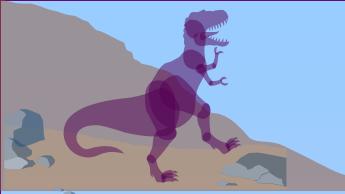


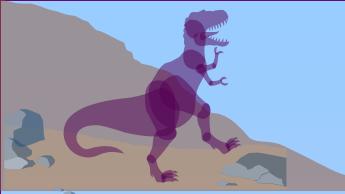
Capitolo 10: Interfaccia del File-System

- Concetto di file
- Metodi di accesso
- Struttura della directory
- Montaggio di un file system
- Condivisione di file
- Protezione



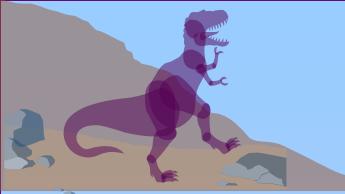
Concetto di file

- L'unità logica di memorizzazione vista dall'utente è il **file** e il sistema operativo provvede ad allocare i file su dispositivi di memorizzazione e ad eseguire le operazioni sui file richieste dall'utente.
- Un **file** è un insieme di informazioni correlate e registrate su memoria secondaria, cui è stato assegnato un nome.
- Tipi di file:
 - ☞ Dati
 - ▀ Numerici
 - ▀ Alfabetici
 - ▀ Binari
 - ☞ Programmi
 - ▀ Sorgente
 - ▀ Oggetto

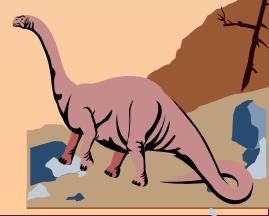


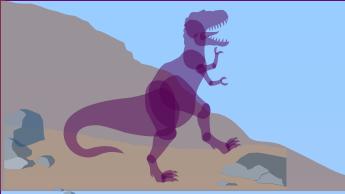
Struttura dei file

- Le informazioni contenute in un file sono definite dal suo creatore e possono essere di molti tipi.
- Un file ha una **struttura** definita secondo il tipo:
 - ☞ un file di testo è formato da una sequenza di caratteri organizzati in righe, un file eseguibile consiste in una serie di sezioni di codice che il loader può caricare in memoria ed eseguire, etc..
 - ☞ UNIX, considera ciascun file come una sequenza di bytes, lasciando ai programmi applicativi il compito di contenere il codice per interpretare la struttura di un file di input.
- In generale un file potrà essere:
 - ☞ vuoto,
 - ☞ visto come una sequenza di parole o bytes,
 - ☞ visto come una sequenza di record logici.



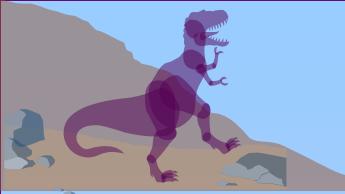
Struttura dei file (II)

- Struttura semplice
 - ☞ Lunghezza fissa
 - ☞ Lunghezza variabile
 - Struttura complessa
 - ☞ Documenti formattati
 - ☞ File rilocabili
 - Possiamo simulare gli ultimi due con i primi metodi inserendo appropriati caratteri di controllo.
 - Chi decide:
 - ☞ Sistema Operativo
 - ☞ Programmi
- 

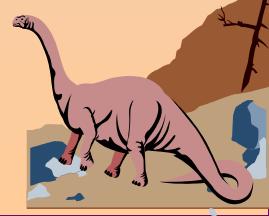


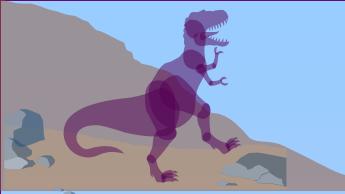
Attributi dei file

- Un file ha alcuni attributi che possono variare secondo il sistema operativo, che tipicamente comprendono:
 - ☞ **Nome** – unica informazione tenuta in una forma leggibile dagli utenti.
 - ☞ **Identificatore** – etichetta unica, in genere numerica, che identifica il file all'interno del file system: il nome utilizzato dal sistema per il file
 - ☞ **Tipo** – necessario per sistemi che supportano tipi differenti.
 - ☞ **Locazione** – puntatore alla locazione del file sul dispositivo.
 - ☞ **Dimensione** – dimensione attuale del file, generalmente in bytes.
 - ☞ **Protezione** – informazioni di controllo: chi può leggere, scrivere o eseguire.
 - ☞ **Ora , data e identificazione dell'utente** – creazione, modifica ed ultimo uso, dati utili alla protezione e per monitorare l'utilizzo.
 - Informazioni sui file sono conservate nella struttura della directory, che risiede a sua volta nella memoria secondaria.
- 



Operazioni sui file

- **Creazione di un file** - necessita di due passaggi:
 - ☞ trovare lo spazio nel file system e
 - ☞ creare un nuovo elemento nella directory in cui registrare il nome del file, la sua posizione ed altre informazioni.
 - **Scrittura di un file** - una chiamata al sistema specificherà il nome del file e le informazioni che si vogliono scrivere.
 - ☞ Dato il nome del file, il sistema cerca la sua posizione nella directory, mantenendo un puntatore alla posizione del file in cui deve avvenire la prossima scrittura.
 - **Lettura di un file** - una chiamata al sistema specificherà il nome del file e la posizione nella memoria dove leggere.
 - ☞ Dato il nome del file, il sistema cerca la sua posizione nella directory, mantenendo un puntatore alla posizione del file in cui deve avvenire la prossima lettura.
- 

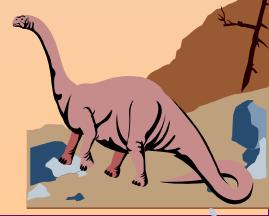


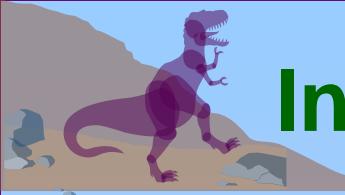
Operazioni sui file (II)

- **Riposizionamento in un file (seek)** - si ricerca l'elemento appropriato nella directory e si assegna un nuovo valore al puntatore alla posizione corrente nel file.
 - ☞ Non richiede nessuna operazione di I/O.
- **Cancellazione di un file** - si cerca l'elemento della directory associato al file designato, si rilascia lo spazio associato al file e si elimina l'elemento della directory.
- **Troncamento di un file** - consente di mantenere immutati gli altri attributi del file,
 - ☞ pur azzerando la lunghezza del file e rilasciando lo spazio occupato.
- Altre operazioni di base comprendono:
 - ☞ l' aggiunta di nuove informazioni alla fine di un file esistente (*append*),
 - ☞ la ridenominazione di un file esistente (*rename*).



Operazioni sui file (III)

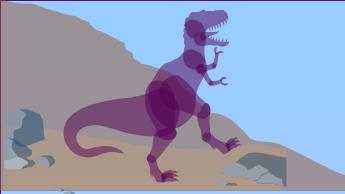
- S.O. mantiene una tabella contenente informazioni riguardanti tutti i file aperti (*tabella dei file aperti*).
 - Un'operazione di *open* inserirà una nuova entry nella tabella dei file aperti, una di *close* ne rimuoverà una.
 - La realizzazione di *open* e *close* in un ambiente multiutente è complicata dal fatto che più utenti possono aprire un file contemporaneamente.
 - S.O. introduce due livelli di tavelle:
 - ☞ una per ciascun processo (tutti i file aperti dal processo, con puntatori per le successive read o write)
 - ☞ una di sistema (a cui puntano le tavelle dei processi e che contiene le informazioni indipendenti dai processi: posizione del file nel disco, date degli accessi, dimensioni, etc.)
 - La tabella dei file aperti ha anche un *contatore delle aperture* associato ad ogni file, ogni *close* lo decrementa.
 - Quando raggiunge il valore zero il file non è più in uso e si elimina l'elemento corrispondente dalla tabella dei file aperti.
- 



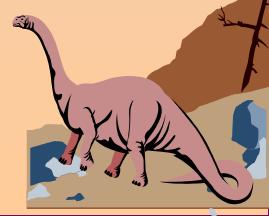
Informazioni associate ad un file aperto

- **Puntatore al file:** ultima posizione di lettura e scrittura sotto forma di un puntatore alla posizione corrente nel file.
 - ☞ Unico per ogni processo che opera sul file
- **Contatore dei file aperti :** tiene traccia del numero di open e close e raggiunge il valore zero dopo l'ultima chiusura
 - ☞ indicando che il sistema può rimuovere l'elemento dalla tabella.
- **Posizione nel disco del file :** l'informazione necessaria per localizzare il file è mantenuta nella memoria,
 - ☞ per evitare di doverla prelevare dal disco ad ogni operazione.
- **Diritti di accesso :** Ciascun processo apre un file in uno dei modi di accesso.
 - ☞ Questa informazione è contenuta nella tabella del processo in modo che S.O. possa permettere o negare le successive richieste di I/O.





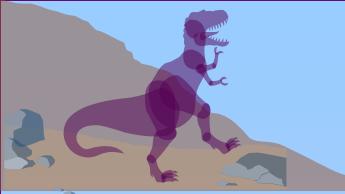
Tipi di file

- Il file system ed in generale S.O. devono saper riconoscere e gestire i diversi tipi di file.
 - Tecnica comune, per riconoscere il tipo:
 - ☞ nome.estensione
 - In alcuni sistemi il nome può essere costituito da 8 caratteri e l'estensione da 3 caratteri.
 - In altri, invece, ogni file ha un suo tipo ma possiede anche un attributo di creazione contenente il nome del programma che lo ha creato.
- 

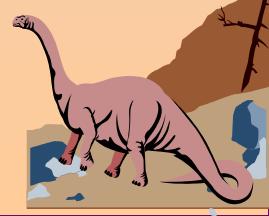


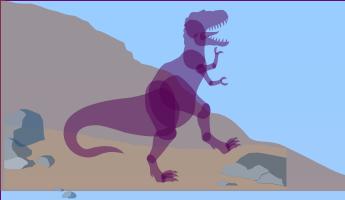
Comuni tipi di file

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 compressed, for archiving or storage
multimedia	mpeg, mov, rm	binary file containing audio or A/V information



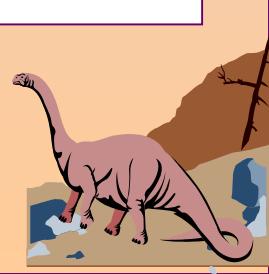
Cosa succede in DOS

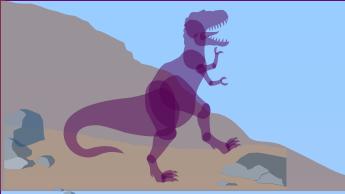
- In MS-DOS i file possono essere suddivisi in
 - ☞ **File eseguibili** che contengono programmi eseguibili sotto il controllo del DOS;
 - ☞ **File dati** che contengono informazioni utilizzabili dai file eseguibili.
 - I file eseguibili possono avere tre possibili estensioni:
 - ☞ COM o EXE - se contengono programmi eseguibili;
 - ☞ BAT - se contengono comandi DOS.
 - I file dati possono avere una estensione qualsiasi (anche mancante) e contengono informazioni non direttamente utilizzabili dall'elaboratore. Il loro utilizzo è legato alla esecuzione di un programma capace di interpretarne il contenuto.
 - In alcuni casi il programma che utilizza i file dati imposta o richiede una particolare estensione.
 - I file dati sono creati da file eseguibili.
- 



Cosa succede in UNIX

- In UNIX, un file è una sequenza di byte, ed è eseguito solo se ha il formato appropriato. Nella figura vediamo un semplice file binario eseguibile.
- Il file ha cinque sezioni: intestazione, testo, dati, bit di rilocazione, tabella dei simboli.
- L'intestazione inizia con una parte detta **magic number** che identifica un file come file eseguibile (per prevenire l'esecuzione accidentale di un file non in questo formato).
- Seguono alcuni interi di 16 bit che danno la dimensione delle varie parti del file, l'indirizzo per la partenza dell'esecuzione e qualche bit di flag.
- Dopo troviamo il testo e i dati del programma: queste parti sono caricate in memoria e rilocate usando il bit di rilocazione. La tabella dei simboli è usata per il debugging.





Metodi di accesso

■ Accesso sequenziale

*read next
write next
reset
no read after last write
(rewrite)*

■ Accesso diretto

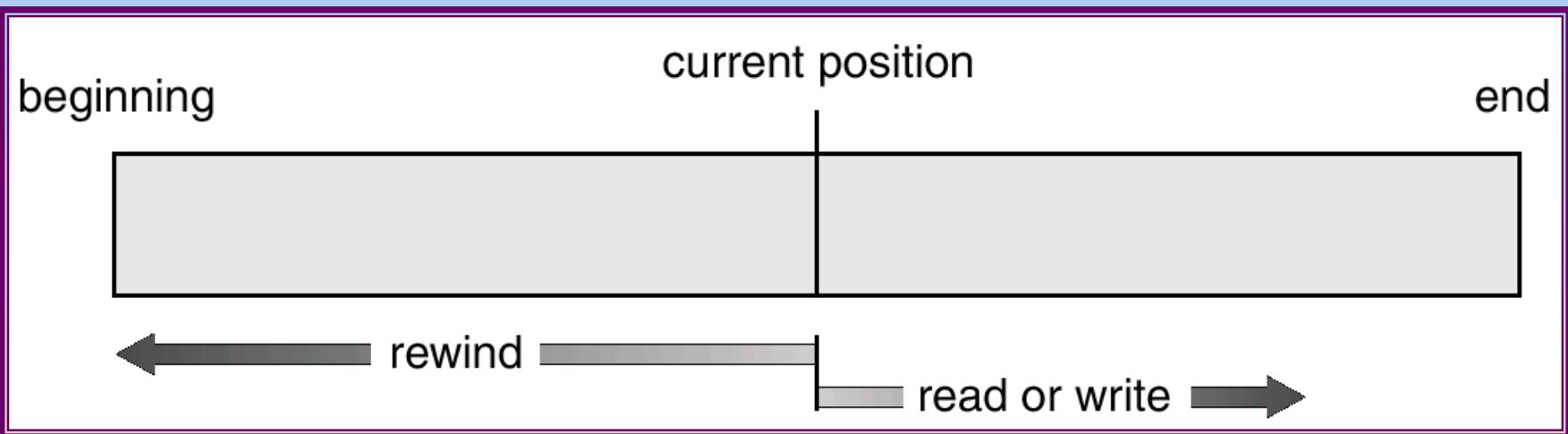
*read n
write n
position to n
read next
write next
rewrite n*

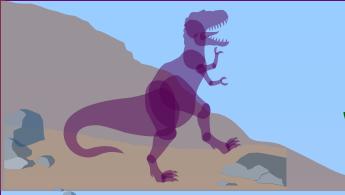
■ Altri metodi di accesso

n = numero di blocco relativo



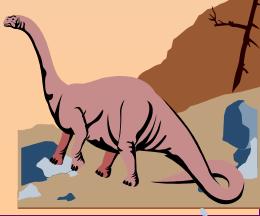
File ad accesso sequenziale

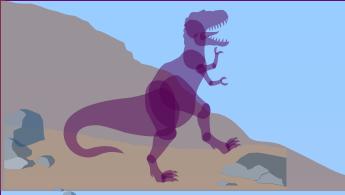




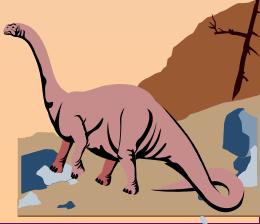
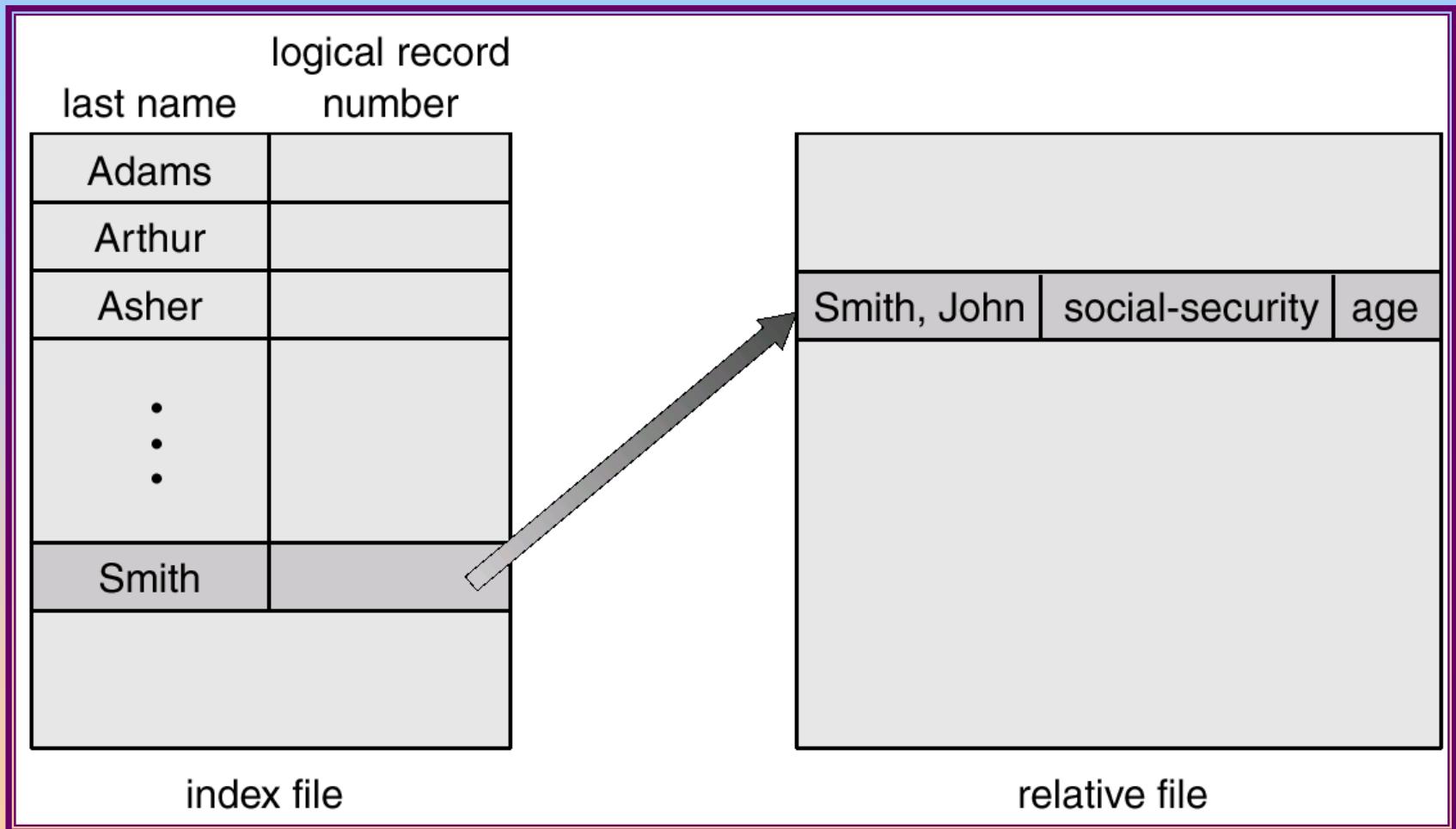
Simulazione dell'accesso sequenziale su un file ad accesso diretto

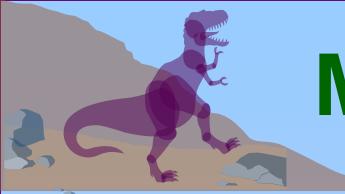
sequential access	implementation for direct access
<i>reset</i>	$cp = 0;$
<i>read next</i>	<i>read cp;</i> $cp = cp+1;$
<i>write next</i>	<i>write cp;</i> $cp = cp+1;$



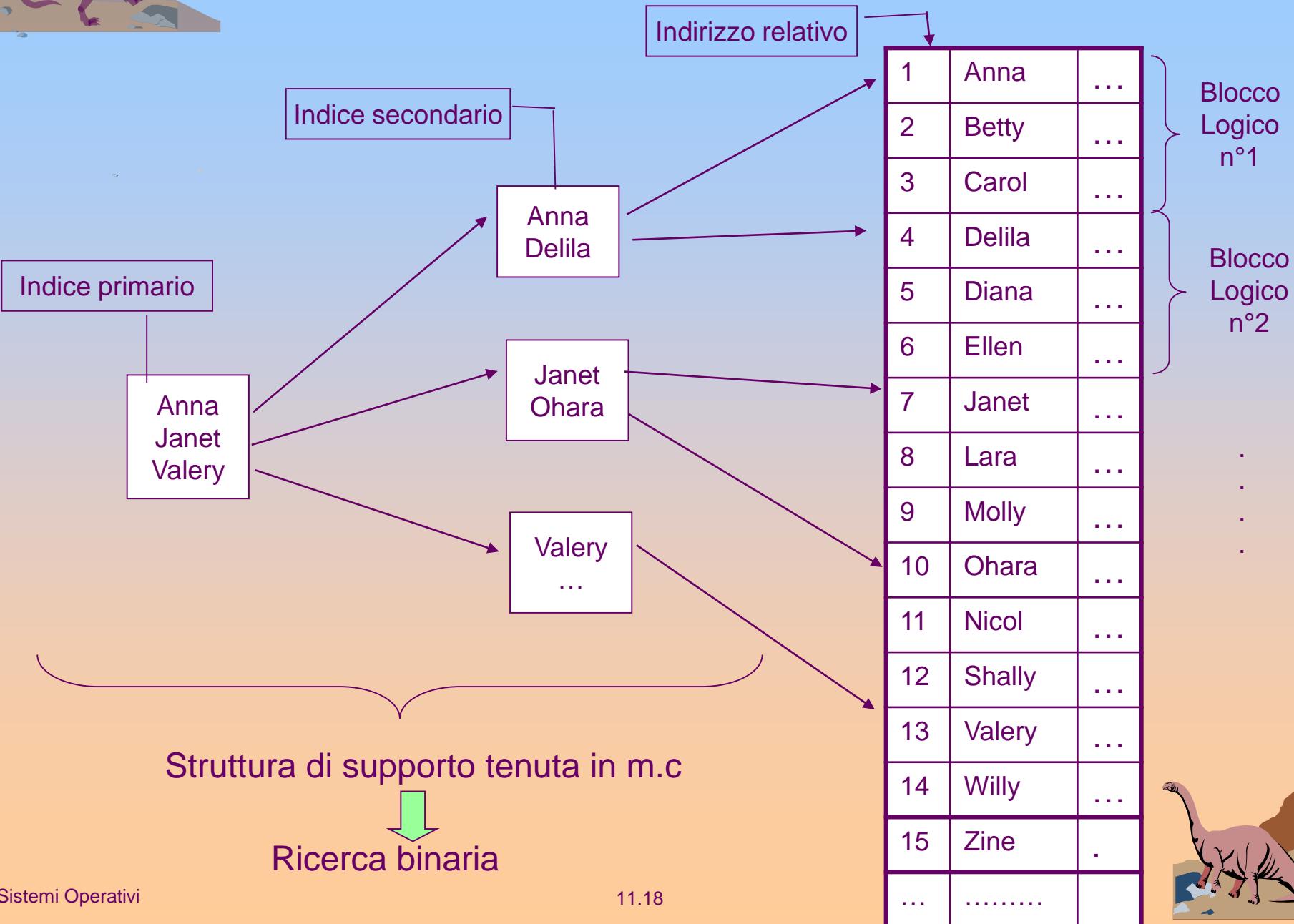


Esempio di indice e relativo file





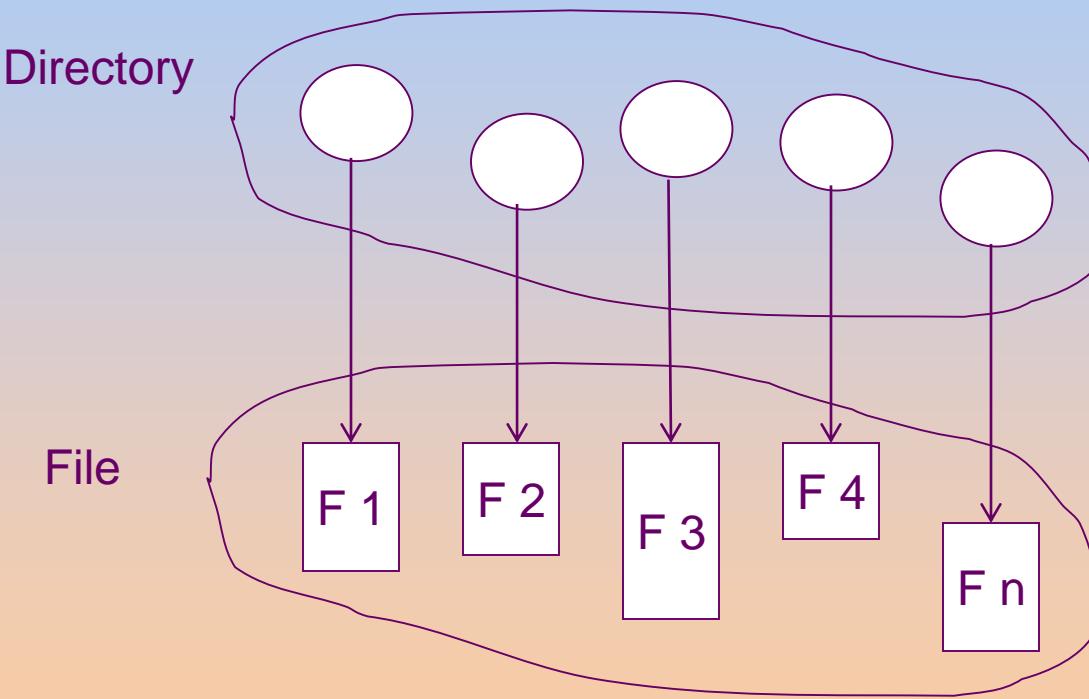
Metodo ISAM (Indexed Sequential Access Method)





Struttura della directory

- I file sono di solito raggruppati in base alle loro caratteristiche o secondo i più vari criteri di comodità.
- Le directory consentono di ordinare il contenuto del disco secondo le necessità dello stesso utente..

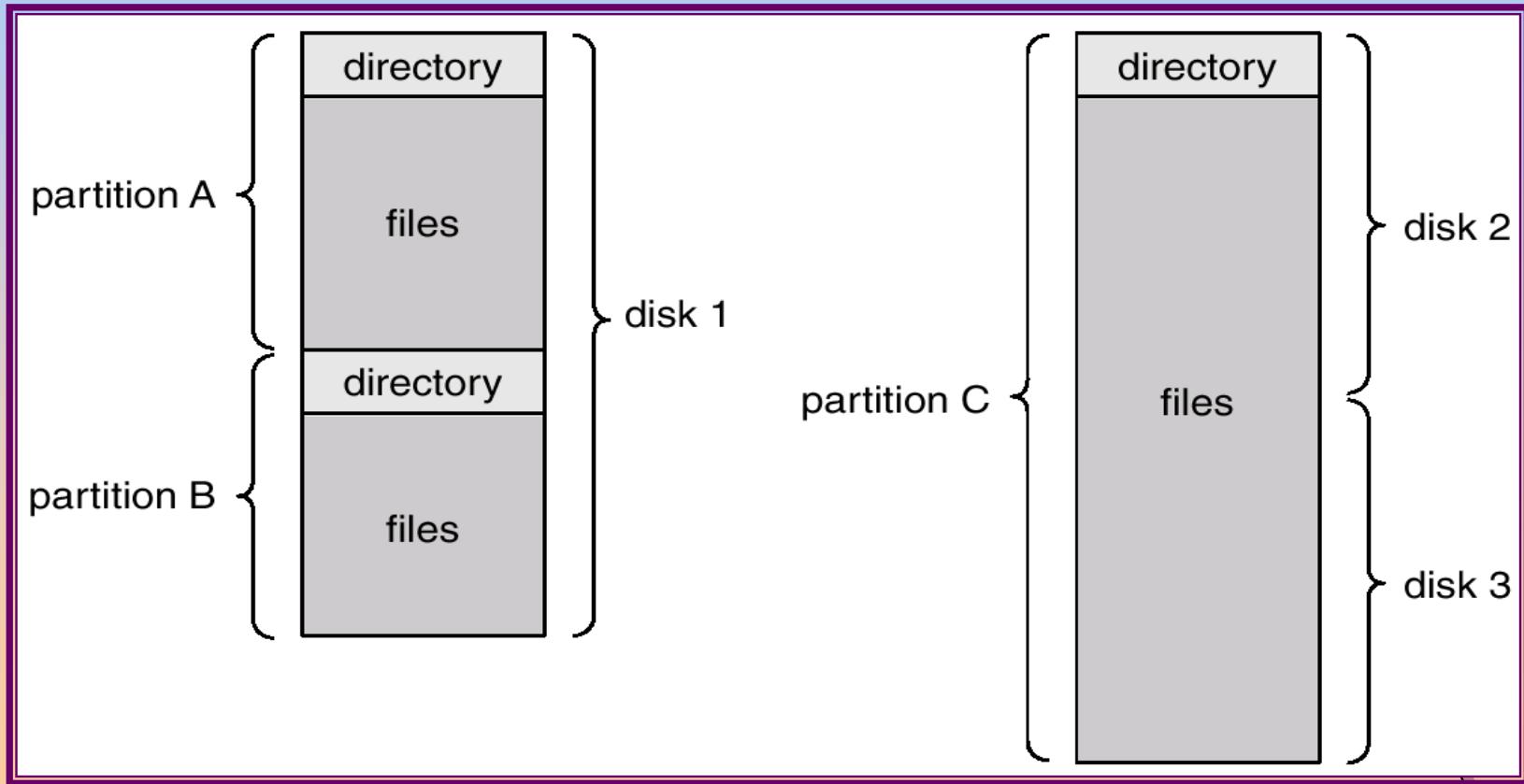


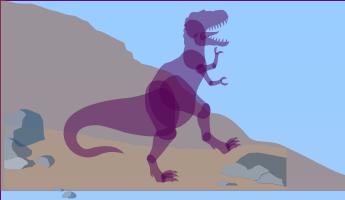
- Sia la struttura della directory che i file risiedono sul disco.
 - Le copie di backup di queste due strutture sono tenute su nastri.
- 



Una tipica organizzazione di file system

- Il file system viene suddiviso in aree separate dette partizioni, ognuna delle quali contiene file e directory.
- Tipicamente ciascun disco contiene almeno una partizione.

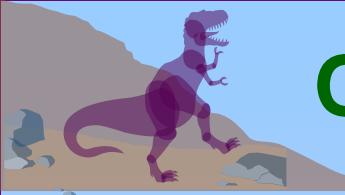




Informazioni nella directory

- Le informazioni sui file, di ciascuna partizione, sono mantenute negli elementi della **directory del dispositivo** (device directory) o **tabella dei contenuti del volume**.
- Questa registra le informazioni di tutti i file della partizione:
 - Nome
 - Tipo
 - Indirizzo
 - Lunghezza corrente
 - Lunghezza massima
 - Data di ultimo accesso
 - Data di ultima modifica
 - ID del proprietario
 - Informazioni di protezione

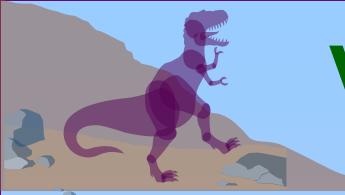




Operazioni che si possono eseguire su una directory

- La directory può essere considerata come una tabella di simboli che traduce i nomi dei file negli elementi in essa contenuti, per cui può essere organizzata in diversi modi per permettere operazioni diverse.
- Operazioni che si possono eseguire su directory sono:
 - Ricerca di un file
 - Creazione di un file
 - Cancellazione di un file
 - Elencazione di una directory
 - Ridenominazione di un file
 - Attraversamento del file system

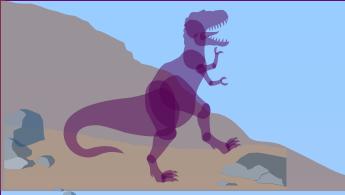




Vantaggi dell'organizzazione logica di directory

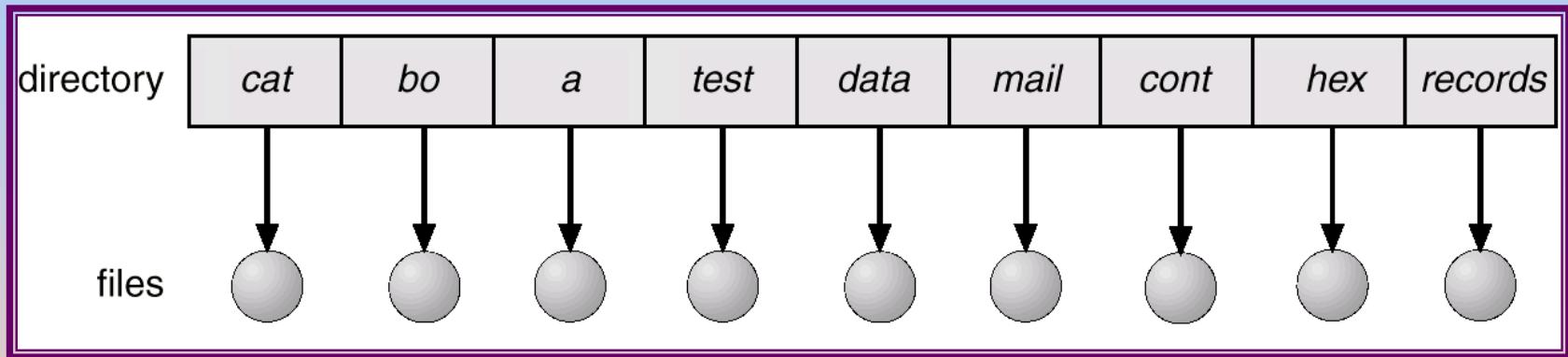
- **Efficienza** – un file viene localizzato velocemente.
- **Nomi** –conveniente uso per gli utenti.
 - ☞ Due utenti possono assegnare lo stesso nome a file differenti.
 - ☞ Lo stesso file può essere raggiungibile con nomi diversi.
- **Gruppo** – gruppi logici di files in base alle proprietà (e.s., tutti i programmi Java, tutti i giochi, ...)





Directory a singolo livello

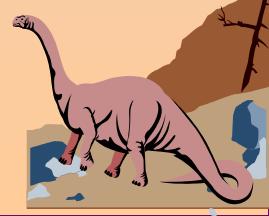
- Struttura più semplice.
- Tutti i file sono contenuti nella stessa directory

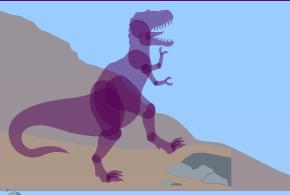


Problemi se aumentano il numero di file e utenti

Problema dei nomi (file con nomi unici)

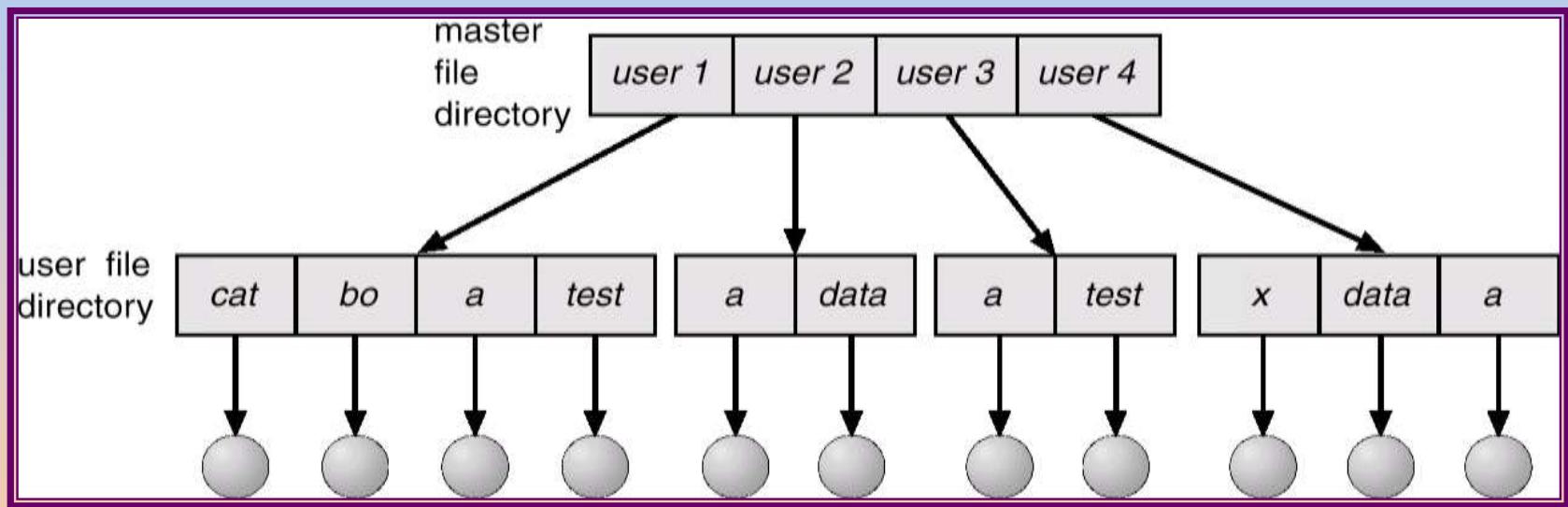
Problema dei gruppi (nessuna distinzione di proprietà)





Directory a due livelli

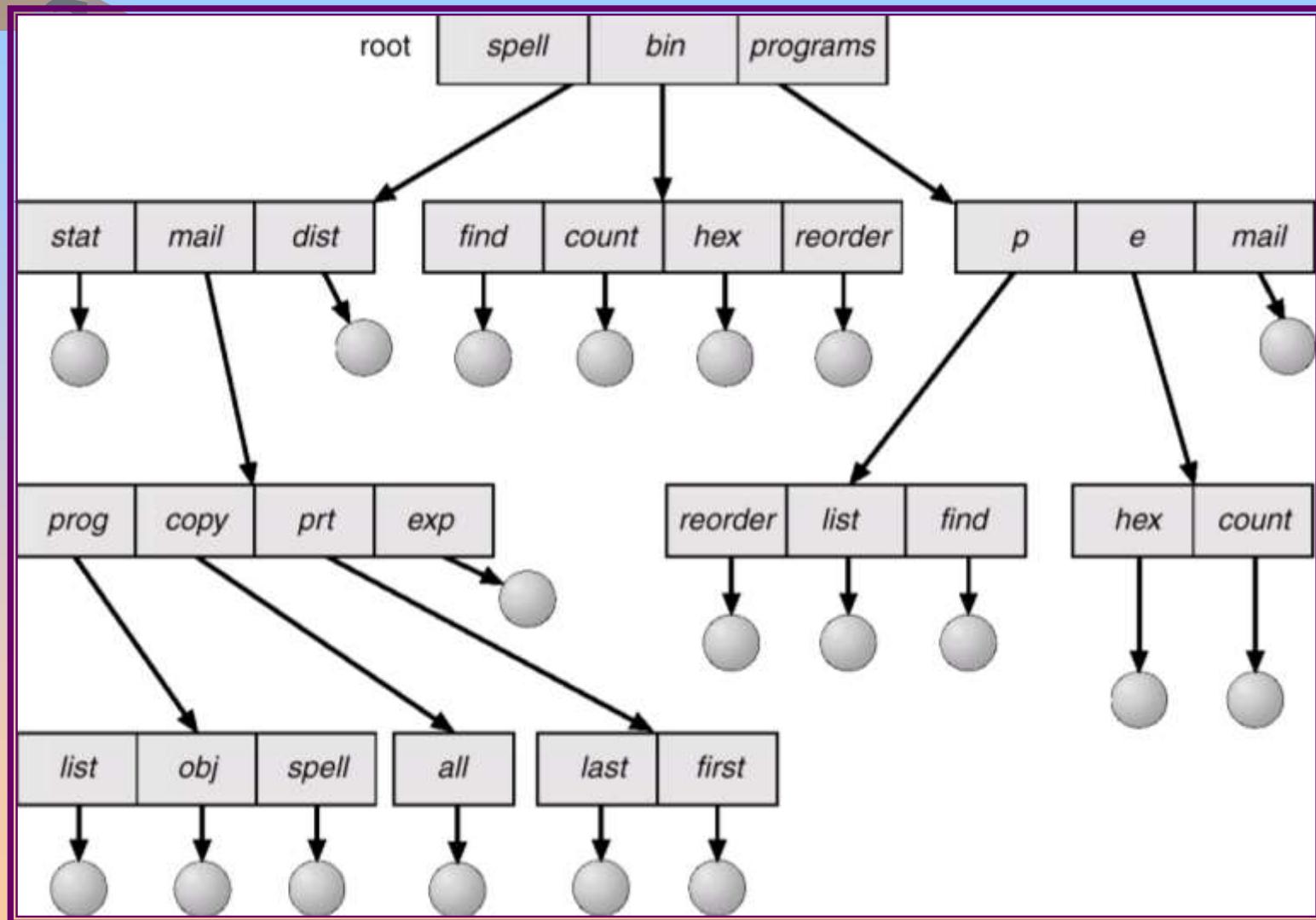
- Directory separate per ogni utente, dette UFD.
- Tutte le UFD hanno una struttura simile e contengono solo i files del proprietario.
- Appena un utente effettua un login la ricerca viene fatta nella MFD.



- File con lo stesso nome creati da utenti diversi
- Nome di percorso (Path name)
- Ricerca efficiente



Directory con struttura ad albero





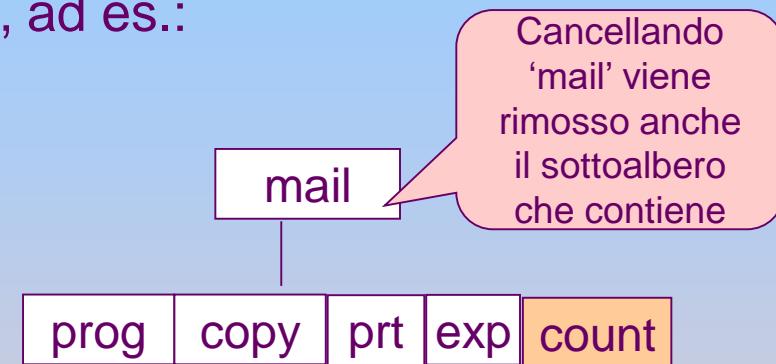
Directory con struttura ad albero (II)

- Utenti diversi possono creare sottodirectory e file direttamente dalla directory corrente, ad es.:

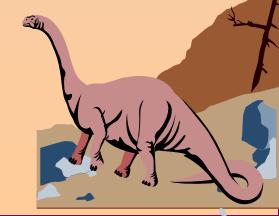
mkdir <dir-name>

se nella directory **/mail**

mkdir count



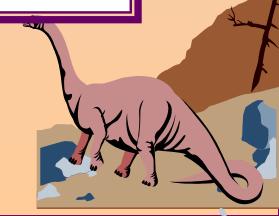
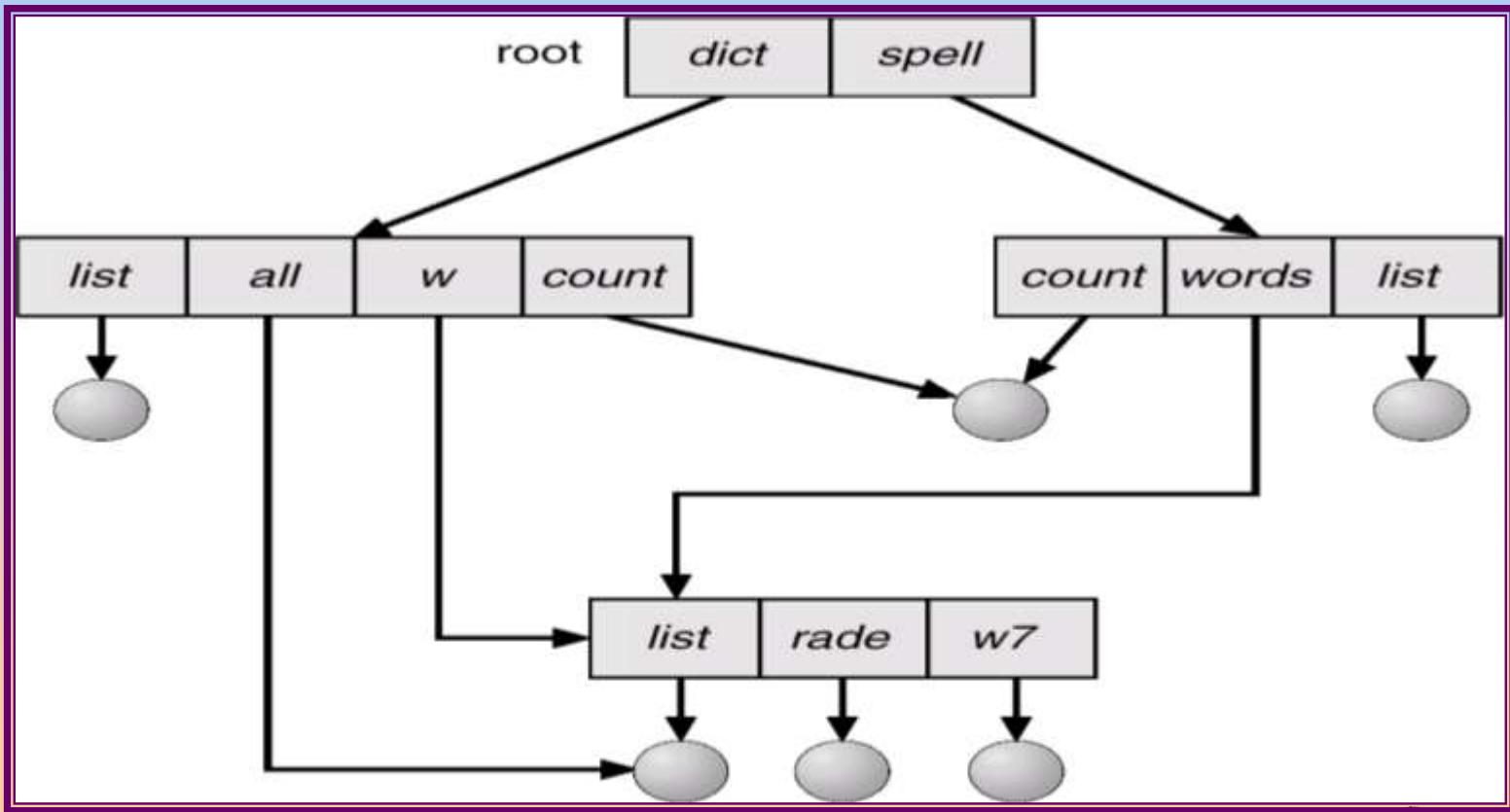
- Ogni utente dispone di una directory corrente che contiene la maggior parte dei file di interesse corrente.
- Se si vuole un file che non è nella directory corrente bisogna o indicare il nome di percorso completo (pathname) del file oppure cambiare directory corrente, facendo diventare tale la directory che contiene il file desiderato.
- I pathname possono essere assoluti o relativi
- Cancellazione di un file: **rm <file-name>**
- Cancellazione di una directory: **rmdir <dir-name>**





Directory con struttura a grafo aciclico

- Permette alle directory di avere **sottodirectory** e **file condivisi**.
- Non si duplicano informazioni ma si usano i link: collegamenti per riferirsi ad esso (ad es.UNIX).



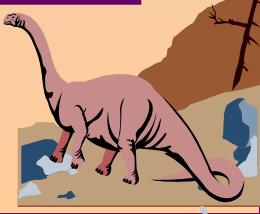
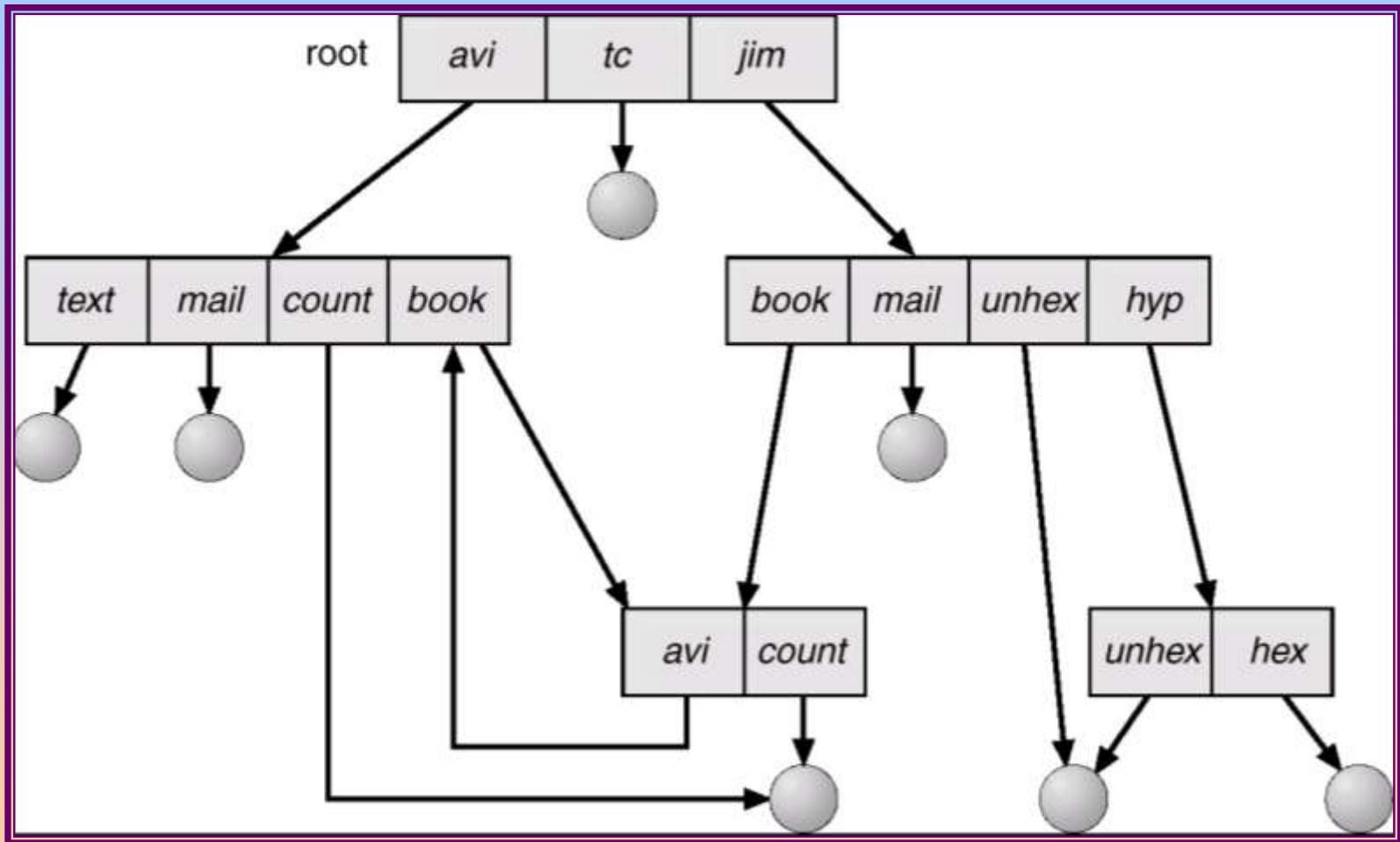


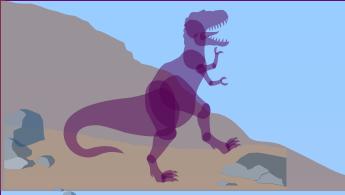
Directory con struttura a grafo aciclico (II)

- Nomi diversi si riferiscono allo stesso file
 - ☞ aliasing -> un file ha più pathname assoluti.
(Problemi in ricerca e in copie di backup)
- La cancellazione causa molti problemi:
 - ☞ Se un file condiviso viene cancellato, riallocando lo spazio che occupava, potrebbero esistere però puntatori ancora non cancellati ad informazioni nel file.
 - ☞ Se esistono link simbolici a un file condiviso che viene cancellato questi potrebbero rimanere in sospeso.
 - ☞ La soluzione sarebbe quella di usare un contatore di tutti i riferimenti e cancellare del tutto un file, quando richiesto, solo se il contatore raggiunge 0.
- In MS-DOS la struttura di directory è ad albero.



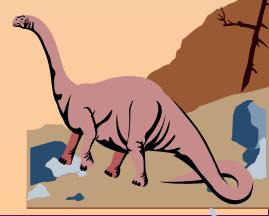
Directory con struttura a Grafo Generale

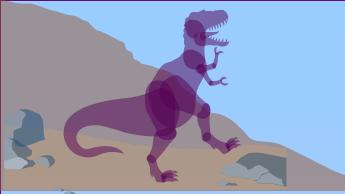




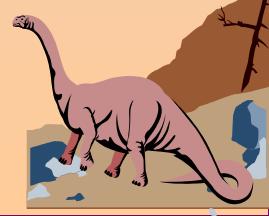
Directory con struttura a Grafo Generale (II)

- Problemi relativi all'attraversamento.
- Come fare a garantire che non vi siano problemi relativi ai cicli?
 - ☞ Consentire solo link ai file e non alle sottodirectory.
 - ☞ Garbage collection (attraversamento del grafo per eliminare tutto ciò che non è più accessibile, cioe' quando il contatore dei riferimenti e' zero).
 - ☞ Ogni volta che viene aggiunto un nuovo link bisogna usare un algoritmo che individui la presenza di cicli.
 - ☒ Se ne trova uno vieta l'operazione, altrimenti la consente.





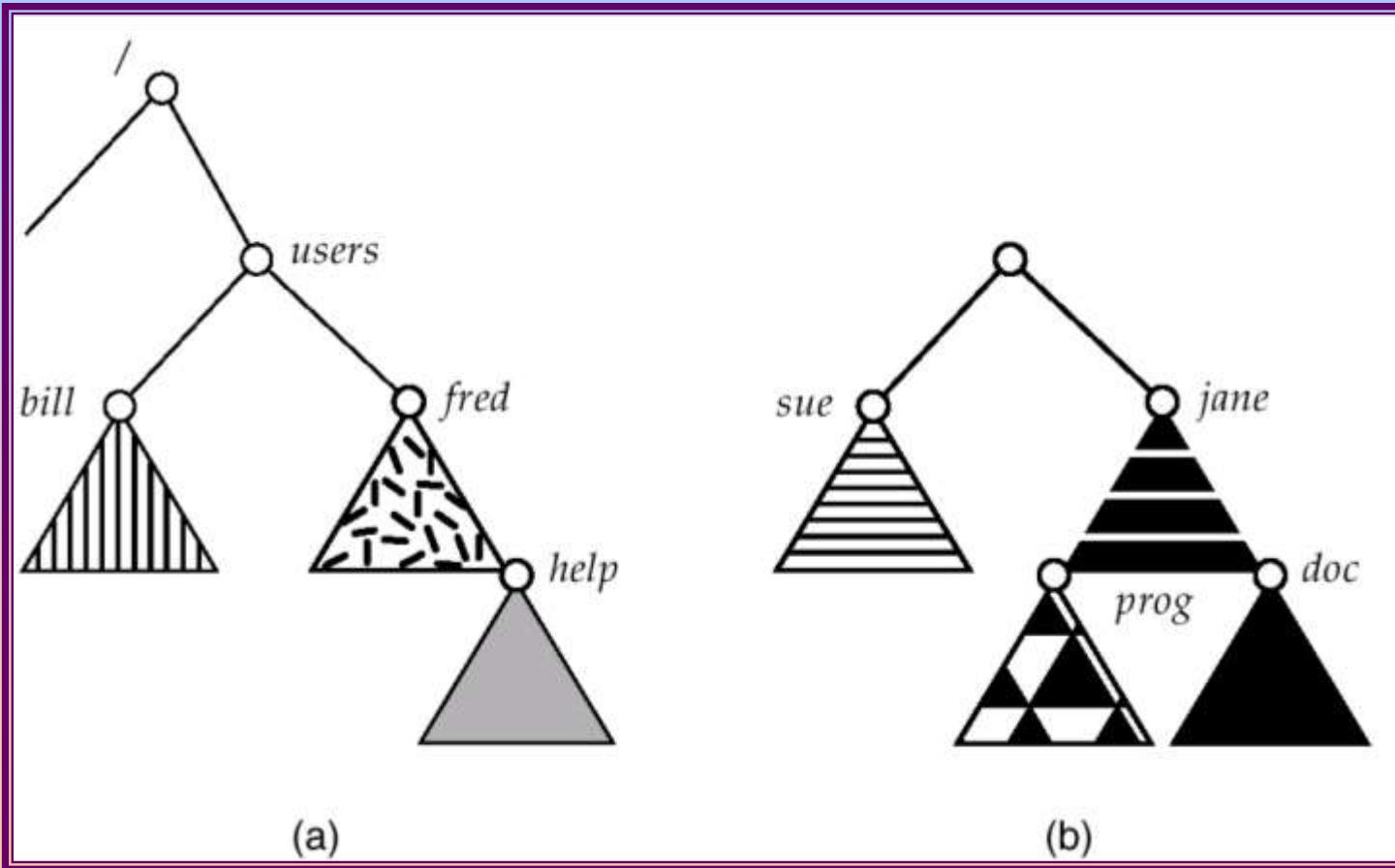
Montaggio del File System

- Il file system deve essere **montato** prima di poter essere messo a disposizione dei processi.
 - Un file system non-montato deve essere montato in un punto detto **punto di montaggio** (mount point).
 - Il sistema operativo riceve il punto della struttura dei file in cui si vuole attaccare file system.
 - ☞ Ad esempio se si vuole montare il file system nel punto /user, successivamente per accedere a file e sottodirectory di quel file system sarà necessario far precedere il nome delle directory da /user.
- 



File System

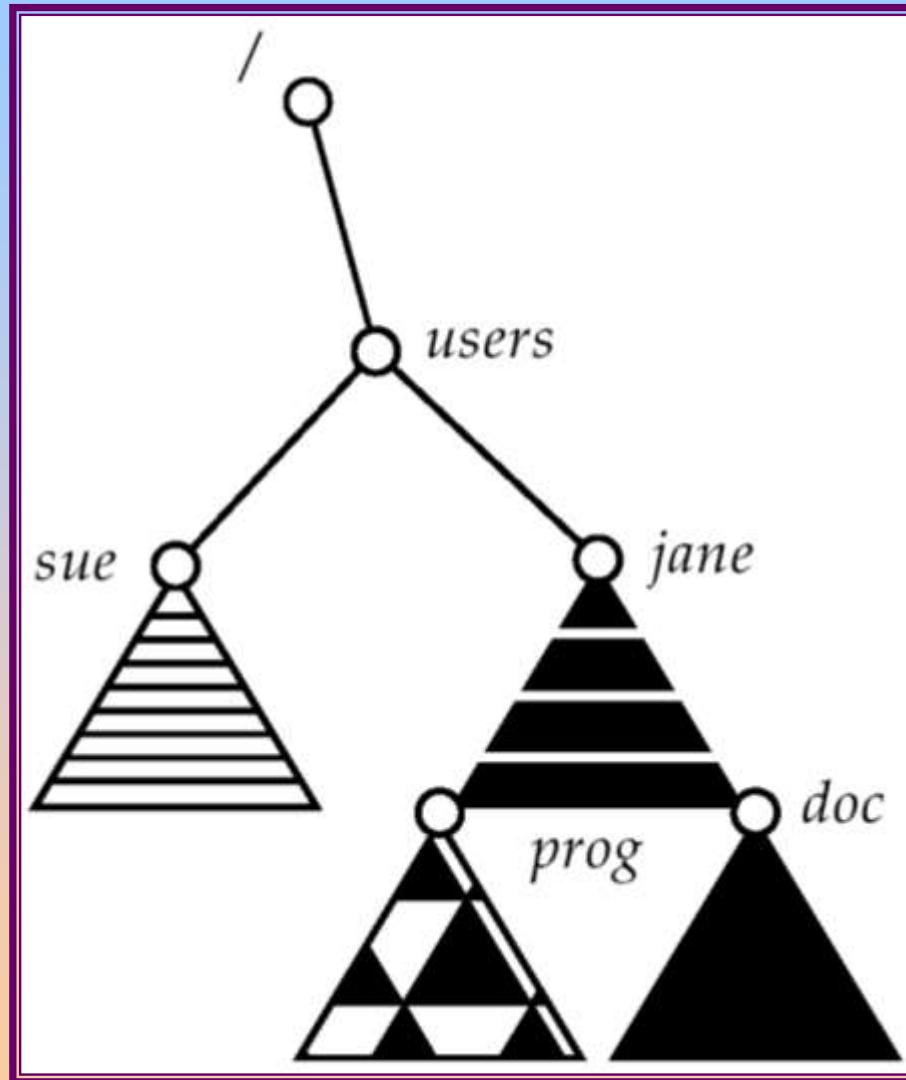
(a) Esistente. (b) Non montato



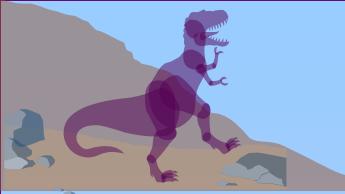
File system: esistente

File system: partizione non montata

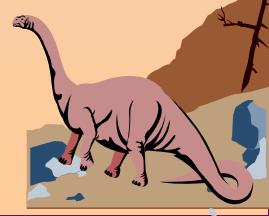
Punto di Montaggio



/users è il *mount point*



Condivisione dei file

- La condivisione di file può essere molto utile in sistemi multiutente:
 - ☞ ottimizza l'uso della memoria e permette maggiore throughput.
 - La condivisione necessita però anche di un opportuno schema di *protezione*.
 - In un sistema distribuito i file possono essere condivisi attraverso la rete.
 - Il ***Network File System*** (NFS) è la più nota implementazione di file system di rete.
- 



Condivisione dei file (II)

■ Utenti multipli

- ☞ Identificazione degli utenti
- ☞ Concetti di *proprietario* e *gruppo*

■ File system remoti

- ☞ Modello client-server
- ☞ Sistemi informativi distribuiti
- ☞ Malfunzionamenti

■ Semantica della coerenza

- ☞ Caratterizzazione del sistema che specifica quando le modifiche apportate ai dati da un utente possono essere osservate dagli altri.

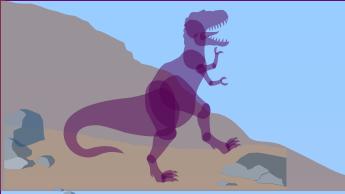
■ Semantica Unix

- ☞ Le scritture di un file aperto sono immediatamente visibili agli altri utenti che hanno contemporaneamente aperto lo stesso file
- ☞ Esiste un metodo di condivisione in cui gli utenti condividono il puntatore alla locazione corrente del file.

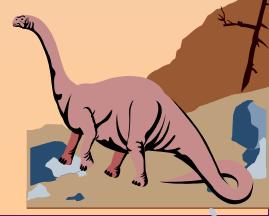
■ Altre semantiche possibili

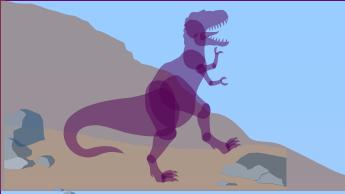
- ☞ Semantica delle sessioni
- ☞ Semantica dei file condivisi immutabili



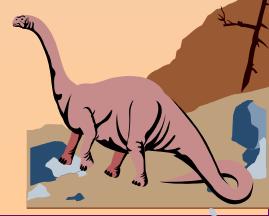


Protezione

- Il proprietario/creatore di un file deve poter controllare:
 - ☞ Le operazioni possibili sui file
 - ☞ A chi permetterne l'esecuzione
 - Se ogni file fosse accessibile a tutti gli utenti, potrebbero verificarsi modifiche o cancellazioni non desiderate.
 - Quindi la necessità di proteggere i file deriva dalla possibilità di accedervi.
 - Tipi di accesso
 - ☞ Lettura
 - ☞ Scrittura
 - ☞ Esecuzione
 - ☞ Aggiunta (append)
 - ☞ Cancellazione
 - ☞ Elencazione
- 



Controllo degli accessi

- Il problema della protezione comunemente si affronta rendendo l'accesso dipendente dall'identità dell'utente.
 - Due dei metodi più comunemente usati sono:
 - ☞ memorizzazione degli elementi (dominio, insieme di diritti) usando una *lista di accesso*
 - ☞ associazione di una *parola chiave* (password) ad ogni file.
 - Lo schema più generale consiste nell'associare un **elenco di controllo degli accessi** a ogni file e directory.
 - In tale elenco sono specificati i nomi degli utenti e relativi tipi di accesso consentiti.
 - Il problema maggiore riguarda la lunghezza degli elenchi.
 - Per condensare la lunghezza dell'elenco alcuni sistemi raggruppano gli utenti di ogni file in tre classi:
 - ☞ **Proprietario**
 - ☞ **Gruppo**
 - ☞ **Universo**
- 

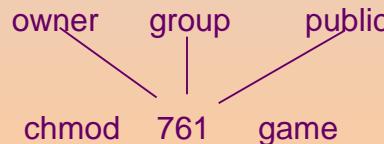


Controllo degli accessi: Unix

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

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

- Si richiede al system manager di creare un gruppo (nome univoco), diciamo G, e aggiungere alcuni utenti al gruppo.
- Per un file particolare (es. *game*) o sotto-directory, definiamo un accesso appropriato.



Definizione del un gruppo di un file
chgrp G game

