



## File System



## Contenuti

- Funzione di un file system
- Interfacce
- Strutturazione
- Condivisione e protezione



## Il concetto di file

- Collezione di informazioni correlate conservate su memoria di massa. **Unità logica** di memorizzazione.
- I file contengono:
  - dati
  - programmi



## Struttura di un file

- Un file ha una certa struttura che dipende dal **tipo** di dati che contiene
- **File di testo**: sequenza di caratteri, parole, linee, pagine.
- **File sorgente**: sequenza di subroutine e funzioni
- **File oggetto**: sequenza di byte organizzate in blocchi che risultano comprensibili al linker del SO
- **File eseguibile**: serie di sezioni di codice binario che il caricatore porta in memoria ed esegue
- Chi stabilisce la struttura:
  - sistema operativo (e.g., eseguibili)
  - programma (e.g., word processor)



## Attributi di un file

- o **Nome** - unica informazione in una forma leggibile dagli esseri umani.
- o **Identificatore** - tag unico (spesso numerico) che identifica il file all'interno del file system
- o **Tipo** - nei sistemi che supportano differenti tipi di file.
- o **Locazione** - puntatore alla locazione fisica del file nel dispositivo.
- o **Dimensione** - dimensione corrente del file.
- o **Protezione** - determina chi può leggere, scrivere, eseguire.
- o **Tempo, data e identificativo dell'utente** - dati utili per la protezione, la sicurezza ed il monitoraggio d'uso.

Le informazioni sui file sono mantenute nella struttura delle directory, che risiede sul disco.



## Operazioni sui file

- o Creazione
- o Scrittura
- o Lettura
- o Riposizionamento all'interno di un file
- o Cancellazione
- o Troncamento



## Apertura di un file

- o Molte operazioni richiedono la ricerca in una directory di una entry (nome del file)
- o Open e close
- o Tabella dei file aperti. Contiene informazioni relative al file **indipendenti** dai processi
- o Tabella dei file aperti dal processo. Contiene informazioni riguardanti l'uso del file **da parte del processo**.



## Informazioni richieste da un file aperto

- o Diversi dati debbono essere gestiti:
  - **File pointer**: puntatore all'ultima locazione letta/scritta, diversa per ogni processo che ha aperto il file
  - **Contatore di open**: contatore del numero di aperture del file - per permettere la rimozione di dati dalla tabella dei file aperti, quando l'ultimo processo chiude il file
  - **Locazione del file su disco**: informazioni necessarie per l'accesso ai dati, tenute in memoria
  - **Diritti d'accesso**: informazioni sulle modalità d'accesso per ogni processo



## Apertura di un file con locking

- o Fornita da alcuni sistemi operativi e file system
- o Media l'accesso ai file
- o Mandatory o advisory:
  - **Mandatory** (obbligatorio) – l'accesso viene concesso o negato a seconda dello stato del lock
  - **Advisory** (consigliato) – I processi possono controllare lo stato del lock e decidere cosa fare



## Locking di un file. Esempio - Java API

```
import java.io.*;
import java.nio.channels.*;
public class LockingExample {
    public static final boolean EXCLUSIVE = false;
    public static final boolean SHARED = true;
    public static void main(String args[]) throws IOException {
        FileLock sharedLock = null;
        FileLock exclusiveLock = null;
        try {
            RandomAccessFile raf = new RandomAccessFile("file.txt", "rw");
            // get the channel for the file
            FileChannel ch = raf.getChannel();
            // this locks the first half of the file - exclusive
            exclusiveLock = ch.lock(0, raf.length()/2, EXCLUSIVE);
            /** Now modify the data . . . */
            // release the lock
            exclusiveLock.release();
        }
    }
}
```



## Locking di un file. Esempio - Java API

```

// this locks the second half of the file - shared
sharedLock = ch.lock(raf.length()/2+1, raf.length(),
    SHARED);
/** Now read the data . . . */
// release the lock
exclusiveLock.release();
} catch (java.io.IOException ioe) {
    System.err.println(ioe);
}finally {
    if (exclusiveLock != null)
        exclusiveLock.release();
    if (sharedLock != null)
        sharedLock.release();
}
}
}
```



## Tipi di file - Nome, estensione

Supportare o non supportare i tipi?

Nei sistemi operativi che supportano i tipi, le estensioni identificano la struttura del file

In generale le estensioni sono un semplice suggerimento per le applicazioni

file type	usual extension	function
executable	exe, com, bin or none	ready-to-run machine-language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, java, pas, asm, a	source code in various languages
batch	bat, sh	commands to the command interpreter
text	txt, doc	textual data, documents
word processor	wp, tex, rtf, doc	various word-processor formats
library	lib, a, so, dll	libraries of routines for programmers
print or view	ps, pdf, jpg	ASCII or binary file in a format for printing or viewing
archive	arc, zip, tar	related files grouped into one file, sometimes compressed, for archiving or storage
multimedia	mpeg, mov, rm, mp3, avi	binary file containing audio or A/V information

## Metodi di accesso

- o Accesso sequenziale

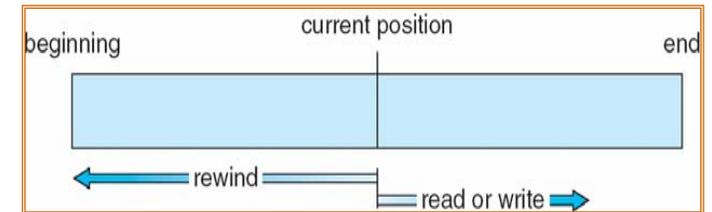
*lettura successivo el  
scrittura successivo el  
riposizionamento (reset)*

- o Accesso diretto

*lettura blocco n  
scrittura blocco n  
posizionamento al blocco n  
lettura  
scrittura*

*n = numero (relativo) di blocco*

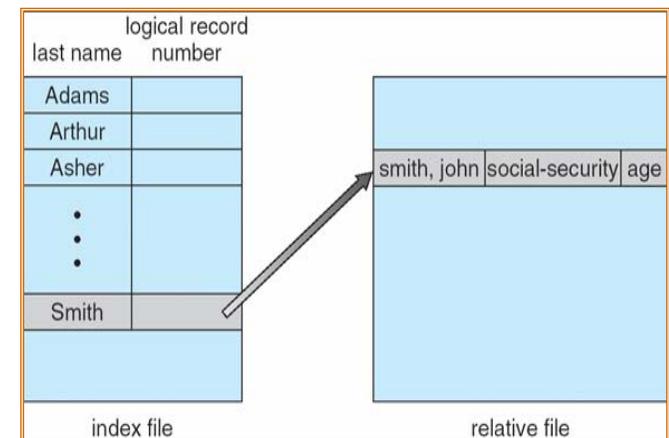
## File ad accesso sequenziale



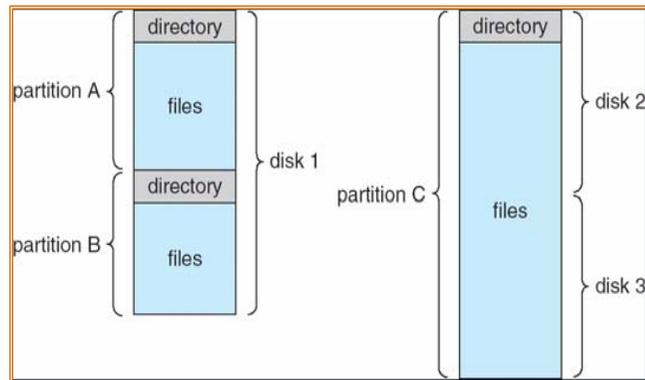
## Simulazione dell'accesso sequenziale in un file ad accesso diretto

sequential access	implementation for direct access
<i>reset</i>	<i>cp = 0;</i>
<i>read next</i>	<i>read cp; cp = cp + 1;</i>
<i>write next</i>	<i>write cp; cp = cp + 1;</i>

## Esempio di file indice e di file relativo



## Tipica organizzazione di un file system



## Attributi di una directory

- o Nome.
- o Indirizzo.
- o Lunghezza corrente.
- o Lunghezza massima.
- o Date di ultimo accesso.
- o Date di ultimo aggiornamento.
- o ID del proprietario.
- o Informazioni sulla protezione.

## Operazioni eseguibili su una directory

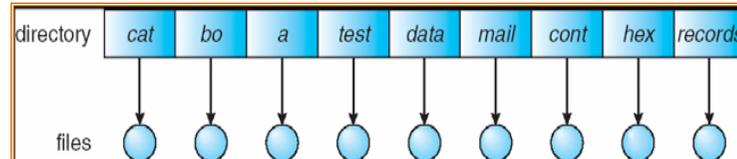
- o Ricerca di un file.
- o Creazione di un file.
- o Cancellazione di un file.
- o Elencare il contenuto di una directory.
- o Rinomina di un file.
- o Attraversamento del file system.

## Organizzare (logicamente) una directory per ottenere:

- o **Efficienza** - localizzazione rapida dei file.
- o **Assegnazione di nomi conveniente per gli utenti.**
  - Due utenti possono scegliere lo stesso nome per file differenti.
  - Lo stesso file può avere parecchi nomi diversi.
- o **Raggruppamento** - raggruppamento logico di file in base alle proprietà (ad esempio tutti i programmi Java, tutti i giochi, ...).

## Directory a singolo livello

- Una sola directory per tutti gli utenti.

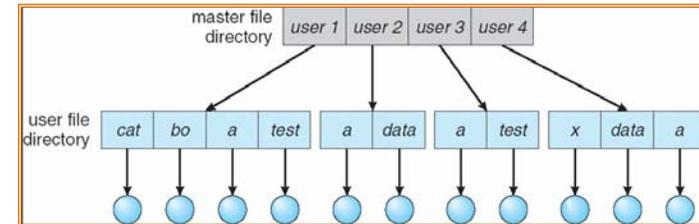


Assegnazione dei nomi?

Raggruppamento?

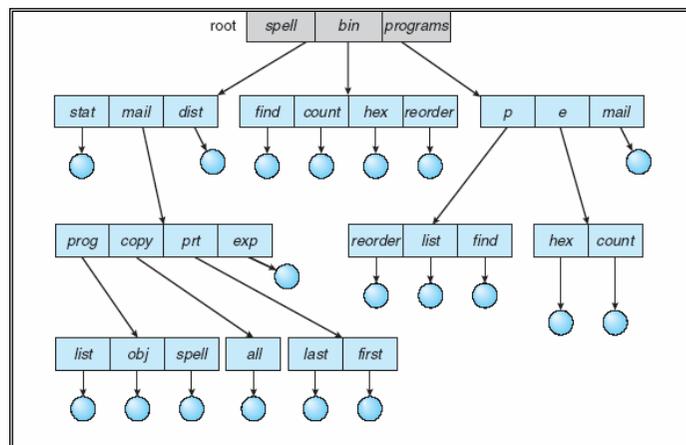
## Directory a due livelli

- Directory separata per ogni utente.



- Nome del percorso (e.g., C:\utente\file /u/prova/file)
- Utenti diversi possono usare lo stesso nome file
- Ricerca efficiente.
- Nessuna capacità di raggruppamento.

## Struttura ad albero



## Struttura ad albero

- Ricerca efficiente.
- Capacità di raggruppamento.
- Directory corrente (directory di lavoro)
- Path relativa ed assoluta
- Path di ricerca
- Cambio della directory corrente
  - `cd /spell/mail/prog`

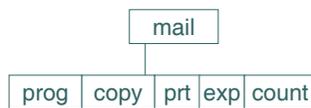
## Struttura ad albero

- o La creazione di un nuovo file viene effettuata nella directory corrente.
- o Cancellazione di un file  
`rm <nome-file>`
- o La creazione di una nuova sottodirectory viene effettuata nella directory corrente

`mkdir <dir-name>`

Eempio: se nella directory corrente `/mail`

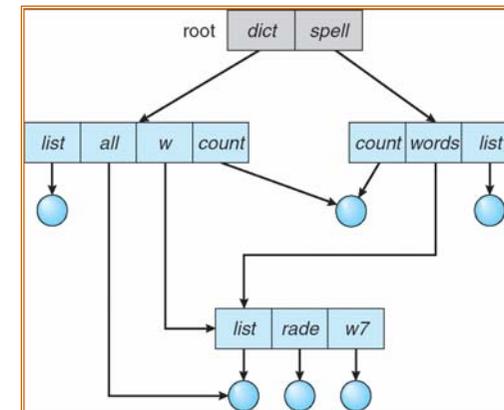
`mkdir count`



Cancellare "mail" ⇒ cancellare l'intero sottoalbero avente "mail" come radice?

## Struttura a grafo aciclico

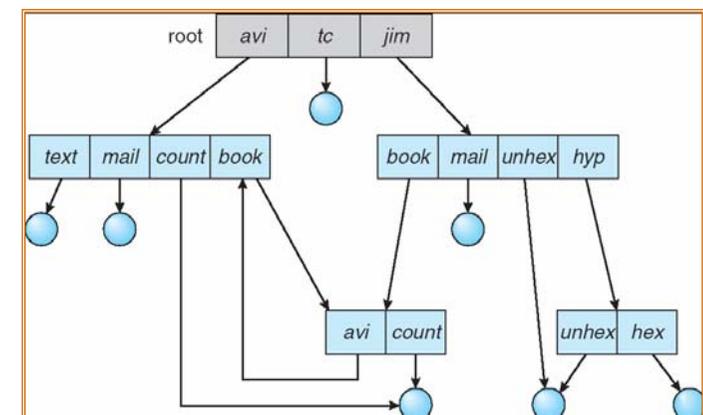
- o Ha sottodirectory e file condivisi.



## Struttura a grafo aciclico

- o Due nomi diversi (aliasing)
- o Nuovi tipi di elementi in una directory
  - **Link** – altro puntatore (pointer) ad un file esistente
  - **Risolvere il link** – seguire il puntatore per localizzare il file
- o Se `words` cancella `list` ⇒ riferimenti pendenti
- o Soluzioni:
  - Puntatori all'indietro
  - Contatore di riferimenti al file

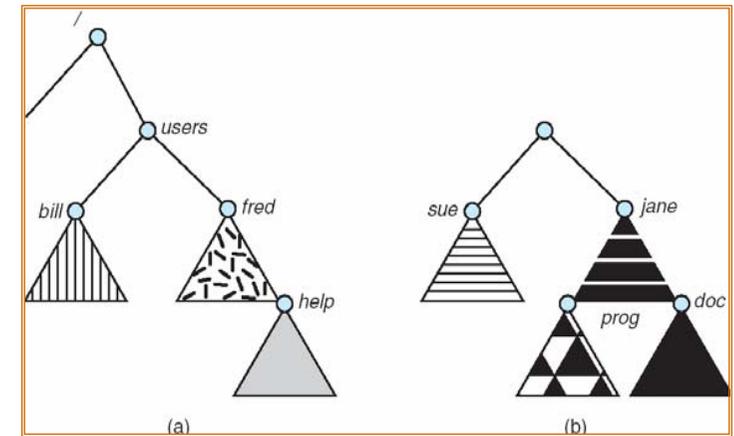
## Grafo generale



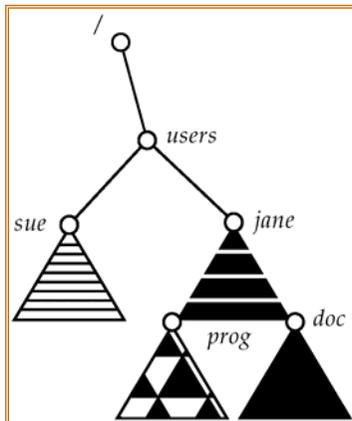
## Montaggio del file system

- Un file system deve essere montato prima di essere usato.
- Un file system non montato (Fig. 11-11(b)) è montato su un punto di montaggio.

## (a) File system esistente (b) Partizione non montata



## Punto di montaggio



## File Sharing

- La condivisione di file su sistemi multiutente è desiderabile.
- Si richiede uno schema di condivisione.
- Nei sistemi distribuiti i file sono condivisi attraverso una rete di comunicazione.



## Utenti Multipli

- **User ID:** identificano gli utenti, permettono di associare permessi e protezioni ai singoli utenti
- **Group ID:** permettono agli utenti di essere in gruppi, permettono di definire diritti di accesso di gruppo



## File System Remoti

- Metodi per usare la rete per fornire accesso a file system su sistemi remoti
  - Manualmente, tramite programmi quali FTP
  - Automaticamente, usando **file system distribuiti**
  - Semi automaticamente tramite il **world wide web**



## File system distribuiti

- Il **modello client-server** permette ai client di montare file system remoti da server
- Un server può servire più client
- Client e identificazione user-on-client è insicura o complicata
- **NFS** è un protocollo standard UNIX di tipo client-server per il file sharing
- **CIFS** è il protocollo standard per Windows
- Chiamate standard al sistema operativo sono tradotte in chiamate remote
- Sistemi di informazioni distribuiti come DNS, NIS, Active Directory ed LDAP implementano un accesso unificato alle informazioni necessarie al calcolo remoto



## Guasti

- File system remoti aggiungono nuove modalità di guasto, a causa di guasti di rete, guasti del un server
- Il recupero da un guasto può richiedere informazioni circa lo **stato** di ciascuna richiesta remota
- Protocolli senza stato quali NFS includono informazioni complete in ogni richiesta, permettendo un recupero veloce ma offrendo minor sicurezza

## Semantica della consistenza

- o Specifica come utenti multipli accedono ad un file condiviso simultaneamente
  - Similitudine con gli algoritmi di sincronizzazione tra processi
    - Meno complesso per la presenza di disk I/O e latenza di rete (per file system remoti)
  - Il file system Unix (UFS) implementa:
    - Scritture su un file condiviso visibili immediatamente agli altri utenti che hanno aperto il file
    - Condivisione del file pointer per permettere a più utenti di leggere e scrivere concorrentemente
  - Andrew File System (AFS) implementa una semantica della sessione
    - Le scritture sono visibili solo in sessioni che cominciano quando il file è stato chiuso

## Protezione

- o Il proprietario/creatore del file dovrebbe essere in grado di controllare:
  - che cosa può essere fatto
  - da chi
- o Tipi di accesso:
  - Lettura
  - Scrittura
  - Esecuzione
  - Accodamento
  - Cancellazione
  - Lista

## Lista degli accessi e gruppi

- o Modo di accesso: lettura, scrittura, esecuzione.
- o Tre classi di utenti:
 

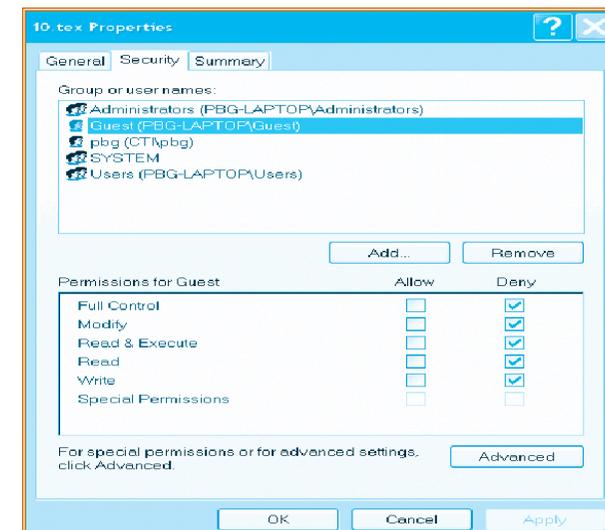
a) accesso proprietario	7	⇒	RWX
b) accesso gruppo	6	⇒	RWX
c) accesso pubblico	1	⇒	001
- o Chiedere al gestore di creare un gruppo (nome unico), per esempio *G*, ed aggiungere alcuni utenti al gruppo.
- o Per un file particolare (ad esempio *game*) o una sottodirectory, definire un appropriato accesso.



Assegnare un gruppo al file

chgrp G game

## Gestione della lista di controllo degli accessi in Windows XP





## Una Directory UNIX

-rw-rw-r--	1	pbg	staff	31200	Sep 3 08:30	intro.ps
drwx-----	5	pbg	staff	512	Jul 8 09:33	private/
drwxrwxr-x	2	pbg	staff	512	Jul 8 09:35	doc/
drwxrwx---	2	pbg	student	512	Aug 3 14:13	student-proj/
-rw-r--r--	1	pbg	staff	9423	Feb 24 2003	program.c
-rwxr-xr-x	1	pbg	staff	20471	Feb 24 2003	program
drwx--x--x	4	pbg	faculty	512	Jul 31 10:31	lib/
drwx-----	3	pbg	staff	1024	Aug 29 06:52	mail/
drwxrwxrwx	3	pbg	staff	512	Jul 8 09:35	test/