

FILE SYSTEM : INTERFACCIA

Interfaccia del File System

- Concetto di File
- Metodi di Accesso
- Struttura delle Directory
- Montaggio del File System
- Condivisione di File
- Protezione

Concetto di File

- File : uno spazio di memoria logicamente contiguo (e non volatile).
- Tipi di file:
 - Dati
 - ❖ numerici
 - ❖ A caratteri
 - ❖ binari
 - Programmi

Struttura dei File

- Nessuna - sequenza di word o di byte
- Struttura semplice a record
 - Linee
 - Lunghezza fissata
 - Lunghezza variabile
- Struttura complessa
 - documenti formattati
 - Dati rilocabili (eseguibile)

- Questi due tipi possono essere simulati inserendo caratteri di control.
- Chi decide:
 - Il sistema operativo
 - Programma.

Attributi dei File

- **Nome** – nome simbolico (*es: lettera.doc, sort.java*).
- **Tipo** – quando ci sono differenti tipi di file.
- **Locazione** – indirizzo del file sul dispositivo fisico.
- **Dimensione** – numero di byte, word o blocchi.
- **Protezione** – bit di lettura, scrittura, esecuzione.
- **Ora, data e id utente** – creazione, modifica, accesso; usate per protezione, sicurezza, e monitoring.
- L'informazione sui file è tenuta nella struttura a directory memorizzata sul disco.

File: Operazioni

- Creazione
- Scrittura
- lettura
- Ri-posizionamento nel file
- Cancellazione
- Troncamento
- Apertura

Open (F_i) – cerca sul disco nelle directory una entry F_i , e sposta il contenuto in memoria centrale.

- Chiusura

Close (F_i) – sposta il contenuto di F_i dalla memoria centrale al disco.

Tipi di File – Nome ed Estensione

file type	usual extension	function
executable	exe, com, bin or none	read 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, rrf, doc	various word-processor formats
library	lib, a, so, dll, mpeg, mov, rm	libraries of routines for programmers
print or view	arc, zip, tar	ASCII or binary file in a format for printing or viewing
archive	arc, zip, tar	related files grouped into one file, sometimes com- pressed, for archiving or storage
multimedia	mpeg, mov, rm	binary file containing audio or A/V information

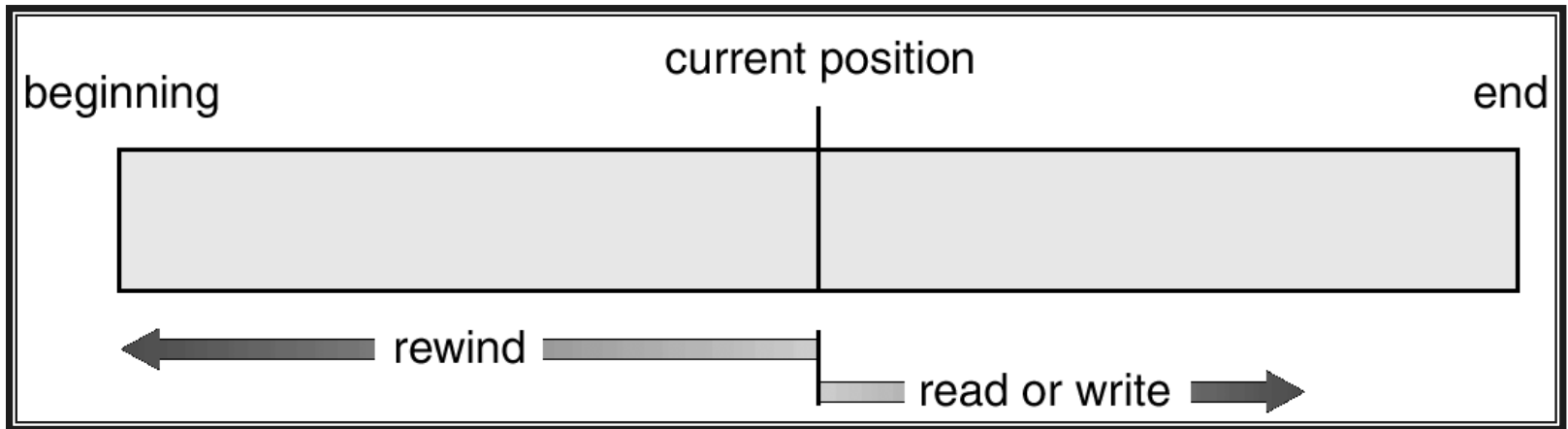
Metodi di accesso

- **Access Sequenziale** : *un record dopo l'altro*
 - *read next*
 - *write next*
 - *reset*

- **Accesso Diretto** : *senza un ordine predefinito*
 - *read n*
 - *write n*
 - *position to n*
 - ❖ *read next*
 - ❖ *write next*

n = numero di blocco relativo

File ad accesso sequenziale

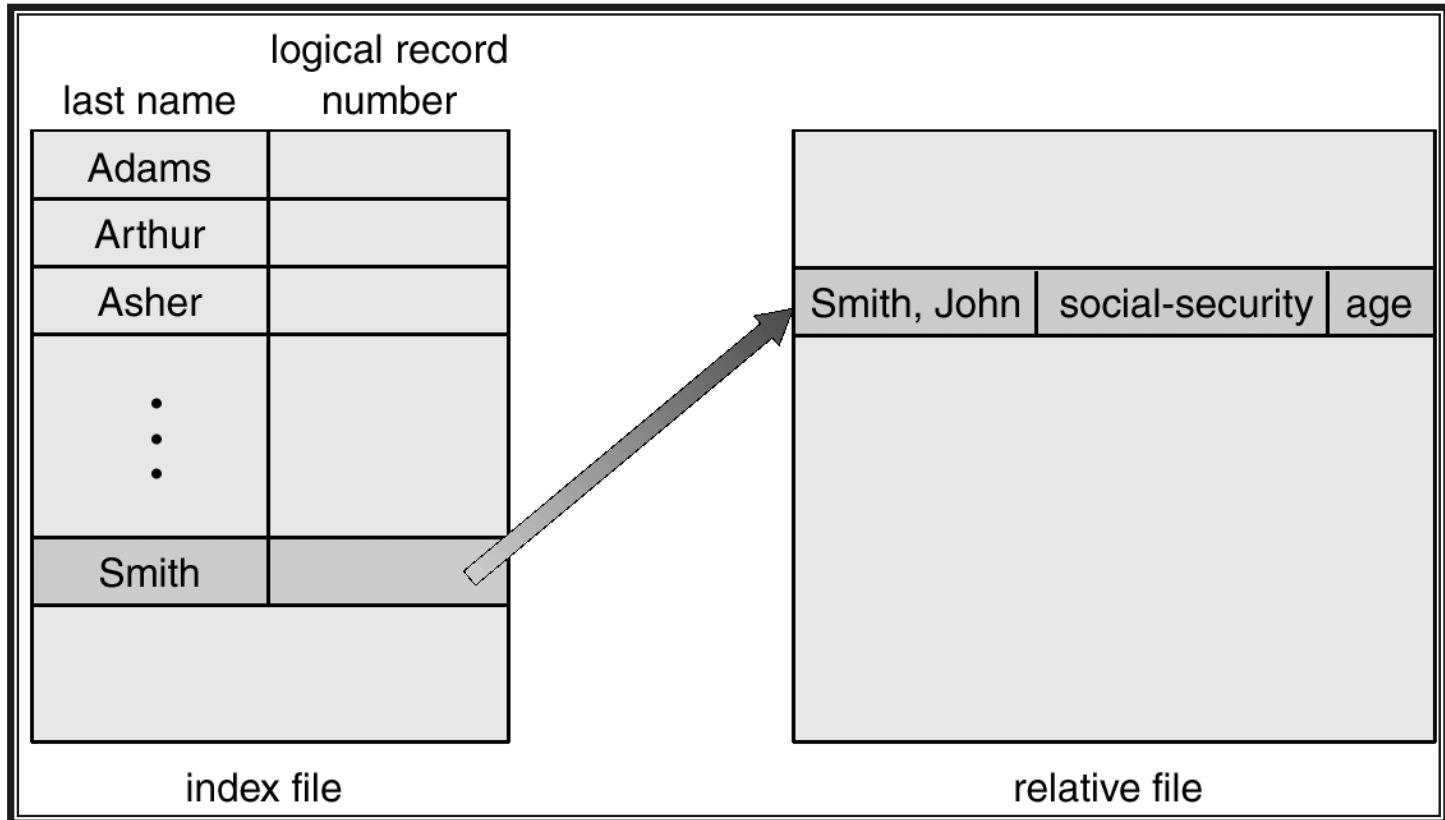


Simulazione dell'accesso sequenziale su un file ad accesso diretto

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

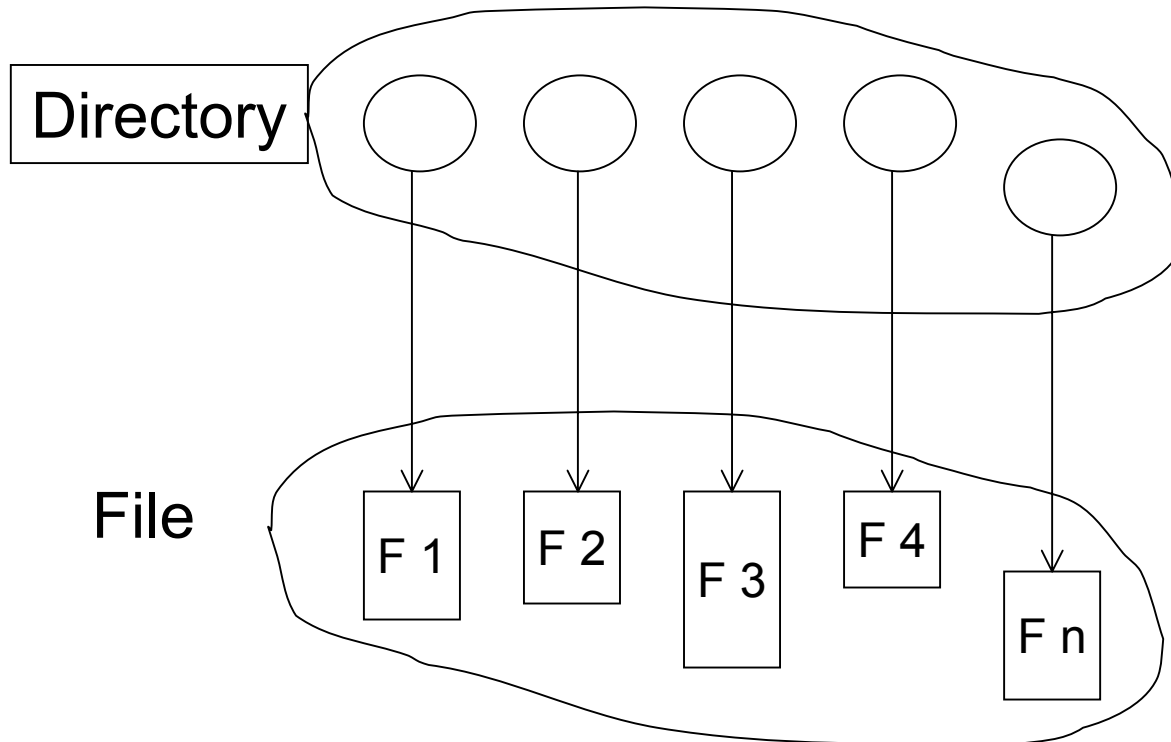
cp : posizione corrente nel file

Esempio: Indice e File Relativi



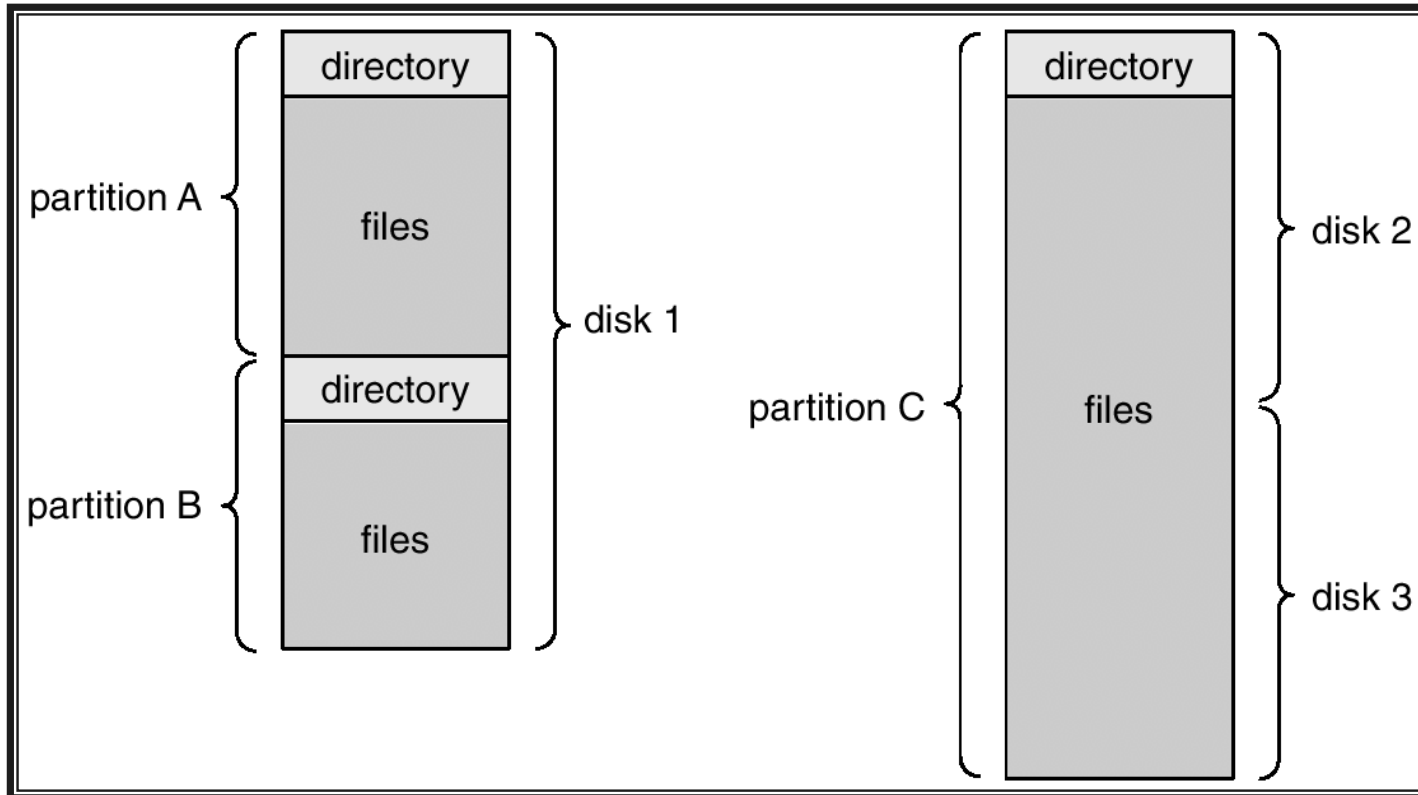
Struttura di Directory

Una collezione di nodi contenenti informazioni su tutti i file.



Sia la struttura di directory sia i file sono su disco.
Copie di backup di essi sono tenute su nastro.

Tipica Organizzazione del File System



Informazione nella Directory del dispositivo

- Ciascuna partizione contiene informazioni sui file che contiene usando la ***directory del dispositivo***:
- Nome
- Tipo
- Indirizzo
- Lunghezza corrente
- Lunghezza massima
- Data ultimo accesso
- Data ultimo aggiornamento (per il dump)
- ID del proprietario (chi paga)
- Informazioni di protezione.

Operazioni sulla Directory

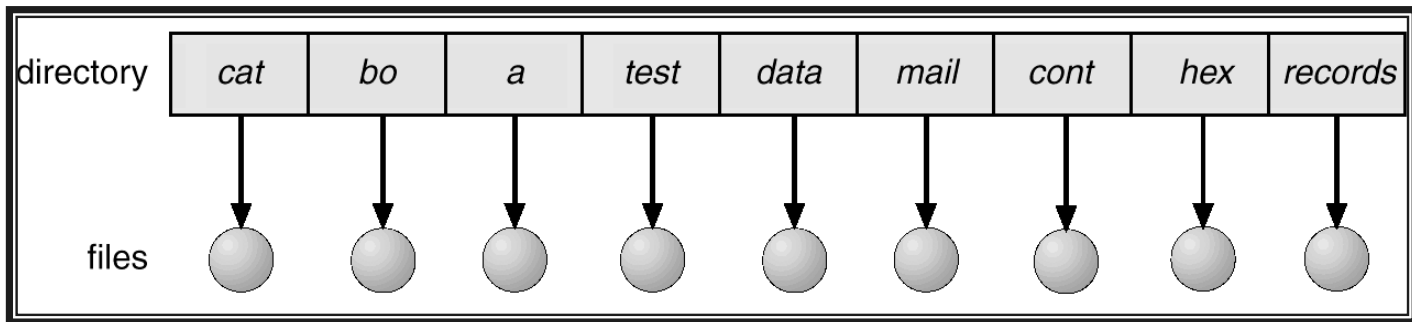
- Ricerca di un file
- Creazione di un file
- Cancellazione di un file
- Elencare i file nella directory
- Rinominare un file
- Attraversare il file system

Organizzazione Logica di una Directory

- **Efficienza:** accedere un file velocemente.
- **Identificazione:** conveniente per gli utenti.
 - Due utenti possono usare nomi uguali per file diversi.
 - Lo stesso file può avere nomi diversi.
- **Raggruppare:** classificare insieme i file in base alle loro proprietà, (es., tutti i programmi Java, tutti i videogiochi, ecc.)

Directory a singolo livello

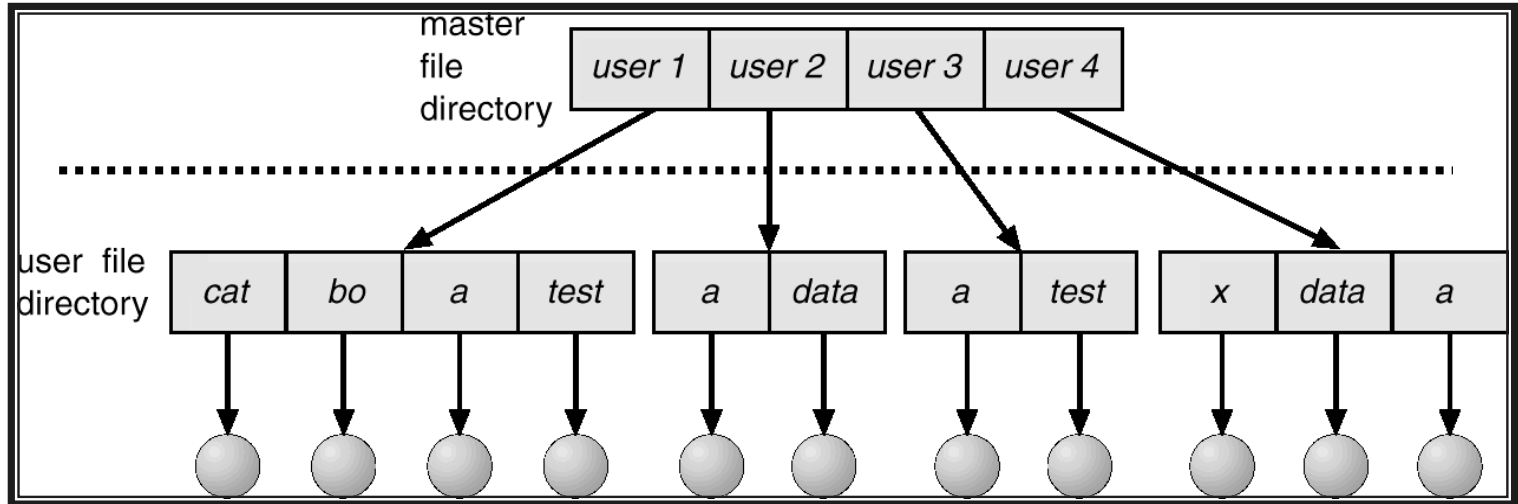
- Una singola directory per tutti i file.



- Problema di identificazione
- Problema di raggruppamento

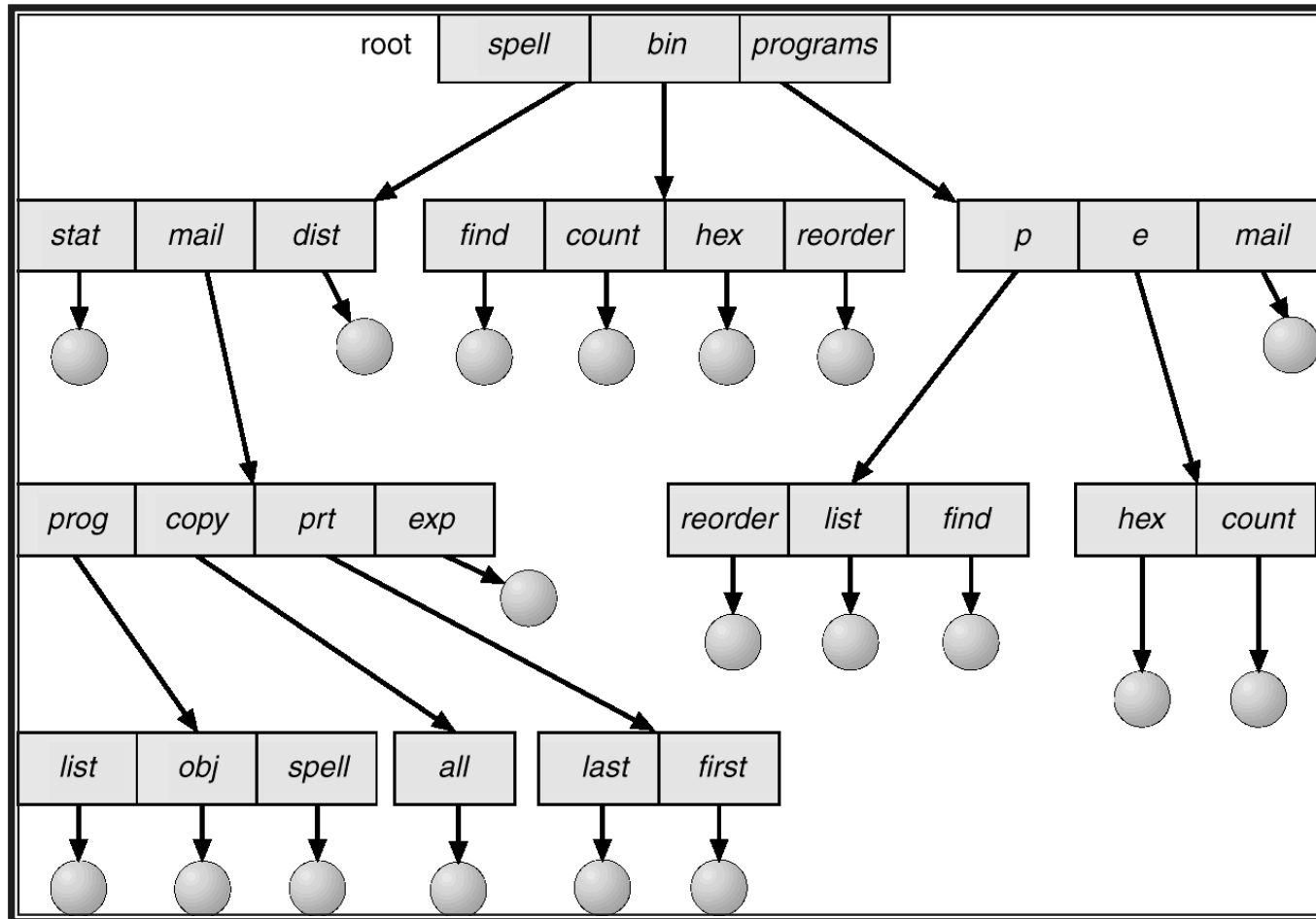
Directory a due livello

- Directory separate per ogni utente.



- Esiste il path name
- Si può avere lo stesso nome di file per utenti differenti
- La ricerca è efficiente
- Problema di raggruppamento

Directory con struttura ad albero



Directory con struttura ad albero

- Path name a più livelli
- Si può avere lo stesso nome di file per utenti differenti
- La ricerca è efficiente
- Il raggruppamento è possibile
- Directory corrente (directory di lavoro - *home*)
 - **cd** /spell/mail/prog *spostamento assoluto*
 - **type** list *file locale*

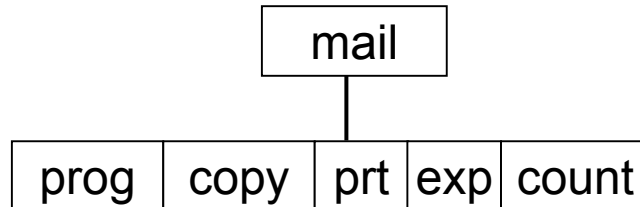
Directory con struttura ad albero

- Path name **Assoluto** o **relativo**
 - La creazione di un nuovo file avviene nella directory corrente.
 - Cancellare un file
- > **rm** <file-name>
- La creazione di una nuova directory avviene nella directory corrente.

> **mkdir** <dir-name>

Esempio: se la directory corrente è: **/mail**

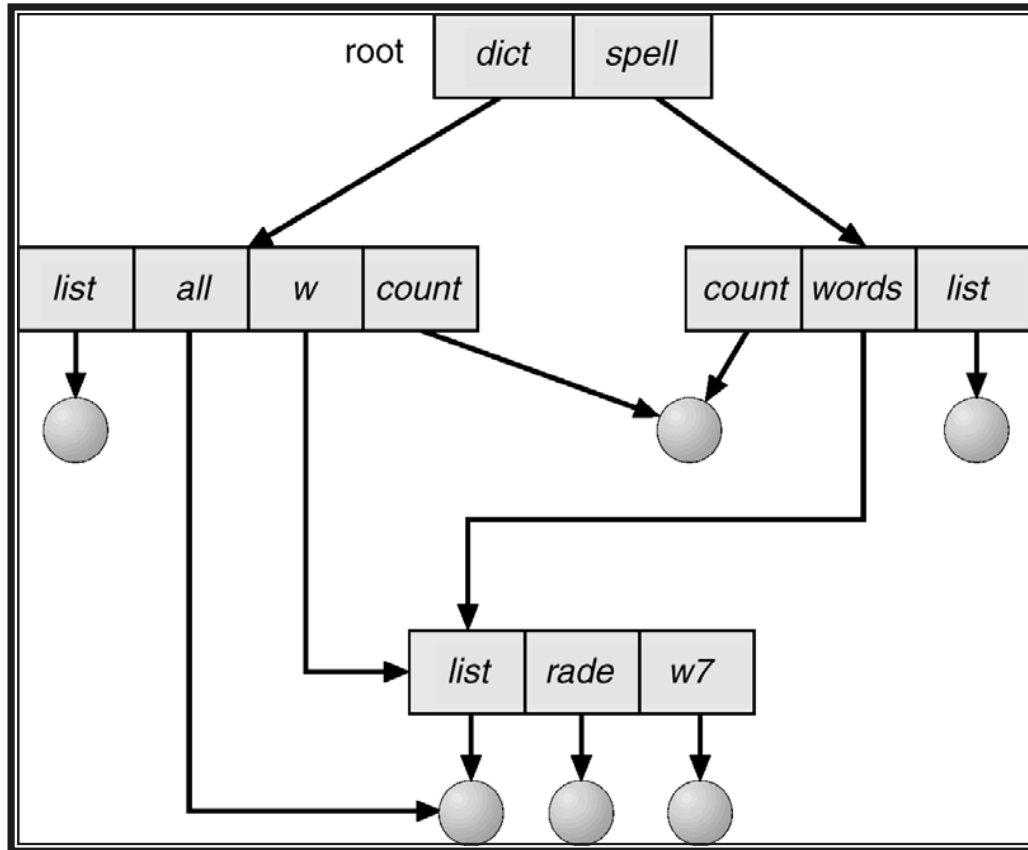
> **mkdir** count



La cancellazione di “mail” \Rightarrow La cancellazione di tutto il sottoalbero “mail”.

Directory a grafo aciclico

- Ha sotto-directory e file condivisi.



Directory a grafo aciclico

- Due nomi differenti per uno stesso file (*aliasing*)
- Se *dict* cancella *list* \Rightarrow puntatore "appeso".

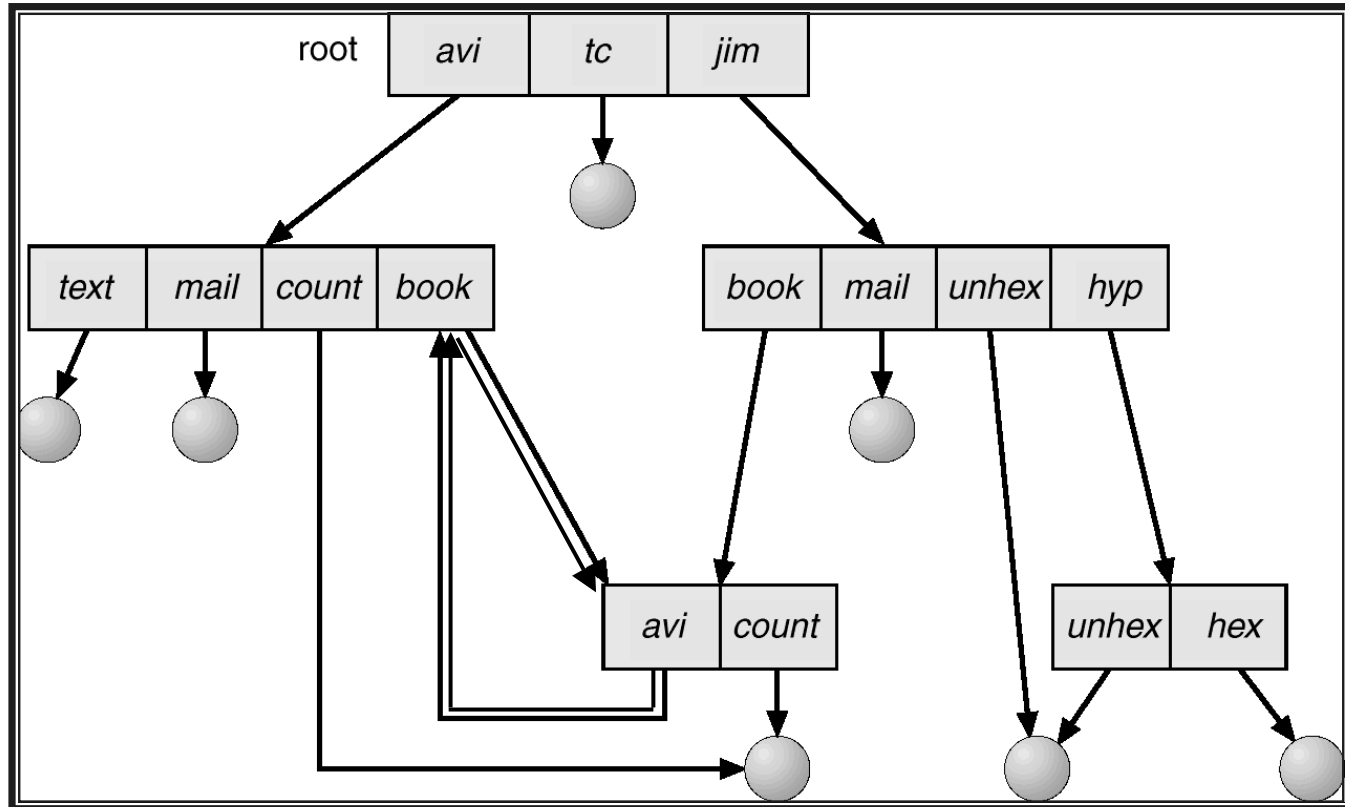
Soluzioni:

- Puntatori all'indietro, per cancellare tutti i puntatori all'elemento eliminato.
- Problema della dimensione (quanti puntatori?).
- Soluzione basata su contatori.

Directory a grafo generale

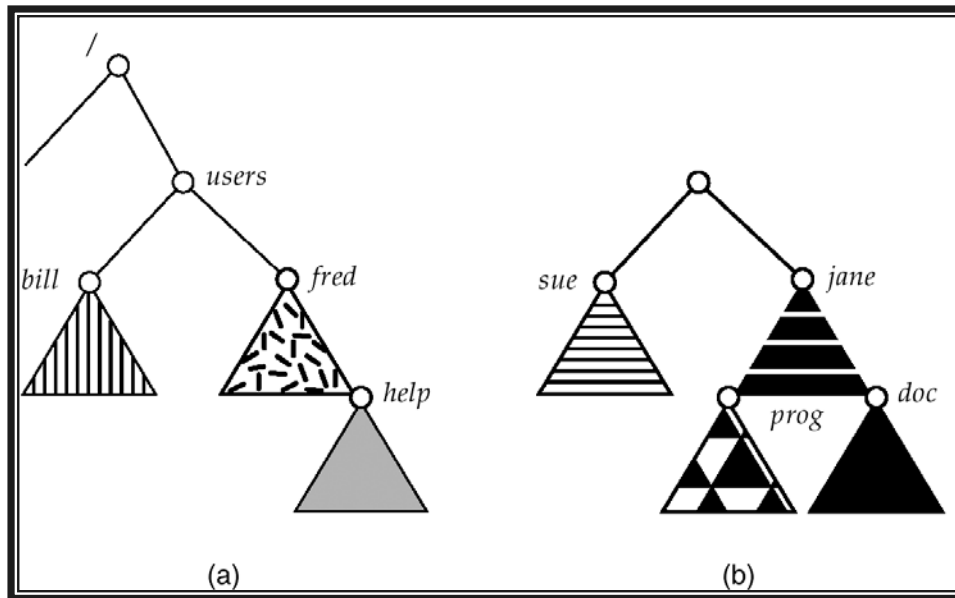
- Come garantire l'assenza di cicli?
 - Permettere link a file non a sotto-directory.
 - Garbage collection.
 - Ogni volta che viene creato un file si può usare un algoritmo per la rilevazione di cicli che eviti i cicli. (costoso)

Directory a grafo generale

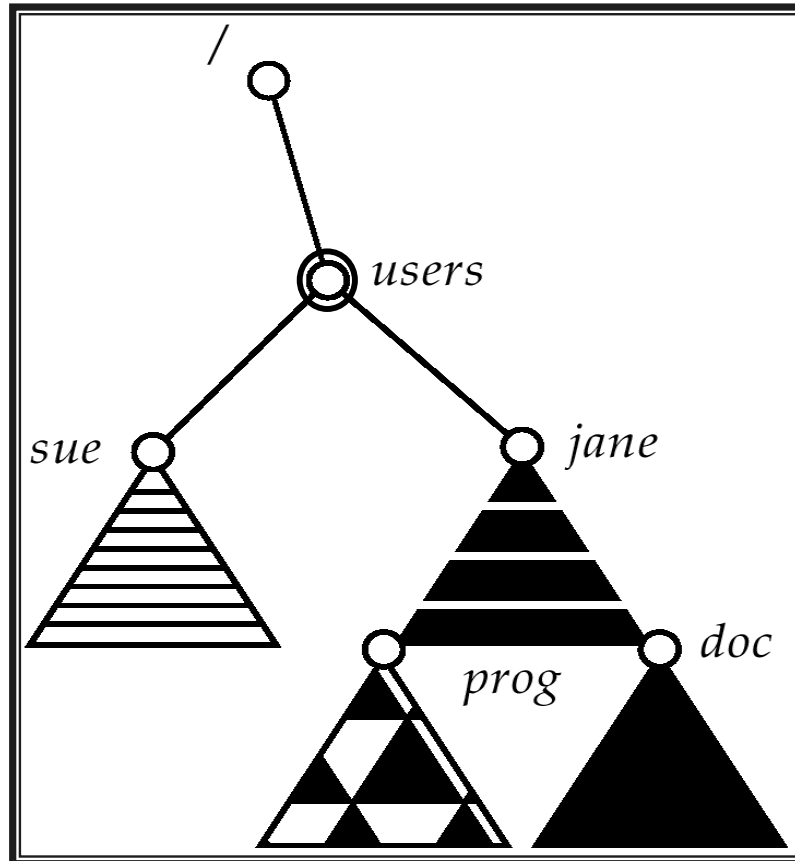


Montaggio di un File System

- Un file system deve essere **montato** prima di poter essere acceduto.
- Un file system non montato (Figura b) deve essere montato in una directory di montaggio (***mount point***).



Punto di mount



Mount point: `/users`

Condivisione di file

- La condivisione di file su sistemi multi-utente è utile.
- Può essere realizzata tramite uno schema di *protezione*.
- Su sistemi distribuiti, i file possono essere condivisi tramite la rete.
- *Network File System* (NFS) è un modello e una realizzazione pratica di file system distribuito.

Protezione

- Il proprietario/creatore di un file deve poter controllare:
 - Le operazioni possibili sul file
 - Chi li può fare.

- Tipo of accesso
 - Lettura
 - Scrittura
 - Esecuzione
 - Aggiunta
 - Cancellazione
 - Lista.

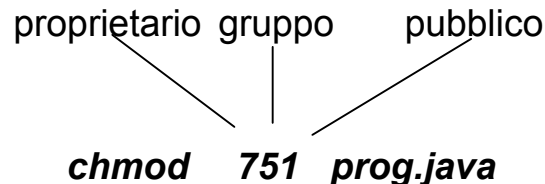
- Metodi
 - Liste di accesso
(gestore, utenti)
 - Password
(file, directory)

Liste di Accesso e Gruppi

- Modo di accesso: read, write, execute
- Tre classi di utenti

a) accesso proprietario	7	⇒	RWX 1 1 1
b) accesso gruppo	5	⇒	RWX 1 0 1
c) accesso pubblico	1	⇒	RWX 0 0 1

- Si chiede al gestore di creare un gruppo, G, e aggiungere alcuni utenti al gruppo.
- Per un file (es: *prog.java*) o sottodirectory, si definisce un accesso appropriato.



Domande

- Discutere l'implementazione di file strutturati nel caso in cui il file system offre solo file di testo.
- Quali operazioni effettua il file system per eseguire una open ?
- Quali sono i problemi principali nella gestione dei file system con struttura a grafo generale ?
- Definire le protezioni di lettura, scrittura ed esecuzione sui file di tre directory *d1*, *d2*, *d3*, la prima con solo accesso del proprietario, la seconda anche con accesso del gruppo ma senza possibilità di modifiche e la terza con accesso pubblico su tutti i file ma solo in lettura.