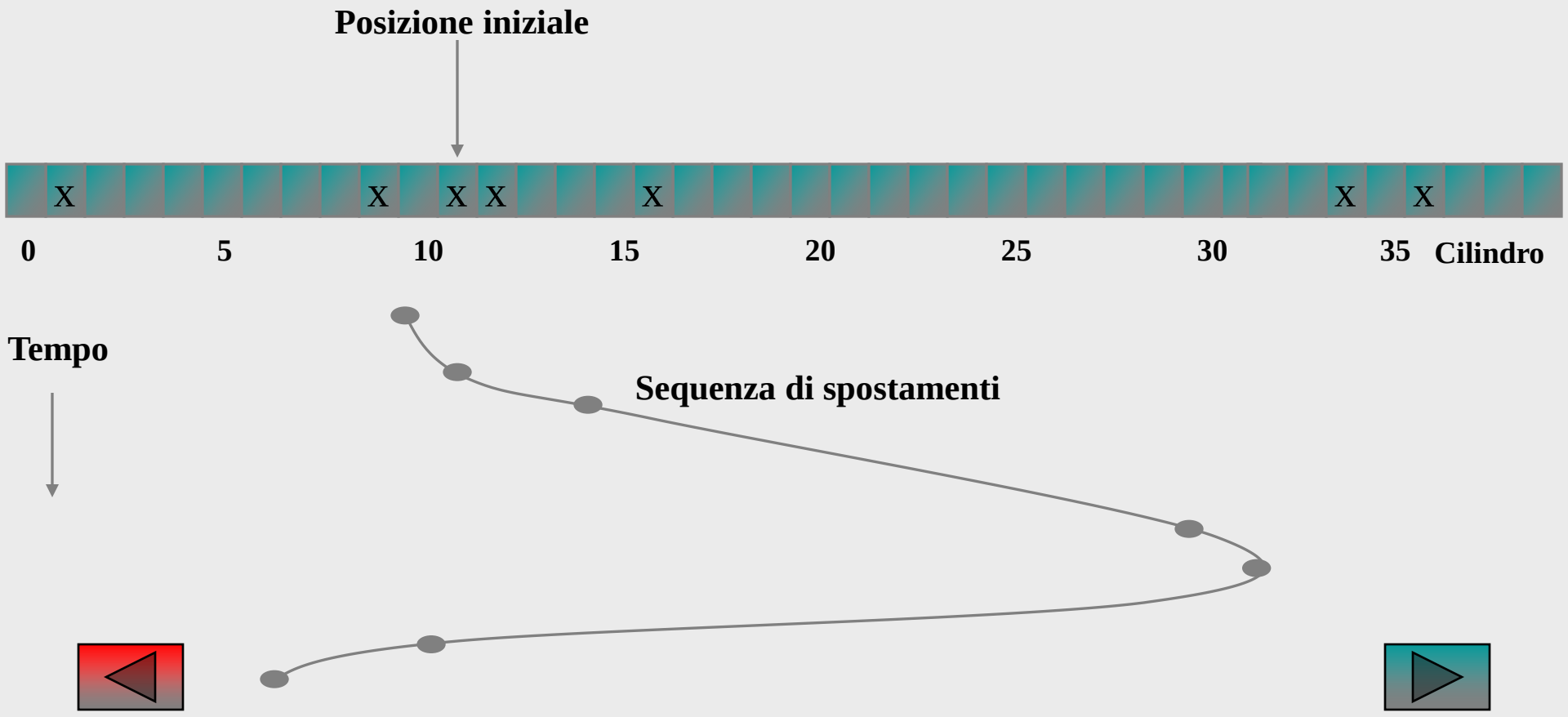


Affidabilità del Fyle System
Prestazioni

- algoritmo dell'ascensore:
  - mantiene in tabella le richieste pendenti;
  - un bit *up/down* indica la direzione attuale del braccio del disco;
  - serve le richieste più vicine proseguendo sempre nella stessa direzione;
  - cambia direzione quando raggiunge l'estremità o quando non ci sono più richieste nella direzione attuale;

Affidabilità del Fyle System
Prestazioni

Algoritmo dell'ascensore



Identificazione degli utenti

I meccanismi di protezione degli archivi sono basati sul presupposto della sicura identificazione degli utenti.

- Nei sistemi interattivi multi-utente, il progenitore di tutti i processi di un generico utente è quello che viene generato quando esso si connette al sistema attraverso un terminale.
- Per ogni processo diverso dal progenitore, la responsabilità di definire il proprietario spetta esclusivamente al sistema operativo
- la sicurezza degli archivi è fondata sulla corretta definizione del proprietario del processo progenitore, che a sua volta dipende dalla sicura identificazione dell'utente che si connette attraverso il terminale.
- La tecnica comunemente usata è quella della parola d'ordine (password),