

File System

Capitolo 10 -- Silberschatz

Interfaccia del File System

- Per gli utenti di un computer, il **File System** è spesso l'aspetto più visibile del SO di quel computer.
- Il File System fornisce infatti i meccanismi per la memorizzazione e l'accesso ai dati e ai programmi del SO e degli utenti del sistema di calcolo.
- Un File System consiste di due parti:
 - Un insieme di file
 - Una struttura di directory, che permette di organizzare tutti i file del sistema

File

Il concetto di file

- I calcolatori mantengono i dati su supporti diversi: dischi, nastri ...
 - Il sistema operativo agevola l'uso offrendo una visione logica uniforme alla memorizzazione delle informazioni
 - **File**: collezione di informazioni correlate conservate su memoria non volatile a cui è stato assegnato un nome. Unità logica di memorizzazione.
 - I file contengono:
 - dati (numerici, caratteri, binari)
 - programmi (sorgenti, linkabili, eseguibili)

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

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

Tipi di file - Nome.estensione

Mac OS X conserva, all'atto della creazione di un file anche il nome del programma che ha creato il file (questo gli dà modo di poter riconoscere il tipo del file)

Unix memorizza un codice (numero magico) all'inizio di alcuni tipi di file in modo da indicarne in modo generico il tipo (eseguibile, script shell, PostScript)

Attributi di un file

- A ciascun file sono associati degli **attributi**
- Tra le altre cose, gli attributi facilitano l'uso e le possibili operazioni che si possono fare su un file
 - **Nome** – unica informazione in una forma leggibile dagli esseri umani.
 - **Identificatore** – tag unico (spesso numerico) che identifica il file all'interno del file system
 - **Tipo** – nei sistemi che supportano differenti tipi di file.

Attributi di un file

- **Locazione**
 - Posizione fisica: puntatore alla locazione fisica del file nel dispositivo.
 - Posizione logica: il pathname del file
 - Questa informazione NON è memorizzata esplicitamente da nessuna parte eccetto che in casi particolari
- **Dimensione** – dimensione corrente del file.
- **Protezione** – determina chi può leggere, scrivere, eseguire.
- **Ora, 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

- Un file può essere visto come un tipo di dato astratto definito solo dalle operazioni che si possono compiere su di esso, rese disponibili dal sistema operativo
- **Creazione:** richiede al SO di trovare spazio per il file, e poi di creare un accesso al file attraverso la directory che “contiene” il file, secondo le modalità di accesso stabilite per quel file
- **Scrittura / Lettura:** il SO mette a disposizione una opportuna system call per specificare il nome del file su cui operare. Il SO deve
 - Gestire il puntatore di scrittura/lettura al punto del file in cui si vuole scrivere/leggere
 - Occuparsi di trovare spazio sufficiente per ospitare l'eventuale espansione del file, in caso di scrittura

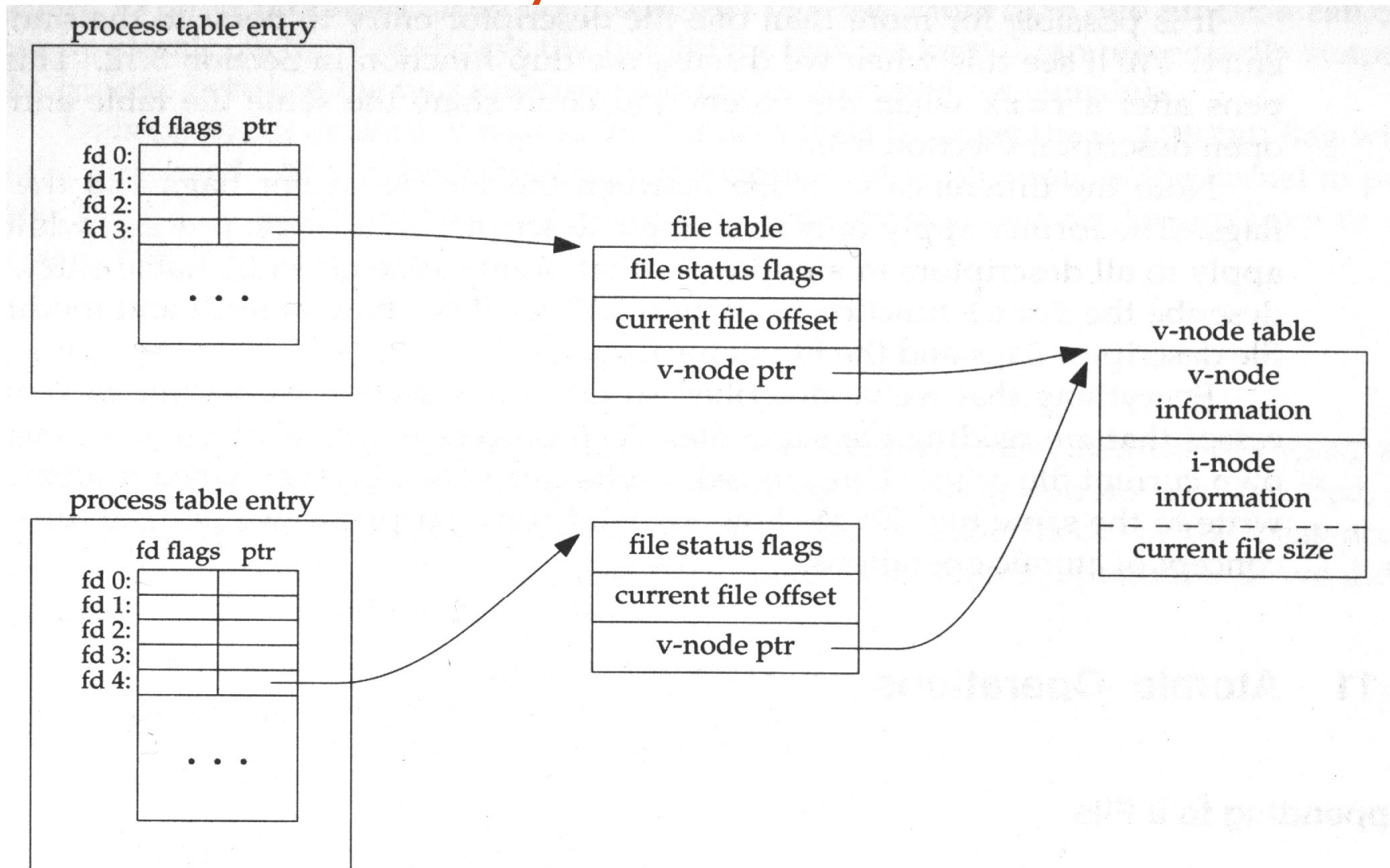
Operazioni sui file

- **Riposizionamento all'interno di un file:** per leggere o scrivere a partire dal punto specificato
- **Cancellazione:** recupera lo spazio occupato dal file sul supporto di memoria secondaria e lo spazio occupato nella directory che lo conserva
- **Troncamento:** cancella i dati memorizzati e recupera lo spazio occupato, ma mantiene tutti gli altri attributi del file
- Oltre a tali operazioni di base, si può:
 - **rinominare, copiare, spostare**

Apertura di un file

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

2 processi su uno stesso file nel FileSystem di Unix



Informazioni richieste da un file aperto

- Diversi dati debbono essere gestiti:
 - **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
 - **File pointer**: puntatore all'ultima locazione letta/scritta, diversa per ogni processo che ha aperto il file

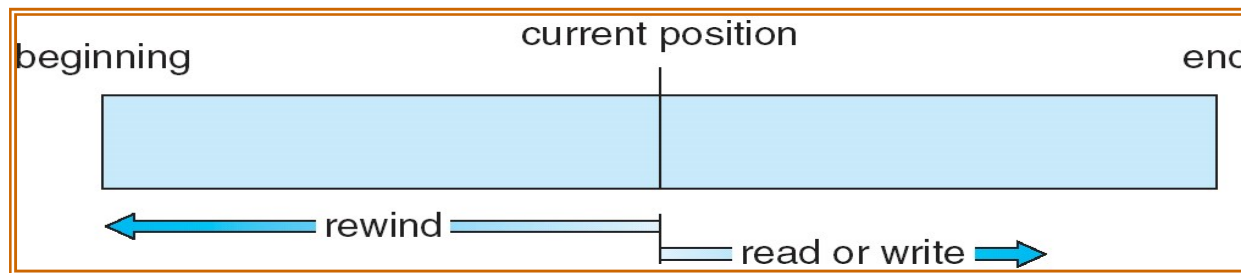
Apertura di un file con locking

- Fornita da alcuni sistemi operativi e file system
- Media l'accesso ai file
- **Condiviso** (più processi concorrenti possono appropriarsi del file – lock di lettura) o **Esclusivo** (un solo processo alla volta può appropriarsi del file – lock di scrittura)
- Mandatory o advisory:
 - **Mandatory** (obbligatorio) – l'accesso viene concesso o negato a seconda dello stato del lock (Windows)
 - **Advisory** (consigliato) – I processi possono controllare lo stato del lock e decidere cosa fare (Unix)

Metodi di accesso

- **Accesso sequenziale**

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



L'accesso sequenziale è basato su un modello di file che si rifà al nastro

Metodi di accesso

□ **Accesso diretto**

Nell'accesso diretto si considera il file formato da elementi logici (blocco) di lunghezza fissa.

Il file è considerato come una sequenza di blocchi che si possono leggere, scrivere in modo arbitrario (si fonda su un modello di file che si rifà al disco).

Utile per accedere a grandi quantità di dati correlati (database)

- lettura blocco **n**
- scrittura blocco **n**
- posizionamento al blocco **n**
 - ▶ lettura
 - ▶ scrittura

n = numero di blocco (relativo all'inizio del file)

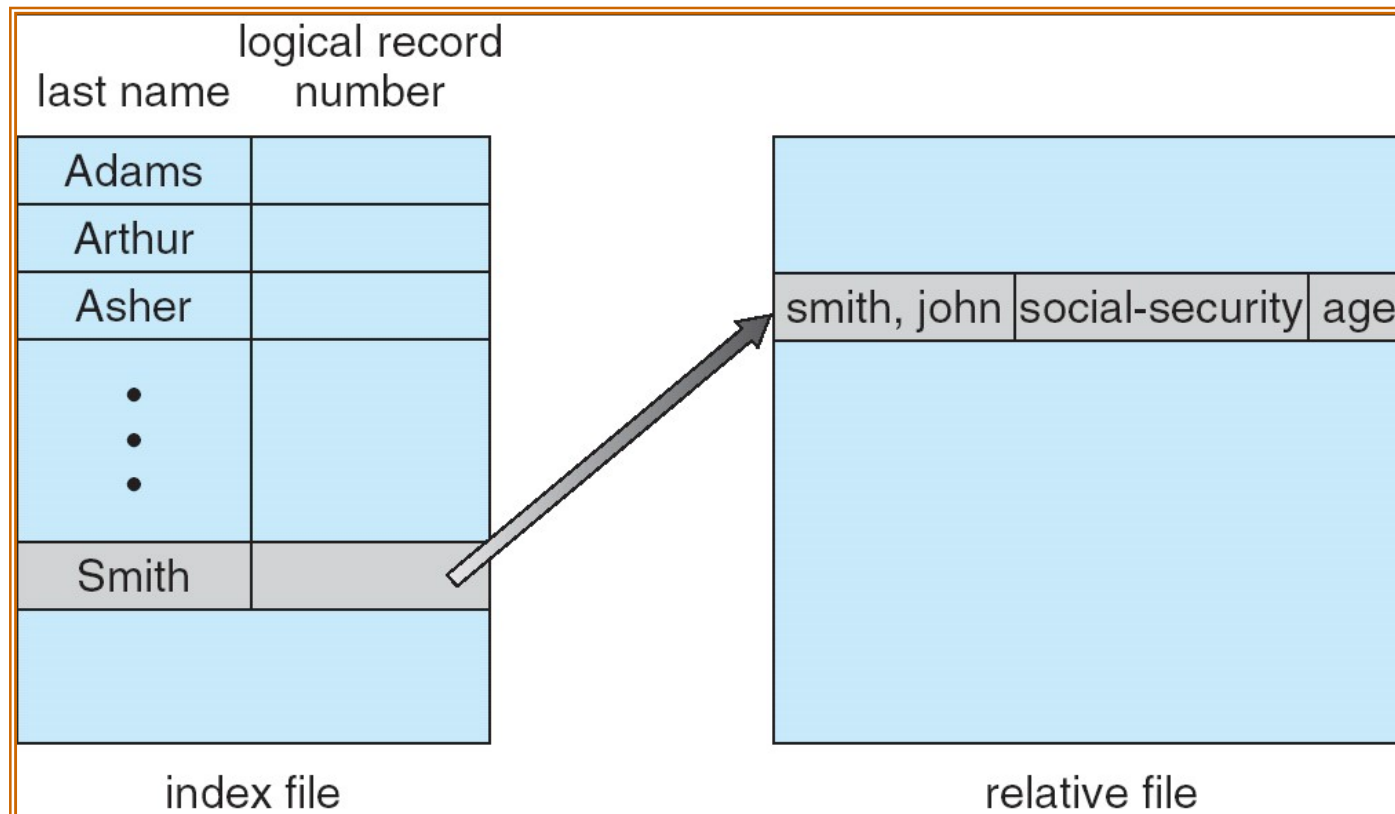
Simulazione dell'accesso sequenziale disponendo di accesso diretto

- Avendo a disposizione un accesso diretto si utilizza una variabile *cp*, che mi definisce la posizione corrente, per simulare l'accesso sequenziale

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

File indice e file relativo

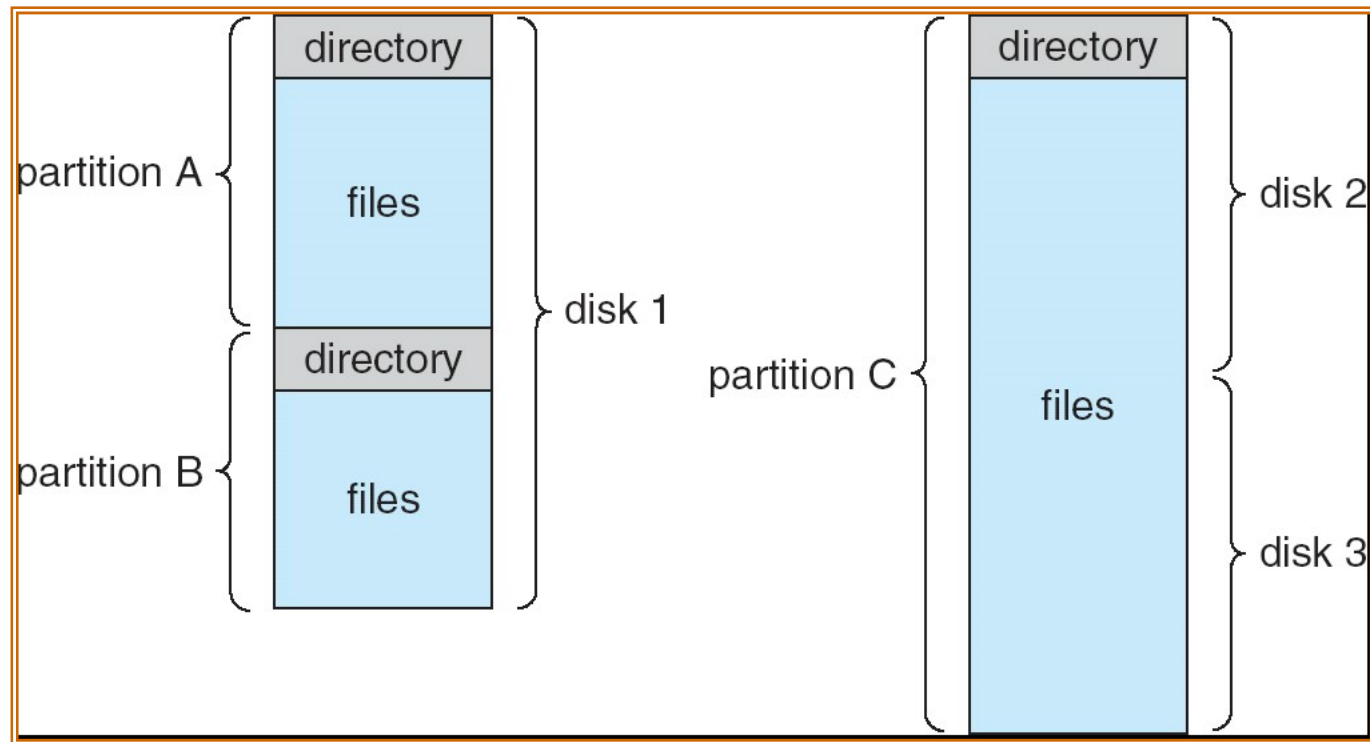
- Il file indice contiene i puntatori ai vari blocchi
 - Per trovare un elemento del file occorre prima cercare nell'indice e quindi usare il puntatore per accedere direttamente al file e trovare l'elemento desiderato



Organizzazione di un File System

Tipica organizzazione di un file system

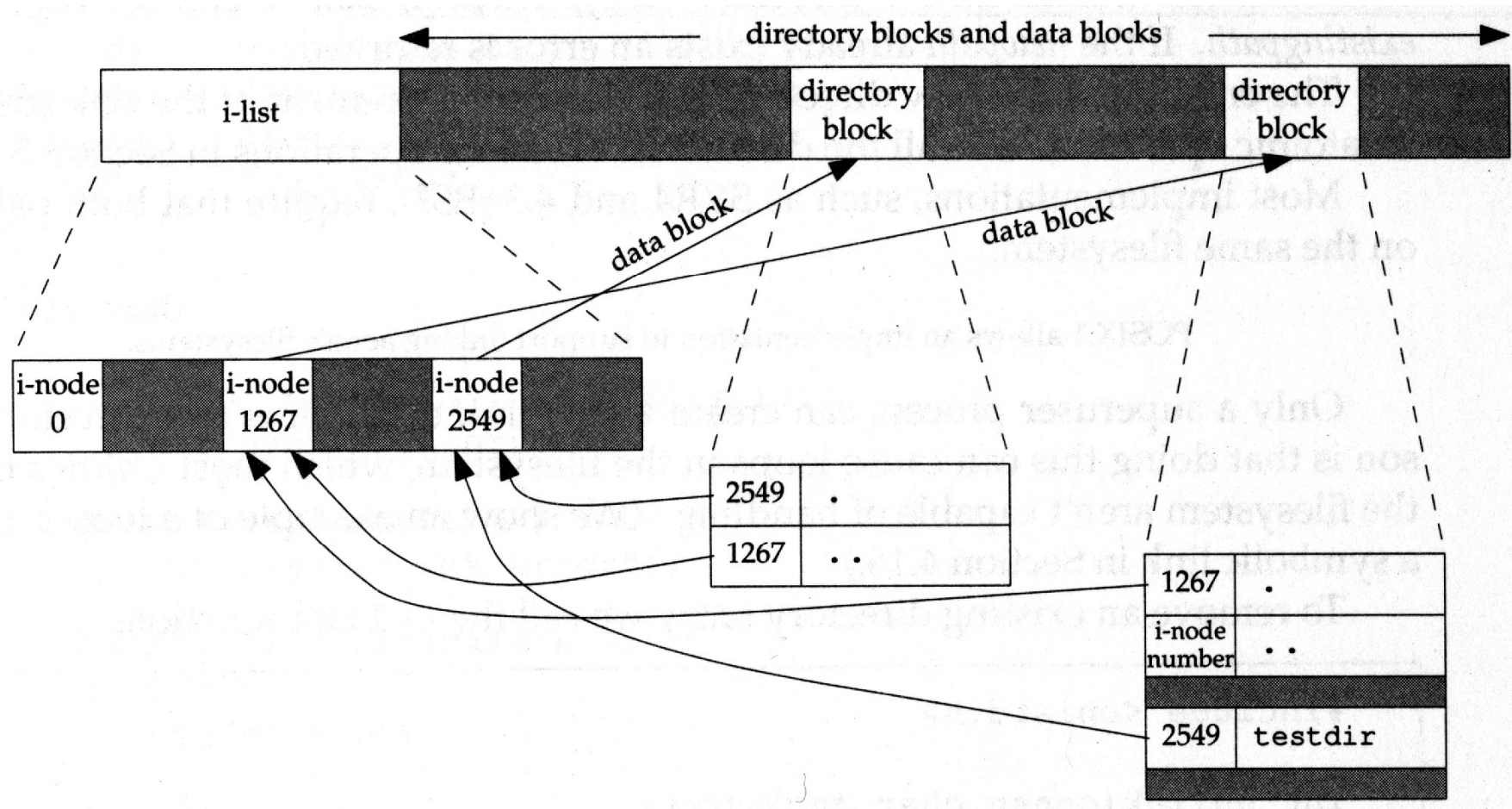
- Ogni file system deve contenere le informazioni sui file presenti nel sistema → **directory**



Attributi di una directory

- Nome.
- Indirizzo.
- Lunghezza corrente.
- Data ultimo accesso.
- Data ultimo aggiornamento.
- ID del proprietario.
- Informazioni sulla protezione.

Esempio dell'organizzazione del filesystem di Unix



Operazioni eseguibili su una directory

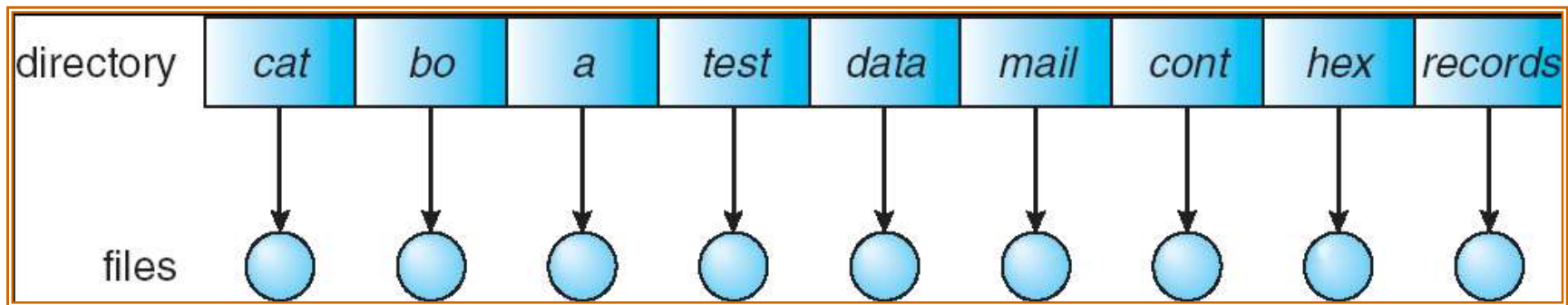
- **Ricerca** di un file.
- **Creazione** di un file.
- **Cancellazione** di un file.
- **Elencare** il contenuto di una directory.
- **Rinomina** di un file.
- **Attraversamento** del file system.

Organizzare (logicamente) i file in directory per ottenere:

- **Efficienza** – localizzazione rapida dei file.
- **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.
- **Raggruppamento** – raggruppamento logico di file in base alle proprietà (ad esempio tutti i programmi Java, tutti i giochi, ...).

Directory a singolo livello

- Tutti i dati sono contenuti in una sola directory
- Una sola directory per tutti gli utenti

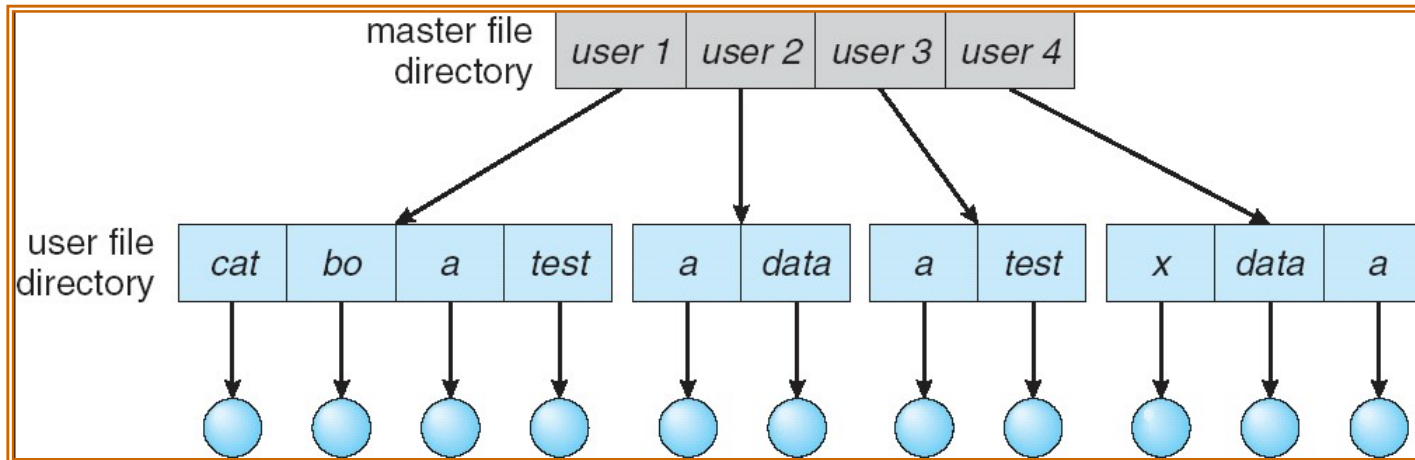


Problemi:

- Assegnazione dei nomi
- Raggruppamento

Directory a due livelli

- Directory separata per ogni utente.

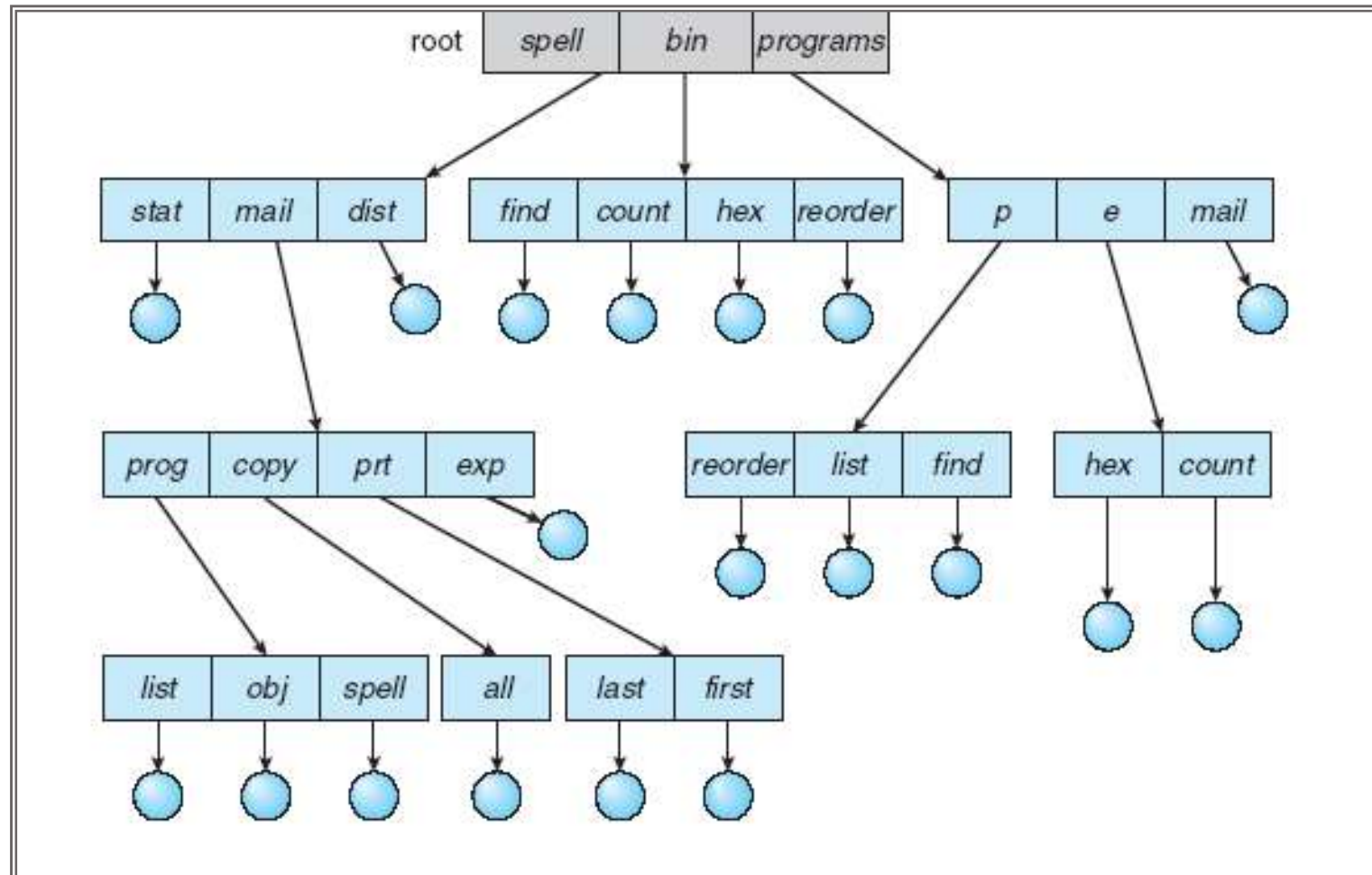


- Utenti diversi possono usare lo stesso nome file
- Ricerca più efficiente rispetto alla precedente.
- Nessuna capacità di raggruppamento.

Windows mantiene una struttura delle directory a due livelli estesa, con lettere per i dispositivi ed i volumi

e.g.: **C:\utente\file**

Struttura ad albero



Struttura ad albero

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

Struttura ad albero

- Cancellazione di un file

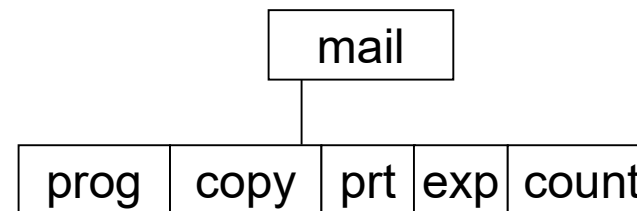
`rm <file-name>`

- La creazione di una nuova sottodirectory viene effettuata nella directory corrente

`mkdir <dir-name>`

Esempio: se nella directory corrente **/mail**

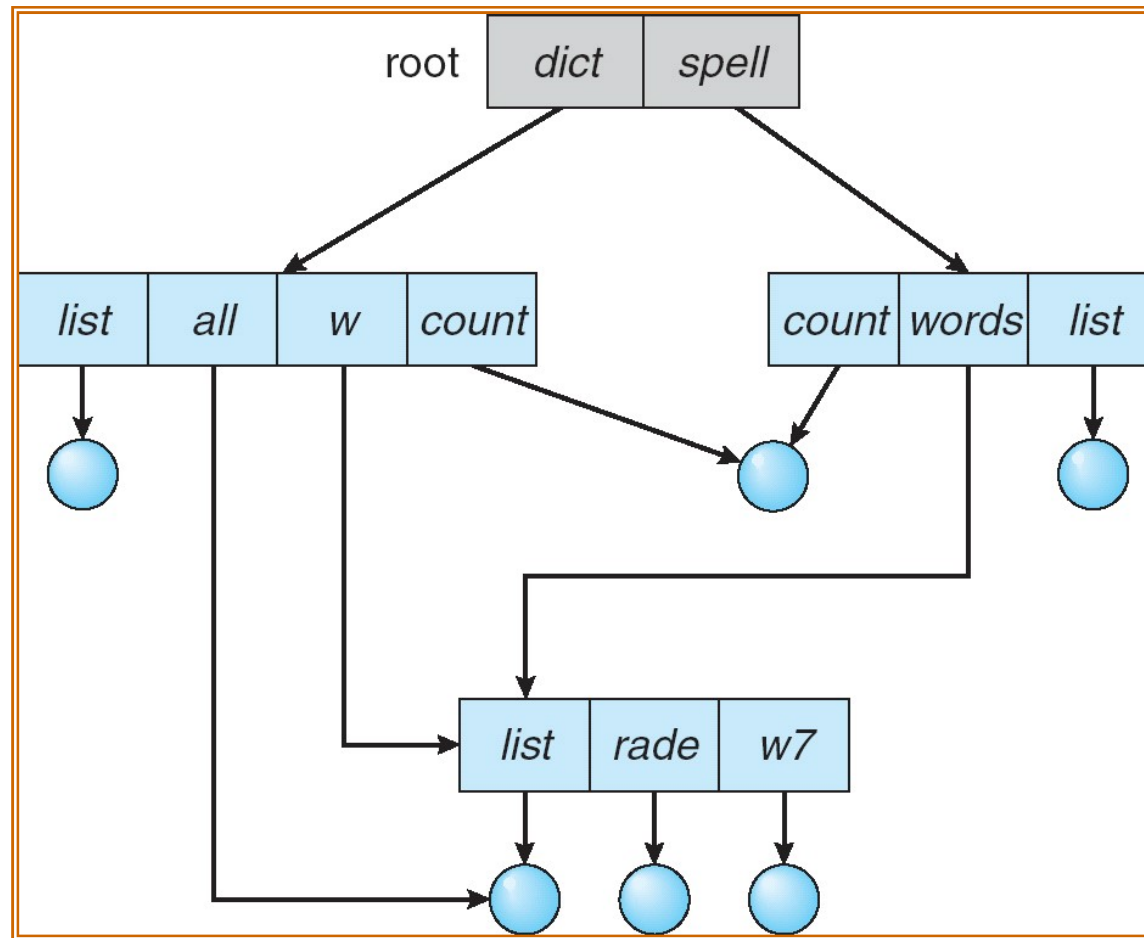
`mkdir count`



Cancellare “mail” ⇒ cancellare l’intero sottoalbero avente “mail” come radice?

Struttura a grafo aciclico

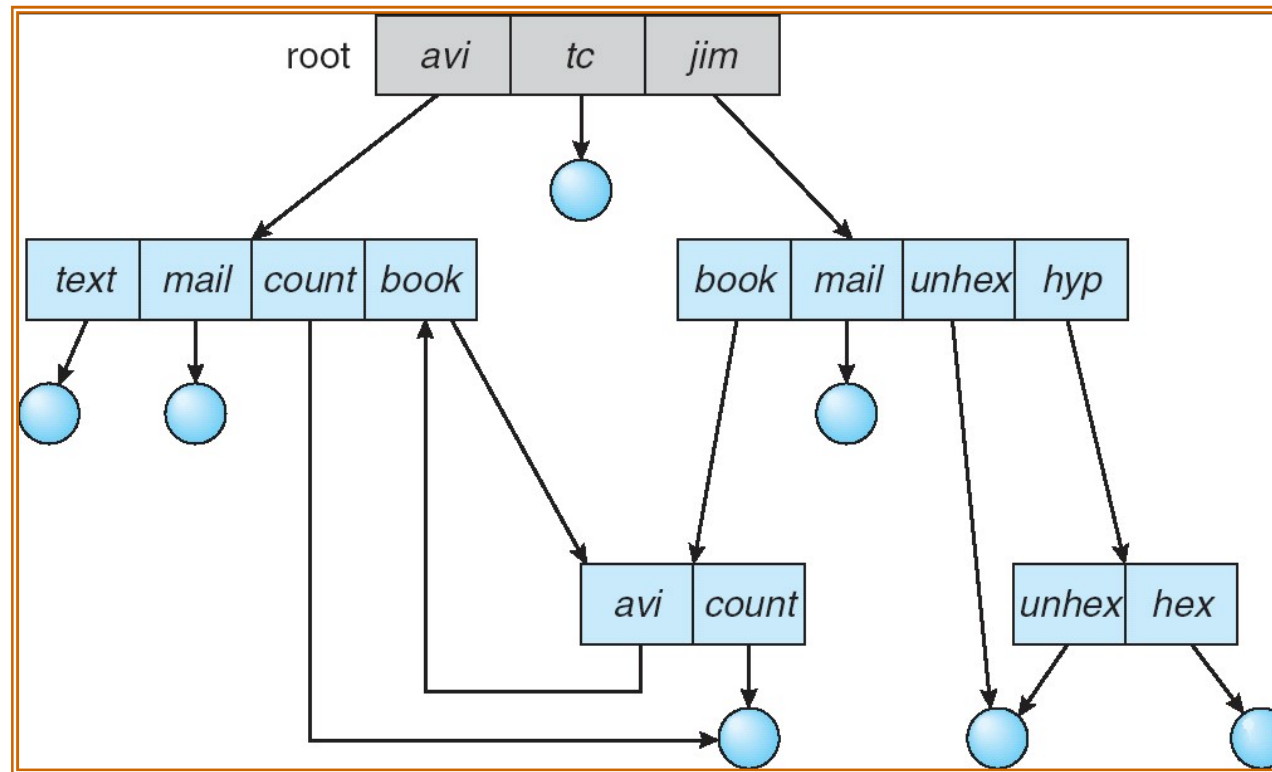
- Ha sottodirectory e file condivisi.



Struttura a grafo aciclico

- Due nomi diversi (aliasing)
- Nuovi tipi di elementi in una directory
 - **Link simbolici** – altro puntatore (percorso) ad un file esistente
 - **Risolvere il link** – seguire il puntatore per localizzare il file
- Se **words** cancella **list** \Rightarrow riferimenti pendenti da parte di **w**
- Soluzioni:
 - Lista di tutti i riferimenti al file
 - Contatore di riferimenti al file (Unix usa questo metodo per gli **hard link**)

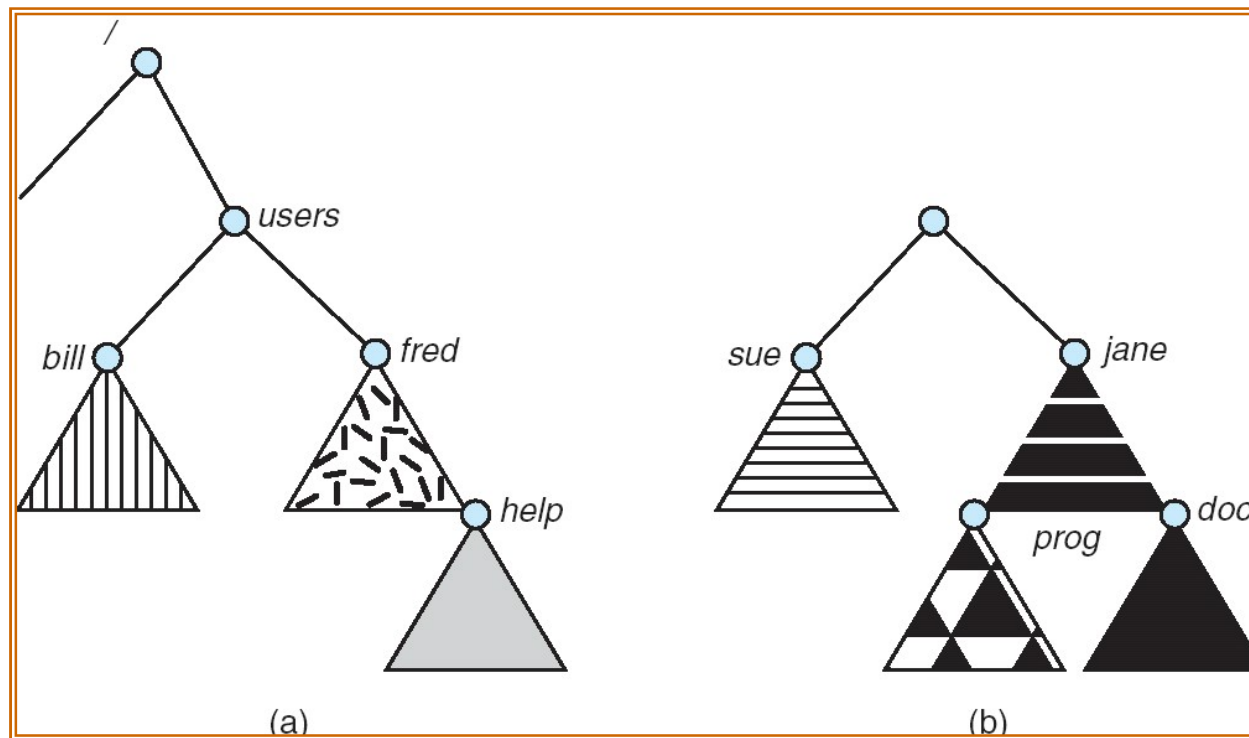
Grafo generale



- Questa situazione è pericolosa: se un programma visita ricorsivamente una directory e le sue sottodirectory, potrebbe non accorgersi di essere entrato in loop

Montaggio di un file system

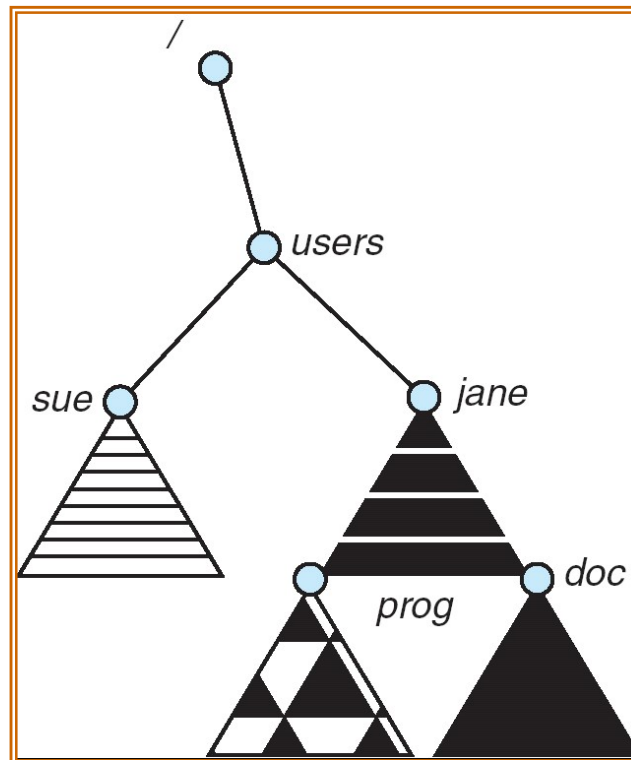
- Un file system deve essere montato prima di essere usato.
- Un file system **non montato** è montato su un punto di montaggio.



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

Punto di montaggio

- Un punto di montaggio è in genere una directory vuota ma non è detto;
- Nel caso in cui la directory di montaggio non è vuota si nascondono i file preesistenti il montaggio fino al momento in cui si smonta



Protezione

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

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

Lista degli accessi e gruppi

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

			RWX
a) accesso proprietario	7	\Rightarrow	1 1 1
b) accesso gruppo	6	\Rightarrow	1 1 0
c) accesso pubblico	1	\Rightarrow	0 0 1

- Chiedere al gestore di creare un gruppo (nome unico), per esempio G, ed aggiungere alcuni utenti al gruppo.
- Per un file particolare (ad esempio game) o un sottodirectory, definire l'accesso in modo appropriato.

proprietario gruppo pubblico
| | |
chmod 761 game

Assegnare un gruppo ad un file
chgrp G game

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/

Gestione della lista di controllo degli accessi in Windows XP

