Il File System

Il file system

È quella parte del Sistema Operativo che fornisce i meccanismi di accesso e memorizzazione delle informazioni (programmi e dati) allocate in memoria di massa.

Realizza i concetti astratti:

- **di file: unità logica di memorizzazione**
- **di direttorio: insieme di file (e direttori)**
- **di partizione: insieme di file associato ad un particolare dispositivo fisico (o porzione di esso)**

- **Le caratteristiche di file, direttorio e partizione sono del tutto indipendenti dalla natura e dal tipo di dispositivo utilizzato.**

Il file

È un insieme di informazioni:

- programmi
- dati
- testi
- …

rappresentati come insieme di ***record logici*** (bit, byte, linee, record, etc.)

- Ogni file è individuato da (almeno) un **nome** simbolico mediante il quale può essere riferito (ad esempio, nell'invocazione di comandi o system call).
- Ogni file è caratterizzato da un insieme di **attributi**.

Attributi del file

- A seconda del sistema operativo, i file possono avere attributi diversi. Solitamente:
  - **tipo**: stabilisce l'appartenenza a una classe (eseguibili, batch, testo, etc)
  - **indirizzo**: puntatore/i a memoria secondaria
  - **dimensione**: numero di byte contenuti nel file
  - **data e ora** (di creazione e/o di modifica)

- In S.O. multiutente anche:
  - utente **proprietario**
  - **protezione**: diritti di accesso al file per gli utenti del sistema

Attributi del file

***Descrittore* del file:**

è la struttura dati che contiene gli attributi di un file.

Ogni descrittore di file deve essere memorizzato in modo persistente:

- il S.O. mantiene l'insieme dei descrittori di tutti i file presenti nel file system in apposite strutture in memoria secondaria (ad es. Unix: ***i-list***)

Operazioni sui file

Compito del S.O. è consentire l'accesso *on-line* ai file.

Tipiche Operazioni:

- **Creazione** : allocazione di un file in memoria secondaria ed inizializzazione dei suoi attributi.
- **Lettura** di record logici dal file.
- **Scrittura**: inserimento di nuovi record logici all'interno di file.
- **Cancellazione**: eliminazione del file dal file system.

➢ Ogni operazione richiederebbe la localizzazione di informazioni su disco; ad esempio:

- gli indirizzi dei record logici a cui accedere
- gli altri attributi del file
- i record logici

-> costo elevato!

Operazioni sui file

Per migliorare l'efficienza:

- il S.O. mantiene in memoria una struttura che registra i file attualmente in uso (file ***aperti***): **tabella dei file aperti**. :
  - per ogni file aperto: **{puntatore al file, posizione su disco,..}**
- Spesso viene fatto il *memory mapping* dei file aperti:

  i file aperti (o porzioni di essi) vengono temporaneamente copiati in memoria centrale -> accessi più veloci.

Operazioni necessarie:

- **Apertura** : introduzione di un nuovo elemento nella tabella del file aperti e eventuale memory mapping del file.
- **Chiusura**: salvataggio del file in memoria secondaria e eliminazione dell'elemento corrispondente dalla tabella dei file aperti.

Struttura interna dei file

Ogni dispositivo di memorizzazione secondaria viene partizionato in **blocchi** (o ***record fisici***):

Blocco: unità di trasferimento nelle operazioni di I/O da/verso il dispositivo; la sua dimensione è fissa.

L'utente vede il file come un insieme di **record logici:**

Record logico: unità di trasferimento nelle operazioni accesso al file (es. lettura, scrittura di blocchi); la sua dimensione può variare.

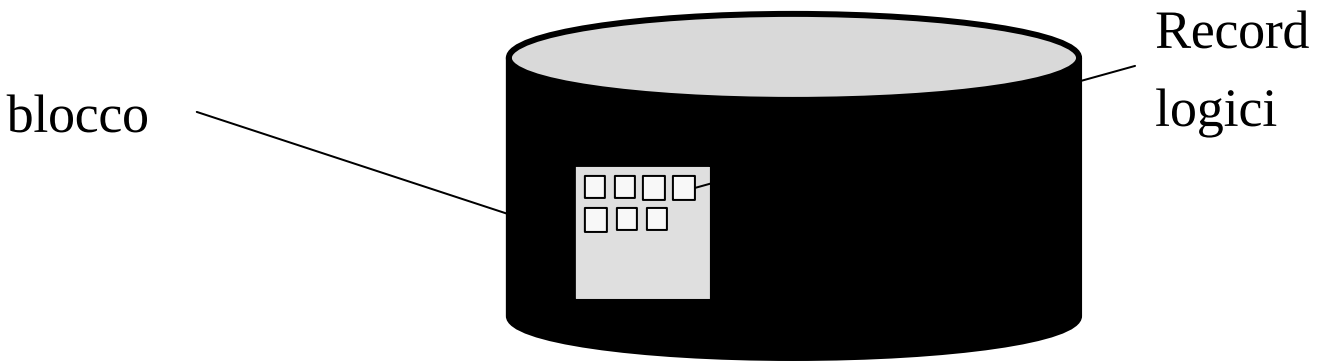
Blocchi & record logici

- Uno dei compiti del Sistema Operativo (del file system) è stabilire una **corrispondenza tra record logici e blocchi**.

Di solito:

Dimensione(blocco) >> Dimensione(record logico)

- *impaccamento* di record logici all'interno di blocchi.



Metodi di accesso

L’accesso a file può avvenire secondo varie modalità:

- accesso **sequenziale**
- accesso **diretto**
- accesso a **indice**

➢ Il metodo di accesso è indipendente:

- **dal tipo di dispositivo** utilizzato;
- **dalla tecnica di allocazione** dei blocchi in memoria secondaria.

Accesso sequenziale

Il file è una **sequenza** [R_1, R_2,.. R_N] di record logici:

- per accedere ad un particolare record logico R_i , è necessario accedere prima agli (*i-1*) record che lo precedono nella sequenza:

- in questo caso le operazioni di accesso sono del tipo:
  - ***readnext***: lettura del prossimo record logico della sequenza
  - ***writenext***: scrittura del prossimo record logico

- ogni operazione di accesso (lettura/scrittura) posiziona il **puntatore al file** sull'elemento successivo a quello letto/scritto.

- **Unix** prevede accesso sequenziale.

Accesso diretto

Il file è un **insieme non ordinato** $\{R_1, R_2, .. R_N\}$ di record logici numerati:

- si può accedere direttamente ad un particolare record logico specificandone il numero.

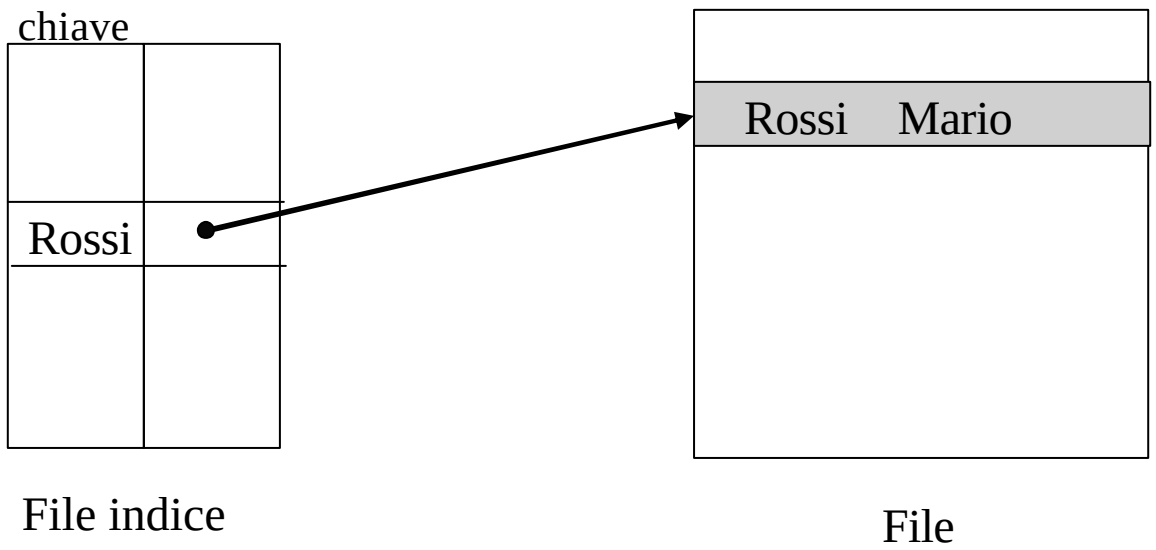
  - in questo caso le operazioni di accesso sono del tipo:
    - ***read i***: lettura del record logico i
    - ***write i***: scrittura del record logico i

- Utile, quando si vuole accedere a grossi file, per estrarre/aggiornare poche informazioni (es. database).

Accesso a indice

Ad ogni file viene associata una struttura dati (ancora un file!) contenente l'***indice*** delle informazioni contenute nel file.

- per accedere a un record logico, si esegue una ricerca nell'indice (utilizzando una *chiave*):



Il direttorio

È lo strumento per organizzare i file all'interno del file system:

- un direttorio può contenere più files
- è realizzato mediante una struttura dati che associa al nome di ogni file (in esso contenuto) la posizione del file nel disco.

Operazioni sui direttori:

- **Creazione** e **cancellazione** di direttori
- **Aggiunta/cancellazione** di file.
- **Listing**: elenco di tutti i file contenuti nel direttorio.
- **Attraversamento** del direttorio.
- **Ricerca** di file nel direttorio.

Tipi di direttorio

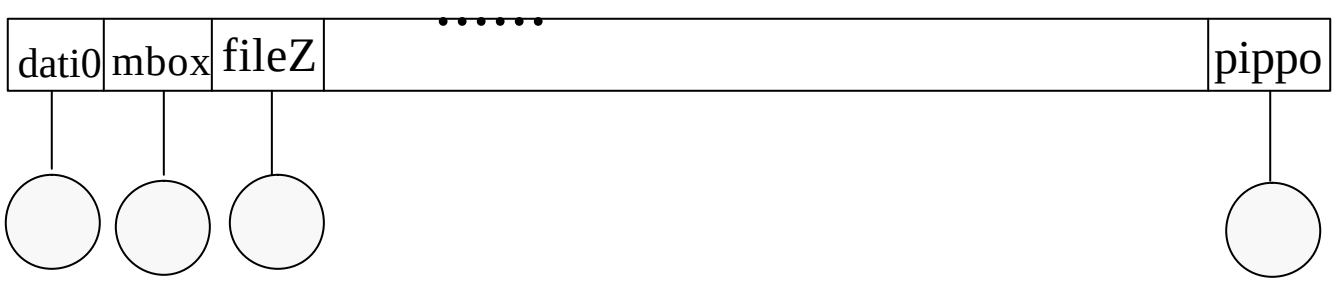
La struttura logica del direttorio può variare a seconda del Sistema Operativo

Schemi più comuni:

- a un **livello**
- a **due livelli**
- ad **albero**
- a **grafo aciclico**

Tipi di direttorio

Struttura a un livello: un solo direttorio per ogni file system;



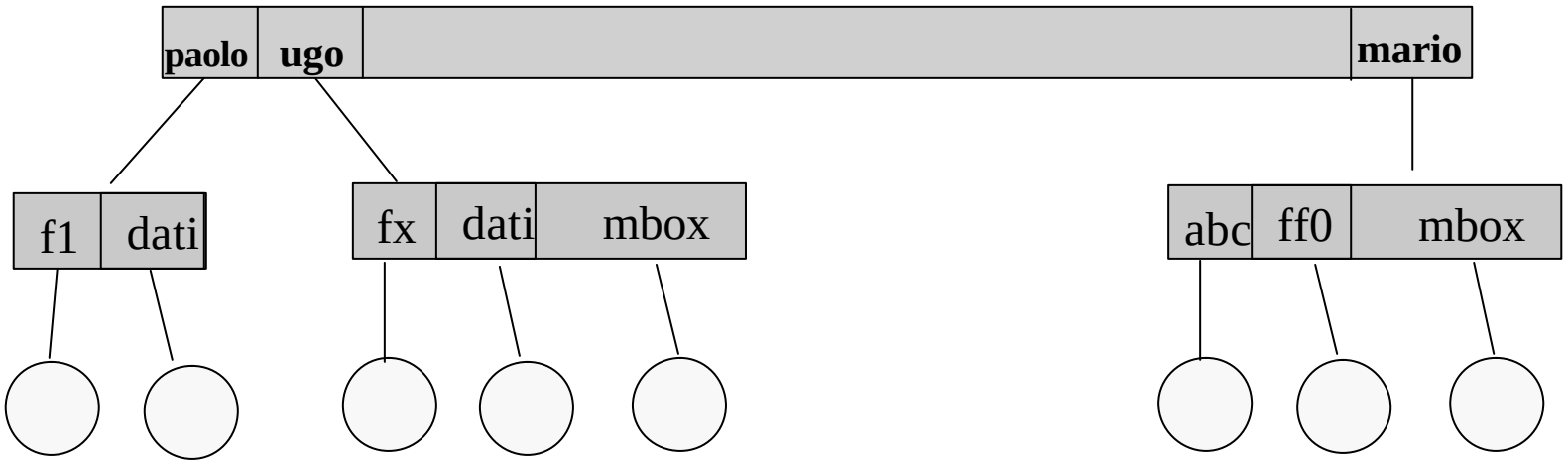
Problemi:

- ✓ unicità dei nomi
- ✓ multiutenza: come separare i file dei diversi utenti?

Tipi di direttorio

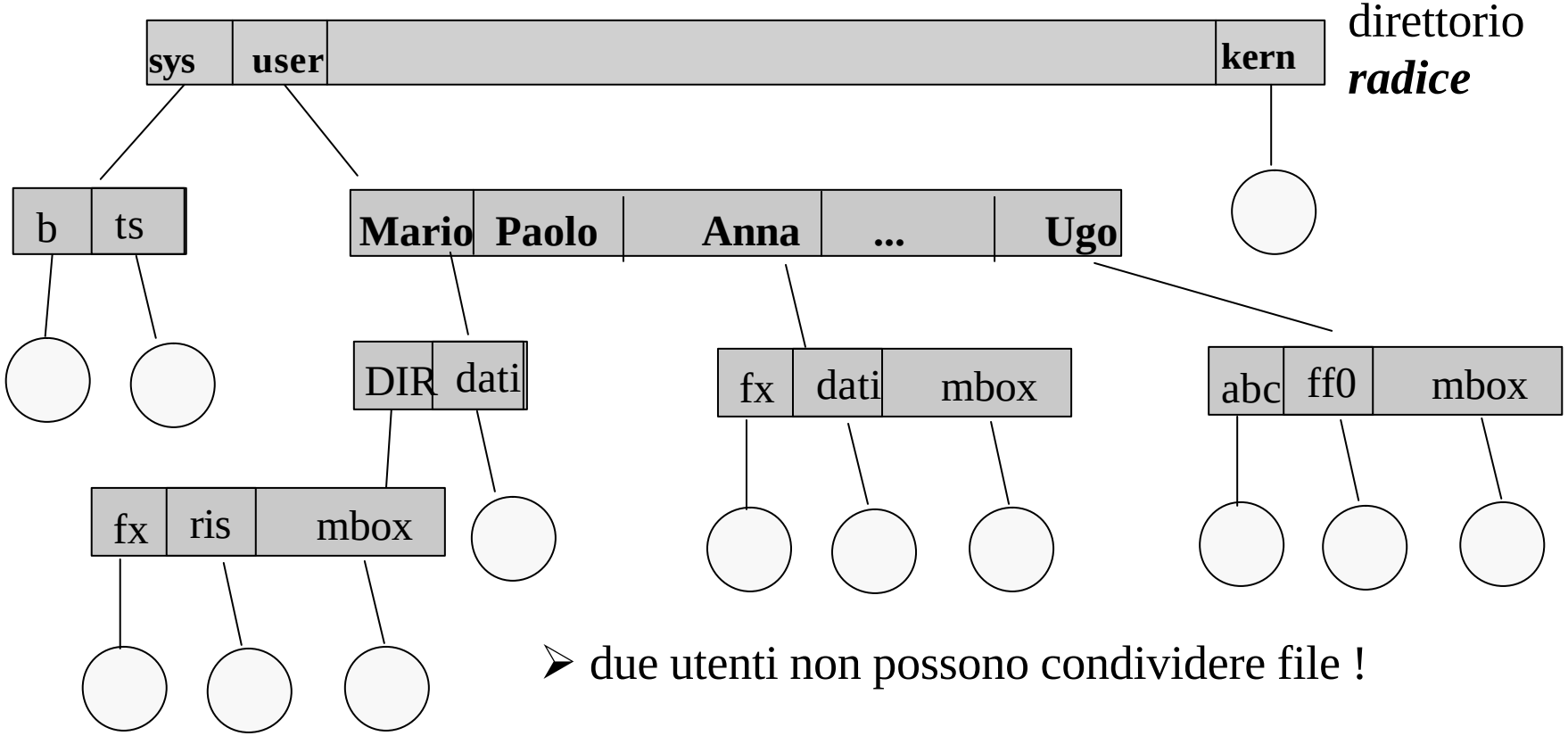
Struttura a due livelli:

- primo livello (direttorio principale): contiene un direttorio per ogni utente del sistema.
- Secondo livello: direttori utenti (a un livello).



Tipi di direttorio

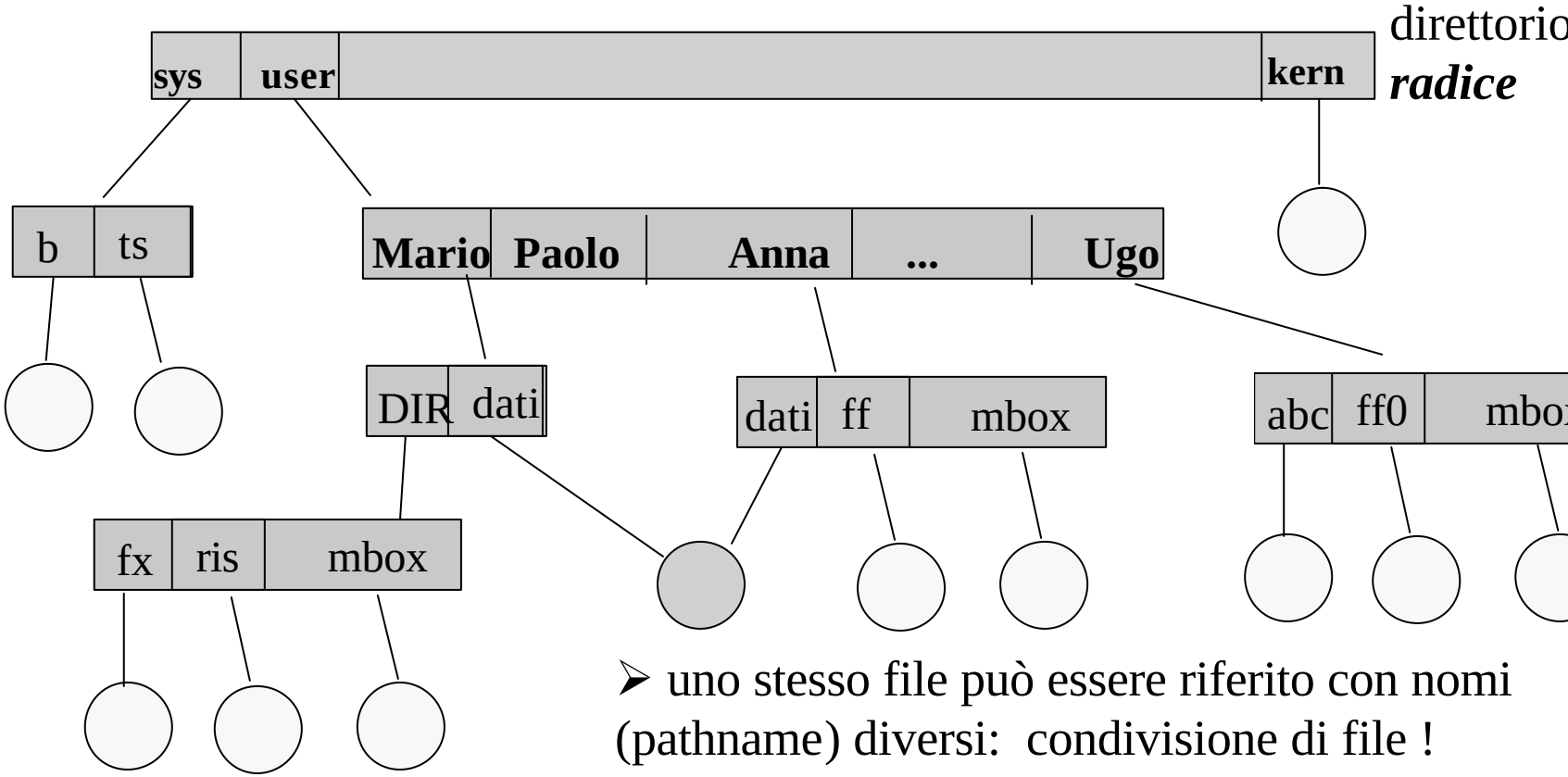
Struttura ad albero: organizzazione gerarchica a N livelli. Ogni direttorio può contenere file e altri direttori.



- due utenti non possono condividere file !

Tipi di direttorio

Struttura a grafo aciclico (es. UNIX): estende la struttura ad albero per permettere la **condivisione di file e direttori.**



- uno stesso file può essere riferito con nomi (pathname) diversi: condivisione di file !

Realizzazione del file system

Il sistema operativo si occupa della rappresentazione del file system sui dispositivi di memorizzazione di massa:

- realizzazione dei descrittori e loro organizzazione
- allocazione dei blocchi fisici
- gestione dello spazio libero

Come può essere rappresentato il file system sul disco?

Metodi di Allocazione

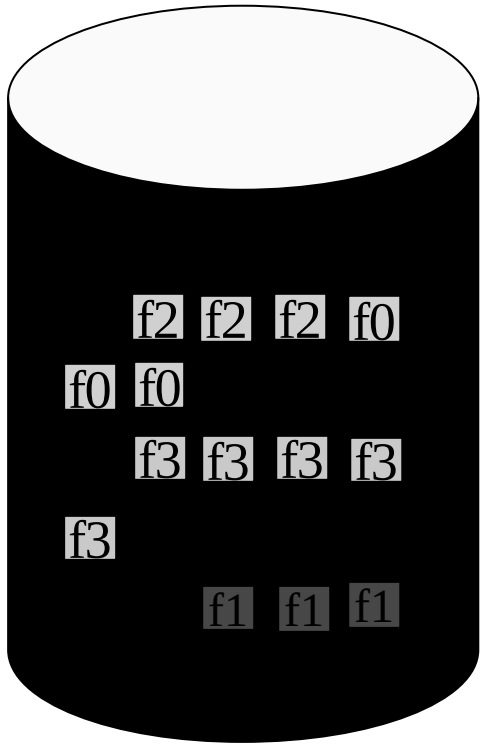
Ogni **blocco** contiene un insieme di **record logici contigui.**

Quali sono le tecniche più comuni per l'allocazione dei blocchi sul disco?

- allocazione **contigua**
- allocazione a **lista**
- allocazione a **indice**

Allocazione contigua

Ogni file è mappato su un insieme di blocchi fisicamente contigui.



Vantaggi:

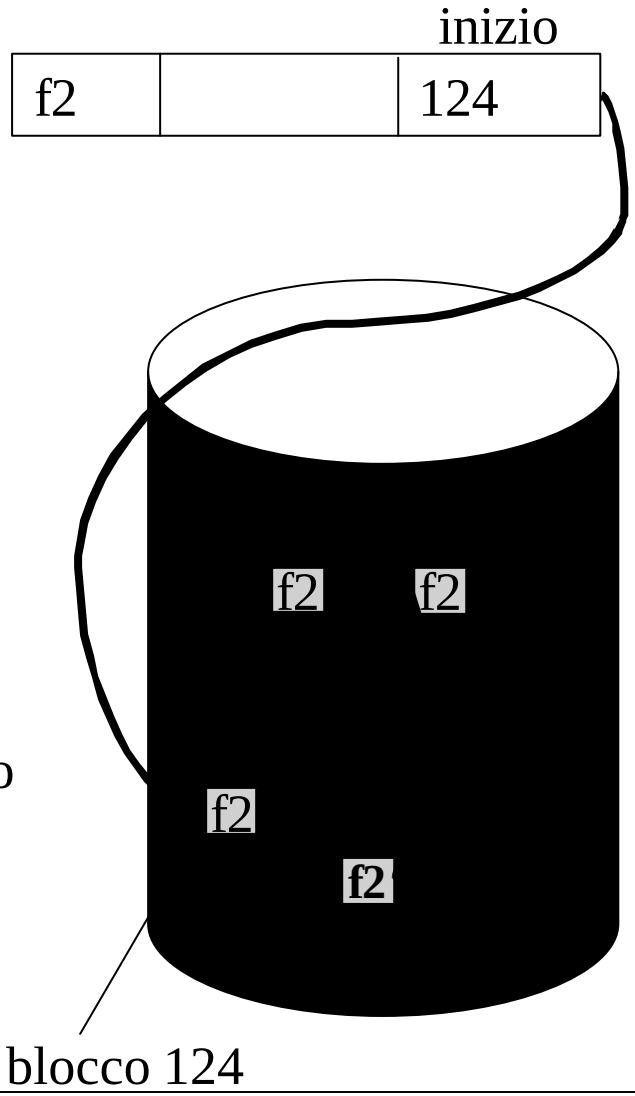
- costo della ricerca di un blocco
- possibilità di accesso sequenziale e diretto

Svantaggi:

- individuazione dello **spazio libero** per l'allocazione di un nuovo file
- **Frammentazione** esterna: man mano che si riempie il disco, rimangono zone contigue sempre più piccole, a volte inutilizzabili:
  - **compattazione**
- **Aumento** dinamico delle **dimensioni** di file

Allocazione a lista (concatenata)

I blocchi sui quali viene mappato ogni file sono organizzati in una lista concatenata.



Vantaggi:

- non c'è frammentazione esterna
- minor costo di allocazione

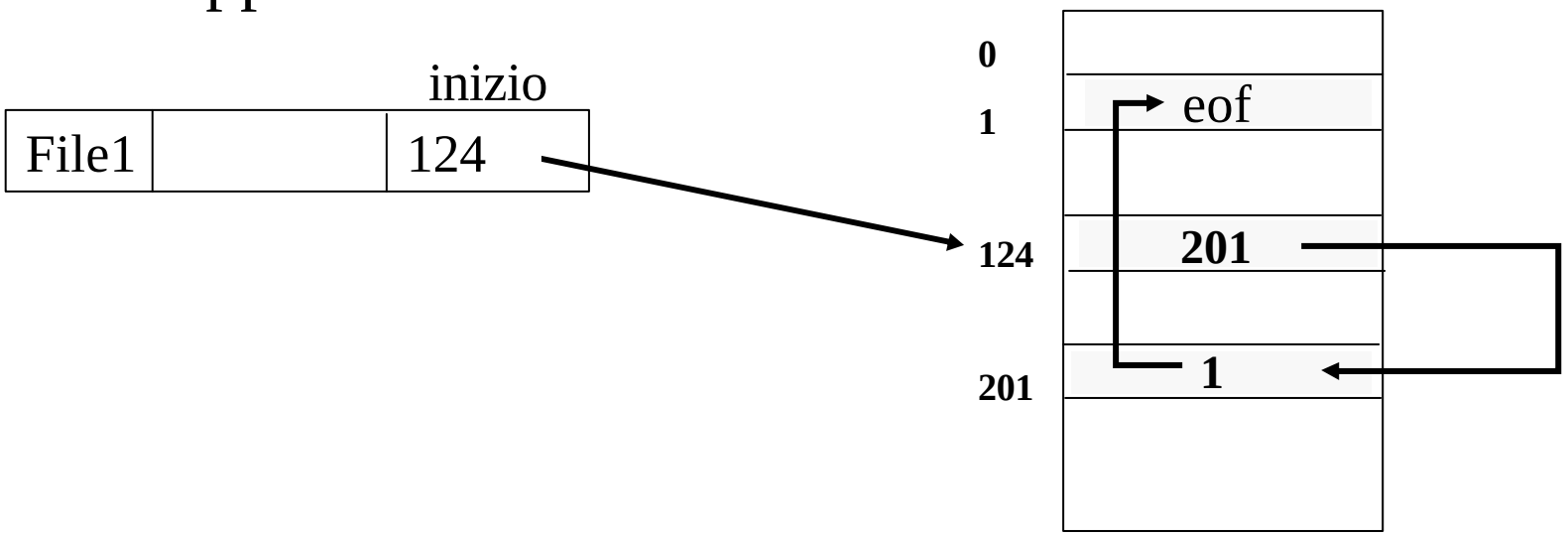
Svantaggi:

- possibilità di errore se un link viene danneggiato
- maggior occupazione (spazio occupato dai puntatori)
- difficoltà di realizzazione dell'accesso diretto
- costo della ricerca di un blocco

Tabella di allocazione dei file (FAT)

Alcuni sistemi operativi (ad es. DOS e OS/2) realizzano l'allocazione a lista in modo più efficiente e robusto:

- per ogni partizione, viene mantenuta una tabella (FAT) in cui ogni elemento rappresenta un blocco fisico.
- Il concatenamento dei blocchi sui quali è allocato un file è rappresentato nella FAT:



Allocazione a indice

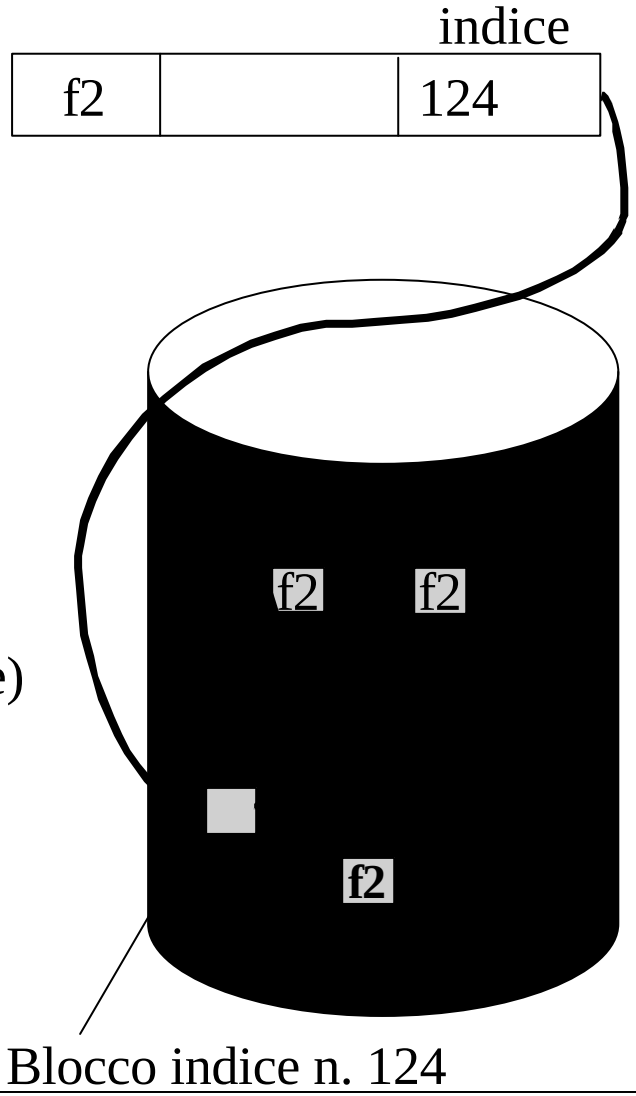
Allocazione a lista: i puntatori ai blocchi sono **distribuiti** sul disco:

- il tempo medio di accesso a un blocco è elevato
- complessità della realizzazione del metodo di accesso diretto

Allocazione a indice: i puntatori ai blocchi utilizzati per l'allocazione di un file sono **concentrati** in un blocco (***blocco indice***)

Allocazione a indice

A ogni file è associato un **blocco** (***indice***) in cui sono contenuti tutti gli indirizzi dei blocchi su cui è allocato il file.



Vantaggi:

- gli stessi dell'allocazione a lista, più
  - possibilità di accesso diretto
  - maggiore velocità di accesso (rispetto a liste)

Svantaggi:

- scarso utilizzo dei blocchi indice

Metodi di Allocazione

- Riassumendo, gli aspetti caratterizzanti sono:
  - grado di **utilizzo** della memoria
  - tempo di **accesso** medio al blocco
  - realizzazione dei **metodi di accesso**

  - **Esistono sistemi operativi che adottano più di un metodo di allocazione; spesso:**
    - file **piccoli** ➔ allocazione contigua
    - file **grandi** ➔ allocazione a indice