

# Il File System

Giuseppe Pozzi

Impianti di Elaborazione  
Facoltà di Ingegneria dell'Informazione  
Politecnico di Milano

giuseppe.pozzi@polimi.it  
- versione del 22 ottobre 2003 -

# File System

## • Bibliografia:

Pelagatti G.  
"Sistemi di Elaborazione – Architetture hardware e software",  
McGraw-Hill, 1992 – pag. 231-240

## • Sommario:

- I file.
- Modalità di accesso.
- I volumi e Unix.

22 ottobre 2003

Impianti di Elaborazione - File system

2

## Usi della memoria di massa

- Conservare programmi e dati quando il calcolatore non è alimentato (memoria non volatile o persistente).
- Memorizzare i dati generati da un processo dopo la sua terminazione e consentirne l'uso ad altri processi (memoria a "lunga vita").
- Archiviare grandi quantità di dati e informazioni.
- Consentire a diversi processi di accedere agli stessi dati contemporaneamente (accesso concorrente).

File: unità di archiviazione in memoria di massa (long-lived object).

22 ottobre 2003

Impianti di Elaborazione - File system

3

## Introduzione

- **File system**: componente del sistema operativo che realizza i servizi di gestione dei **file**
- File system svolge le sue funzioni accedendo ai dischi tramite l'invocazione di routine messe a disposizione dal gestore del disco (**disk driver**)
- Il driver del disco fornisce una interfaccia di accesso ai dati su disco nascondendone le caratteristiche fisiche
- Il disco è rappresentato da un dispositivo logico, chiamato **volume** composto da un array di **blocchi**
- Il **blocco** è porzione di disco costituita da un numero intero di settori che viene trasferita in memoria con un'unica operazione

22 ottobre 2003

Impianti di Elaborazione - File system

4

# File

- Il concetto di *file* offre una visione logica omogenea delle informazioni memorizzate
- La visione non dipende dal tipo di dispositivo fisico su cui le informazioni vengono memorizzate
- Un *file* è costituito da:
  - Un insieme di informazioni omogenee
  - Un nome simbolico
  - Un insieme di attributi
- Un *file* può contenere:
  - Dati
  - Programmi
  - Riferimenti

# Attributi (i)

- Ad un file sono associati alcuni **attributi** che ne descrivono alcune caratteristiche:
  - *Nome:*
    - E' un nome simbolico con cui ci si riferisce ad esso
  - *Tipo:*
    - Definisce il tipo dei dati contenuti (estensione del nome)
  - *Locazione:*
    - E' un puntatore alla posizione fisica sul dispositivo

# Attributi (ii)

- *Dimensione:*
  - Dimensione dei dati espressa in bytes o blocchi
- *Protezione:*
  - Definisce le politiche di gestione degli accessi
- *Ora e Data:*
  - Indicano il momento della creazione, ultima modifica o ultimo accesso
- *Proprietario:*
  - Indica il nome dell'utente che ha creato il file

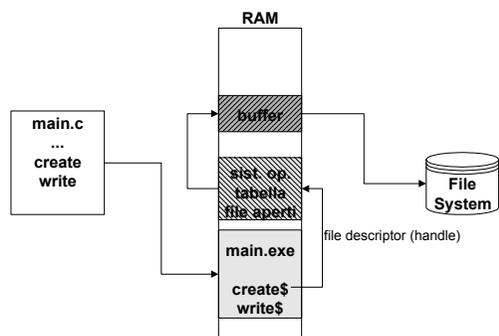
# Operazioni su file

- Sui file possono essere compiute diverse operazioni
- Le operazioni vengono svolte attraverso richieste di servizi al sistema operativo

## *Creazione*

- Viene aggiunto un nuovo file al **file system**
- Le operazioni richieste sono:
  - Allocazione
  - Creazione del nuovo descrittore del file
  - Aggiunta del descrittore al file system

## Creazione di un file



## Operazioni su file

- **Scrittura:**
  - Aggiunge dati ad un file già creato
  - Per scrivere dati su un file è necessario fornire:
    - Il nome del file
    - I dati da scrivere
  - Il file system mantiene un puntatore alla posizione in cui deve essere effettuata la scrittura successiva
- **Letture:**
  - Preleva dati da un file già creato
  - Per leggere dati da un file è necessario fornire:
    - Il nome del file
    - Un puntatore alla zona di memoria destinata a contenere i dati
  - Il file system mantiene un puntatore alla posizione in cui deve essere effettuata la lettura successiva

## Operazioni su file

- **Riposizionamento:**
  - Sposta la posizione dei puntatori di lettura e di scrittura
  - Le operazioni consentite dipendono dal tipo di accesso al file
  - Spesso viene mantenuto dal file system un solo puntatore valido per lettura e per scrittura
- **Cancellazione:**
  - Elimina un file
  - Per eliminare un file è necessario specificarne il nome
  - Le operazioni necessarie sono:
    - Deallocazione dello spazio sul dispositivo fisico
    - Aggiunta dello spazio deallocato alla lista dello spazio disponibile sul dispositivo
    - Rimozione del descrittore del file dal file system

## Gestione dei file

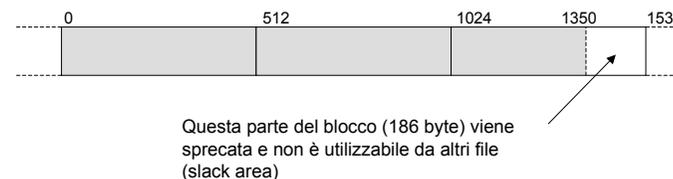
- Alcuni dei servizi forniti dal sistema operativo:
  - **Open:**
    - Sulla base del nome individua la posizione del file sul disco
    - Copia il descrittore del file nella tabella dei file aperti
  - **Close:**
    - Sulla base del nome o di un identificatore localizza il descrittore del file nella tabella dei file aperti
    - Elimina il descrittore dalla tabella dei file aperti

## Struttura del file system

- **Struttura logica:**
  - I dati sono organizzati in unità logiche di lunghezza fissa ma arbitraria dette *record*
  - Nel sistema UNIX un *record* è un byte
- **Struttura fisica:**
  - I dati sono organizzati in unità fisiche di lunghezza fissa e dipendente dal dispositivo dette *blocchi*
  - Dimensioni tipiche dei blocchi di unità a disco rigido: da 32 a 4096 byte, tipicamente 512
- La struttura logica e fisica sono in genere differenti

## Struttura

- Tale differenza, unitamente alla dimensione fissata dei blocchi provoca uno spreco di spazio
- Esempio: file di 1350 byte e disco con blocchi da 512 byte



- Questo fenomeno è noto come *frammentazione interna*

## Organizzazione dei file

- I file sono organizzati principalmente in due modi:
- **Accesso sequenziale:**
  - I dati vengono letti e scritti in sequenza a partire dall'inizio del file
  - Le operazioni disponibili per tali file sono:
    - Lettura e scrittura
    - Posizionamento all'inizio o alla fine del file
    - Posizionamento sul record precedente o successivo

## Organizzazione dei file

- **Accesso diretto o casuale:**
  - I dati vengono letti e scritti in una qualsiasi posizione specifica
  - La posizione deve essere specificata in termini di blocco logico, relativamente all'inizio del file
  - I blocchi logici devono avere dimensione fissa per consentire il calcolo della posizione effettiva dei dati

# File system

- I moderni dispositivi di memorizzazione di massa consentono di salvare milioni di file
  - Necessaria una strutturazione gerarchica
- Spesso sorge la necessità di condividere un file o un gruppo di file tra più utenti
- Un file system è organizzato in:
  - **Partizioni:** Contengono insieme di file correlati
  - **Directory:** Una partizione è suddivisa in directory. Le directory contengono informazioni sui file e fungono da indice.
  - **File:** Contengono effettivamente i dati o i programmi

# Protezione

- I dati memorizzati nei file di un file system necessitano di **protezione**
- Protezione da **danni fisici:**
  - Malfunzionamenti dei dispositivi
  - Danni meccanici e/o elettrici
- **Soluzione:** **backup** e **mirroring**
- Protezione da **accessi impropri:**
  - Riservatezza
  - Modifica o eliminazione accidentale di dati importanti
- **Soluzione:** definizione di una **politica di accesso**  
⇒ **Protezione:** *definizione di una politica di accesso e relativa implementazione*

# Protezione: Tipi di accesso

- Alcune banali politiche di accesso sono:
  - *Ogni utente accede solo ai propri file:* si tratta di una scelta limitante, ad esempio per i gruppi di lavoro
  - *Ogni utente accede a tutti i file:* è assente una politica di accesso
- La soluzione consiste nell'**accesso controllato**

# Regole di accesso ai file

- Si definiscono **regole di accesso** ai file sulla base di:
  - Identità e gruppo di lavoro dell'utente
  - Proprietà dei file
- Tali regole dipendono dal tipo di operazione richiesta:
  - Lettura
  - Scrittura o eliminazione
  - Esecuzione o lista
  - Aggiunta

## Protezione: Liste di accesso

- L'accesso e le operazioni consentite dipendono dalla identità dell'utente
- Ad ogni file è associata una *lista di accesso* che indica quali *operazioni* sono consentite a quali *utenti*
- Quando una operazione viene richiesta il sistema operativo controlla la lista di accesso per verificare se:
  - Il richiedente è contemplato
  - Il richiedente ha il permesso di compiere quel tipo di operazione

## Liste di accesso

- La soluzione basata su liste di accesso presenta alcuni svantaggi:
  - Le liste di accesso possono essere di dimensioni notevoli
  - devono essere create e mantenute per ogni file
  - Il tempo di accesso ad un file si allunga

## Protezione: UNIX

- Soluzione semplificata implementata in UNIX
- Utenti identificati in base a:
  - *Username*: Identificativo dell'utente
  - *Group*: Identificativo di gruppo, condiviso da più utenti
- Utenti raggruppati in tre classi:
  - *Owner*: Solo il proprietario del file
  - *Group*: Solo i membri del gruppo del proprietario del file
  - *All*: Tutti gli utenti

## Protezione: UNIX

- Le operazioni su file sono raggruppate in tre classi:
  - *Read*: Lettura, copia
  - *Write*: Scrittura, modifica, eliminazione
  - *Execute*: Esecuzione
- Ad ogni file sono associati:
  - *Owner*
  - *Group*
  - *Control access list*

## Protezione: UNIX

- La *control access list* è formata da tre gruppi di bit:
  - Ogni gruppo di bit si riferisce ad una delle tre classi di utenti
  - Ogni bit del gruppo si riferisce ad una delle tre operazioni



## Allocazione: UNIX *i-node*

- Il sistema operativo UNIX utilizza uno schema di allocazione realizzato mediante una tabella (TOC - table of contents) i cui elementi sono detti *i-node* (*index-node*)
- Ogni *i-node* contiene la lista degli attributi e degli indirizzi dei blocchi di disco a cui sono associati i blocchi del file
- I primi indirizzi dei blocchi di disco sono memorizzati nell'*i-node* stesso

## Implementazione di file e directory in UNIX

- Ogni file o directory ha associato un *i-node*
- *I-node*: 64 byte di informazioni
- *I-node* dei file in un disco sono memorizzati in sequenza numerica all'inizio del disco o di ogni cilindro
- Dato il numero dell'*i-node* UNIX può localizzare la sua posizione calcolandone l'indirizzo su disco

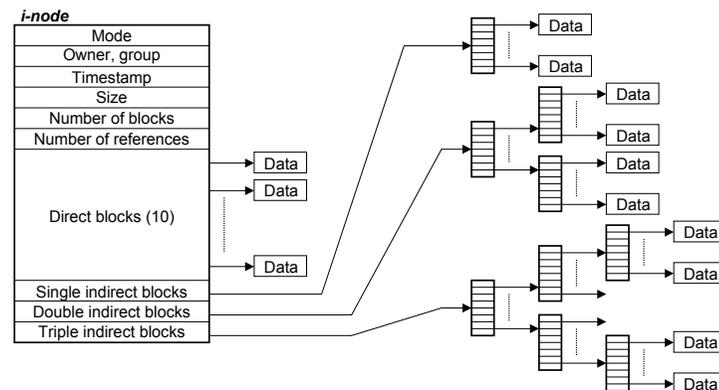
## Informazioni nell'*i-node*

- Tipo di file, bit di protezione (9 rwx) e altri bit
- Numero di link al file
- Proprietario e gruppo del proprietario
- Lunghezza del file in byte
- 13 indirizzi su disco (blocchi su disco contenenti il file)
- Data e ora in cui il file è stato letto e scritto per l'ultima volta
- Data e ora dell'ultima modifica dell'*i-node*

# UNIX *i-node*

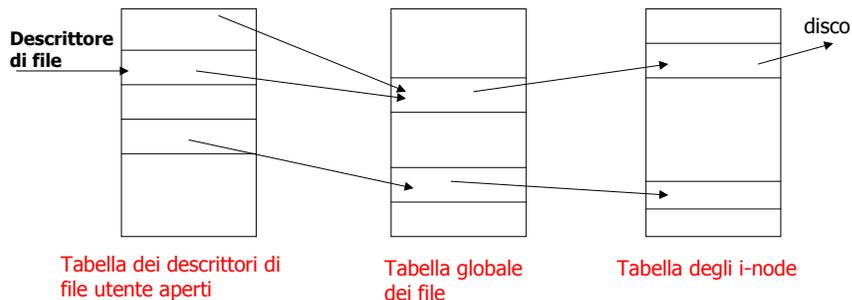
- Per file di dimensioni maggiori l'i-node contiene l'indirizzo di un blocco del disco chiamato **single indirect block** che a sua volta contiene altri indirizzi di blocchi del disco
- Se questo non è sufficiente l'i-node mette a disposizione un altro indirizzo, chiamato **double indirect block** che contiene l'indirizzo di un blocco che a sua volta contiene una lista di single indirect block
- Ognuno di questi blocchi punta a sua volta a qualche centinaia (128 di solito) di single indirect block
- Esiste anche un **triple indirect block** nel caso la doppia indicizzazione non sia sufficiente

# File system UNIX



## Strutture dati del file system

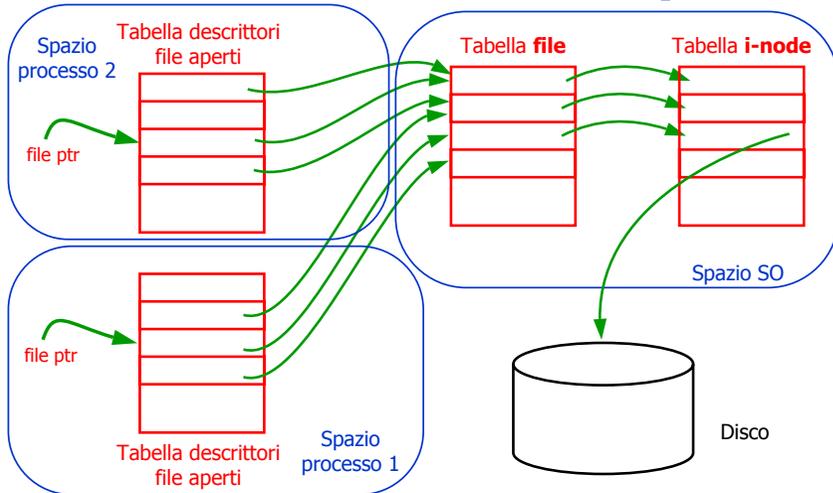
UNIX contiene tre tabelle per la gestione dei file



## Strutture dati del file system

- **Tabella dei descrittori di file utente:**
  - tabella associata ad ogni processo utente contenente una riga per ogni file aperto dal processo con l'indirizzo della riga della tabella globale dei file aperti relativa al file.
- **Tabella globale dei file aperti:**
  - tabella del sistema operativo che contiene una riga per ogni file aperto nel sistema.
  - Ogni riga contiene l'indirizzo del corrispondente i-node nella tabella degli i-node, l'indicatore della posizione corrente del file e il contatore al numero di riferimenti da parte dei processi a questo file.
- **Tabella degli i-node:**
  - Le righe contengono la copia in memoria degli i-node del volume per maggiore efficienza nei riferimenti.

# Strutture dati del file system



22 ottobre 2003

Impianti di Elaborazione - File system

33

# Gestione dello spazio libero

- All'atto della creazione e scrittura di un nuovo file è necessario individuare sul disco il primo blocco libero
- Una soluzione consiste nell'uso un *vettore di bit* in cui:
  - La posizione del bit indica il numero del blocco
  - Se il bit vale 0 il blocco è già utilizzato
  - Se il bit vale 1 il blocco è disponibile

22 ottobre 2003

Impianti di Elaborazione - File system

34