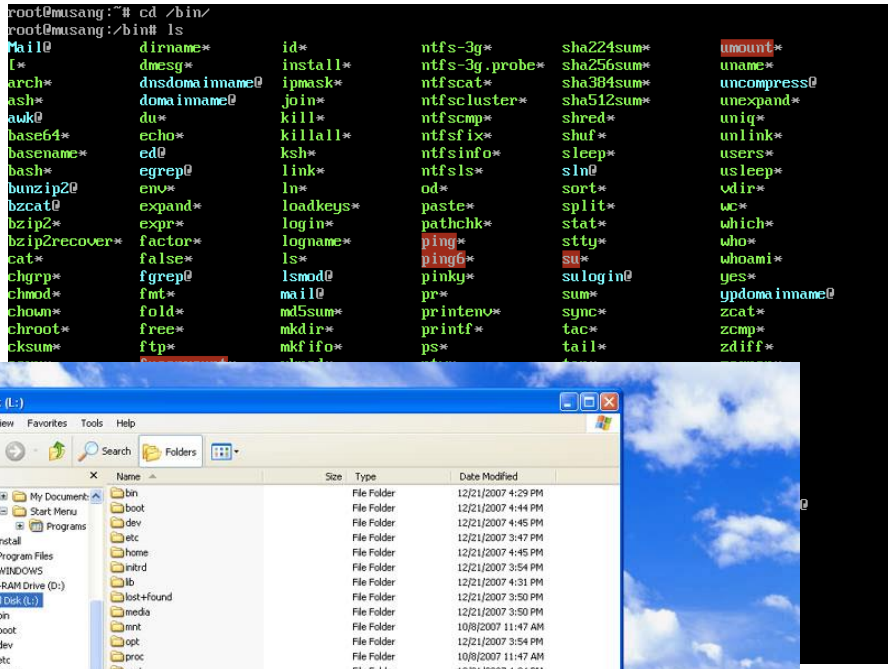
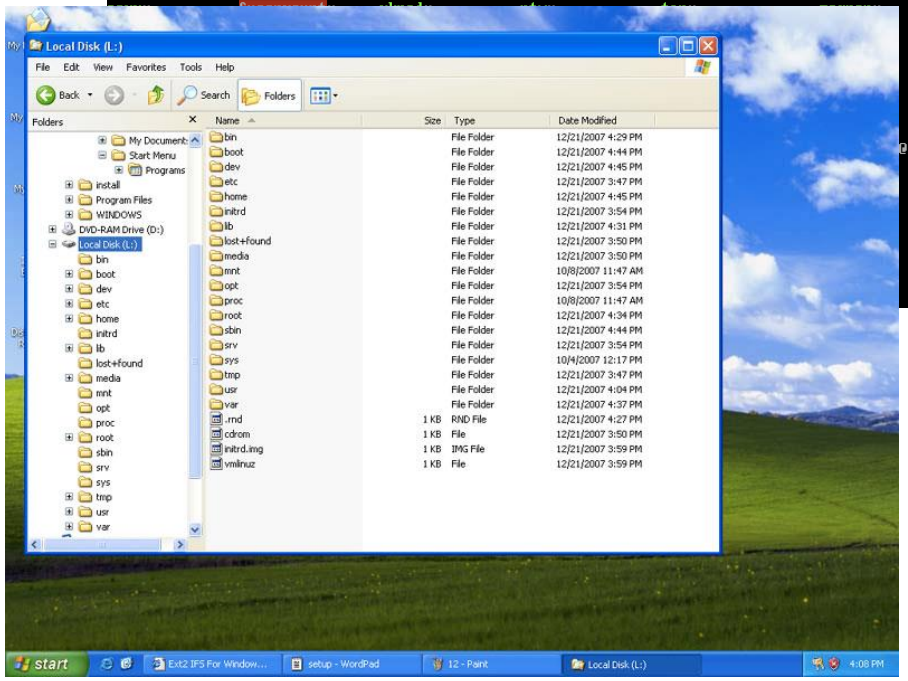
Interfaccia del file system

Interfaccia del file system

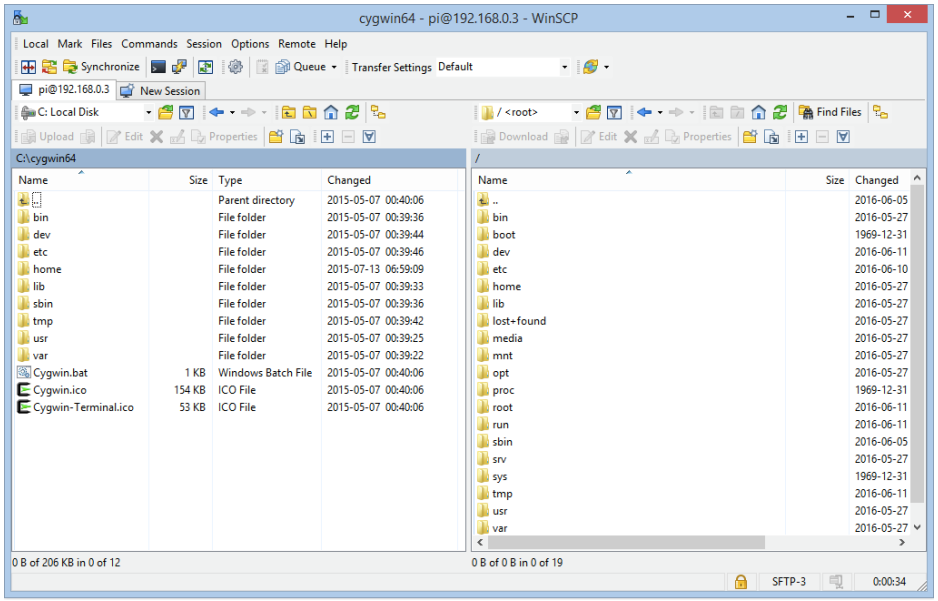
- Concetto di file
- Metodi di accesso
- Struttura delle directory
- Montaggio del file system
- Condivisione di file
- Protezione





Concetto di file

- File: uno spazio di memoria **logicamente contiguo** (e non volatile) a cui è assegnato un nome.
- Un file rappresenta un documento/una scheda/una unità di informazione
- Tipi di file:
  - Dati
    - numerici
    - a caratteri
    - binari
  - Programmi
    - Codice oggetto
    - Codice eseguibile



Struttura dei file

A. Nessuna - sequenza di word o di byte
B. Struttura semplice a record
   - Linee
   - Lunghezza fissata
   - Lunghezza variabile
C. Struttura complessa
   - Documenti formattati
   - Dati rilocabili (eseguibile)

- I file di tipo B e C possono essere simulati inserendo caratteri di controllo.

- Chi decide:
  - Sistema operativo
  - Programma.

Attributi dei file

- **Nome** – nome simbolico (*es: lettera.doc, libro.pdf, 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.
- Una directory realizza il concetto di archivio/cassetto.

Comuni tipi di file: estensioni e funzioni

Tipo di file	Estensione usuale	Funzione
Eseguibile	exe, com, bin, o nessuna	Programma, in linguaggio di macchina, eseguibile
Oggetto	obj, o	Compilato, in linguaggio di macchina, non collegato
Codice sorgente	c, cc, java, pas, asm, a	Codice sorgente in vari linguaggi di programmazione
Batch	bat, sh	Comandi all'interprete dei comandi
Testo	txt, doc	Testi, documenti
Elaboratore di testi	wp, tex, rtf, doc	Vari formati per elaboratori di testi
Libreria	lib, a, so, dll	Librerie di procedure per programmatori
Stampa o visualizzazione	ps, pdf, jpeg	File ASCII o binari in formato per stampa o visione
Archivio	arc, zip, tar	File contenenti più file tra loro correlati, talvolta compressi, per archiviazione o memorizzazione
Multimediali	mpeg, mov, rm, mp3, avi	File binari contenenti informazioni audio o A/V

Operazioni sui file

- Creazione
- Scrittura
- Lettura
- Copia
- Ri-posizionamento nel file
- Cancellazione
- Troncamento

- Apertura

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

- Chiusura

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

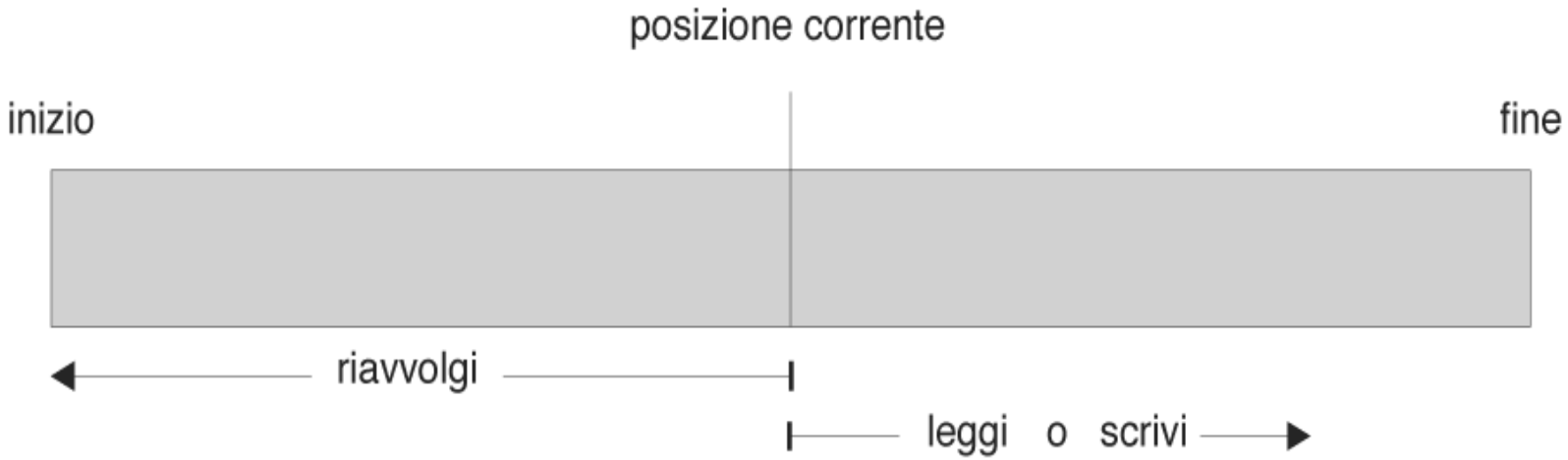
Metodi di accesso

- **Accesso 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

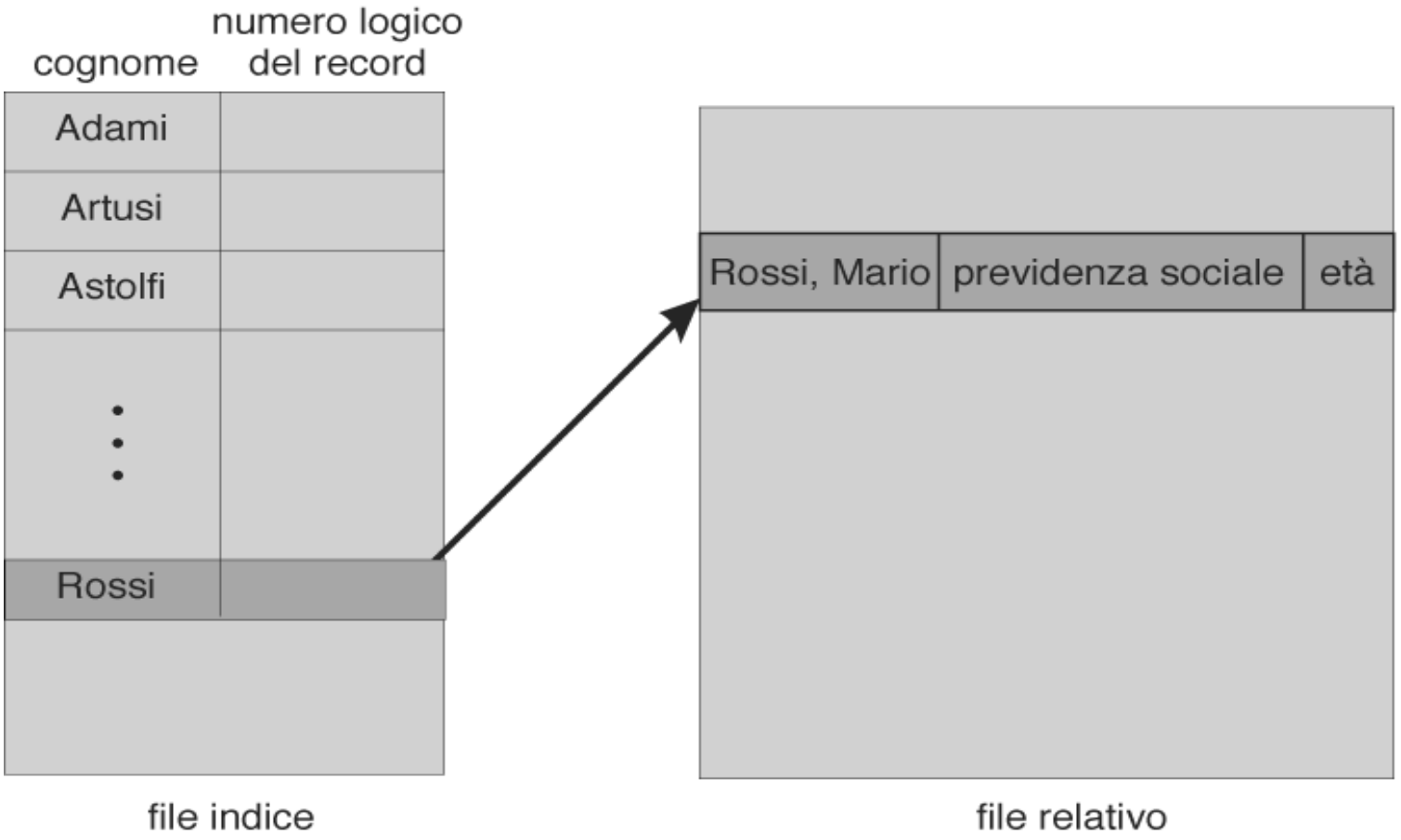


Simulazione dell'accesso sequenziale su un file ad accesso diretto

sequential access	implementation for direct access
reset	*cp* = 0;
read next	*read cp*; *cp* = *cp*+1;
write next	*write cp*; *cp* = *cp*+1;

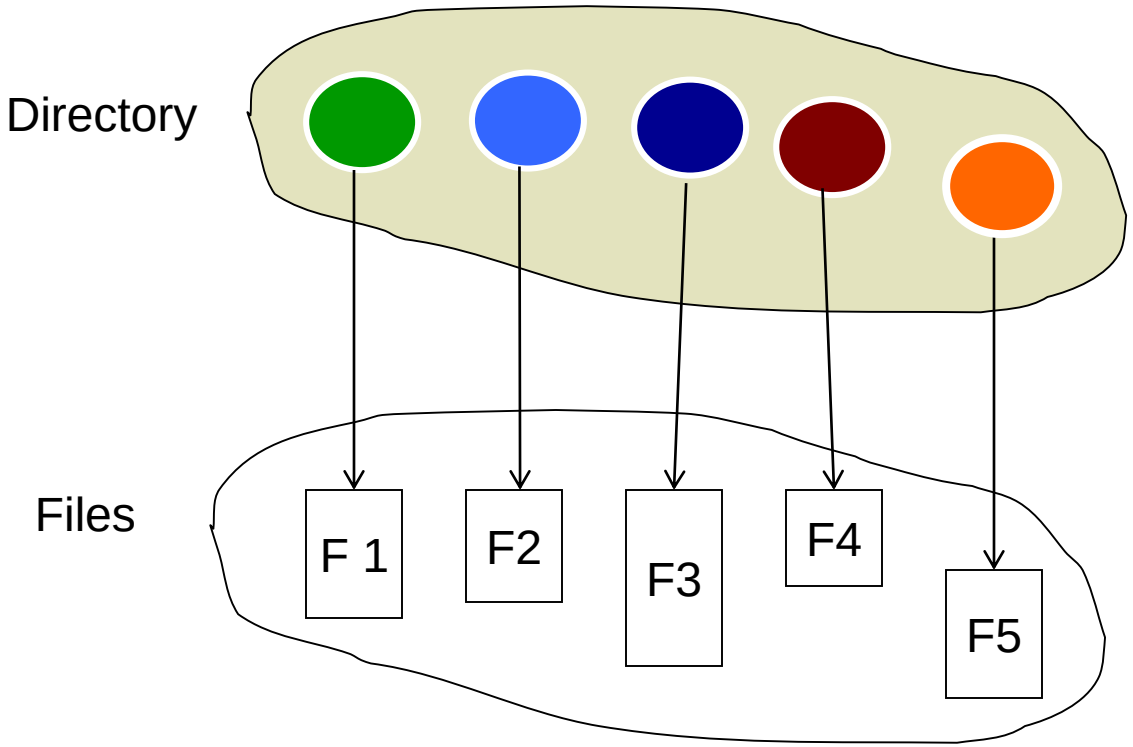
cp : posizione corrente nel file

Esempio di indice e relativo file



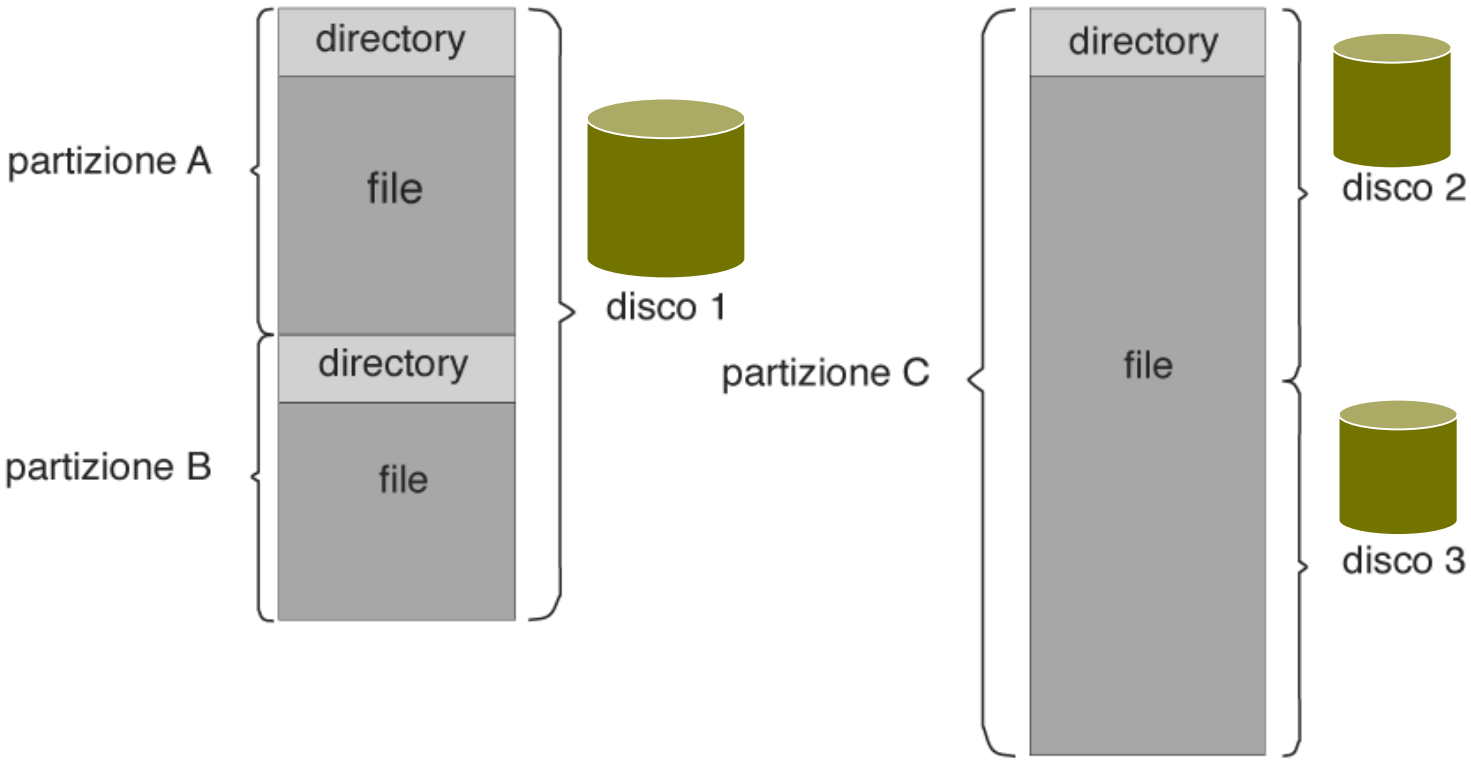
Struttura di directory

Directory: Una collezione di nodi contenenti informazioni su tutti i file.



Sia la struttura di directory, sia i file sono memorizzati su disco.

Tipica organizzazione di un 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
  - ID del proprietario
  - Informazioni di protezione.

Operazioni sulle directory

- Ricerca di un file
- Creazione di un file
- Cancellazione di un file
- Elencare i file nella directory
- Elencare le proprietà dei file nella directory
- Rinominare un file
- Creare una directory
- Eliminare una directory
- Attraversare il file system.

Organizzazione logica di una directory

- **Efficienza**

  Accedere ad un file velocemente.

- **Identificazione**

  Conveniente per gli utenti.

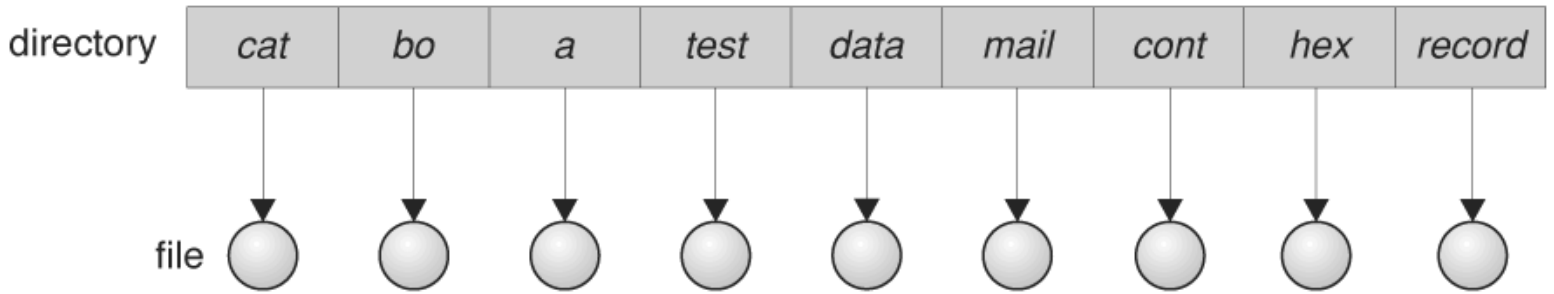
  - Due utenti possono usare nomi uguali per file diversi.
  - Lo stesso file può avere nomi diversi.

- **Raggruppare**

  Classificare insieme i file in base alle loro proprietà, (es., tutti i programmi Java, tutti i videogiochi, ecc.)

Directory a livello singolo

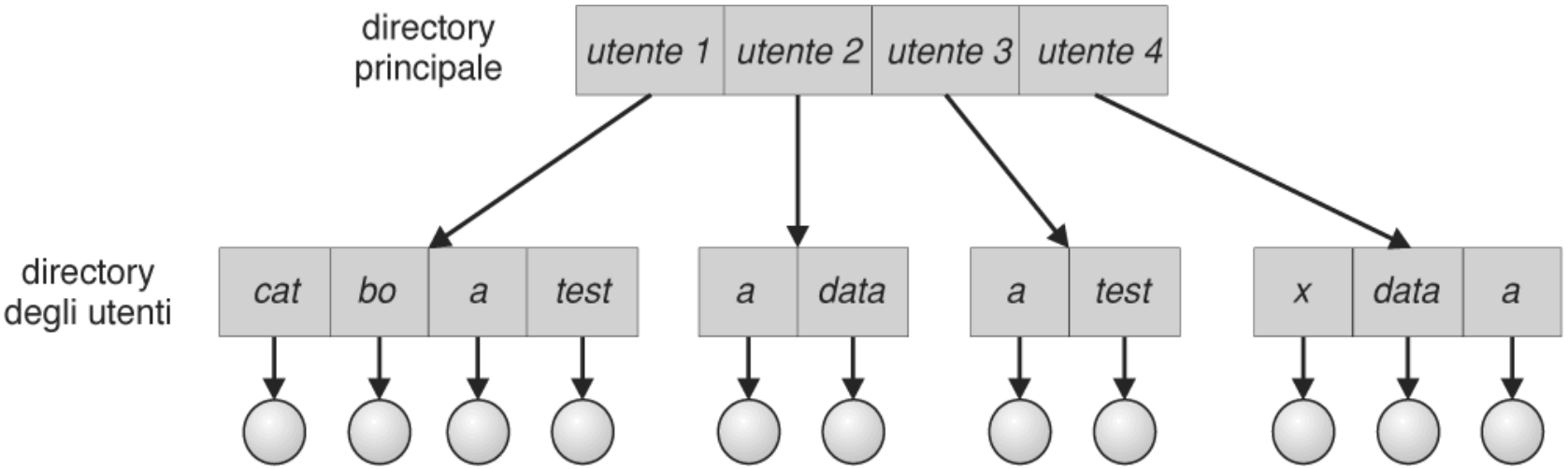
- Una singola directory per tutti i file.



- ✓ Problema di identificazione.
- ✓ Problema di raggruppamento.

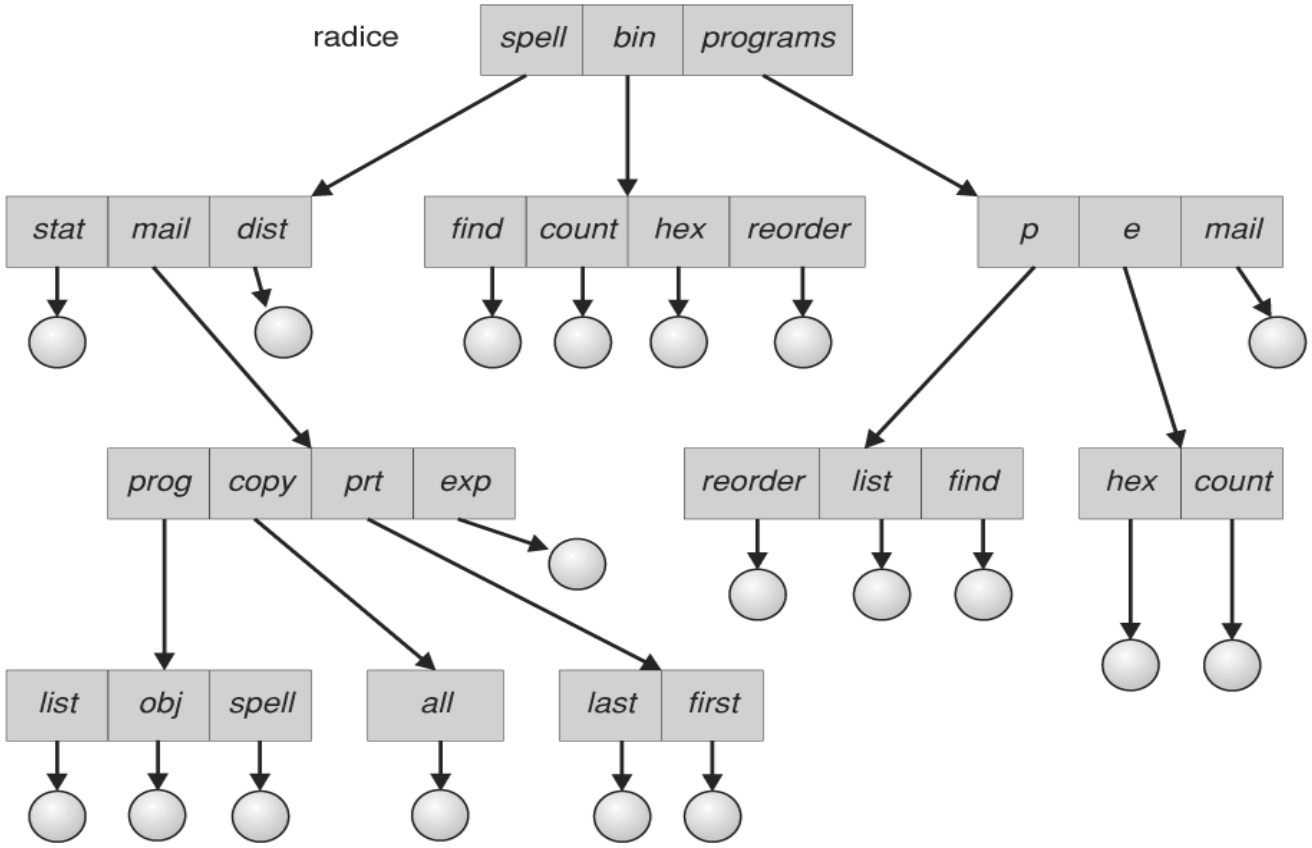
Directory a due livelli

- 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*)
- Path name **assoluto** o **relativo**
  - **cd** /spell/mail/prog — *spostamento assoluto*
  - **type** list — *file locale*

Directory con struttura ad albero

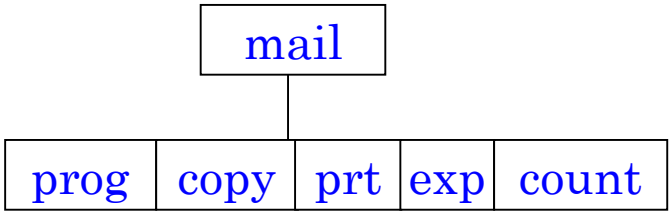
- La creazione di un nuovo file avviene nella directory corrente.
- Cancellare un file

  > **rm** <file-name>

- La creazione di una nuova directory avviene nella directory corrente.

  > **mkdir** <dir-name>

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

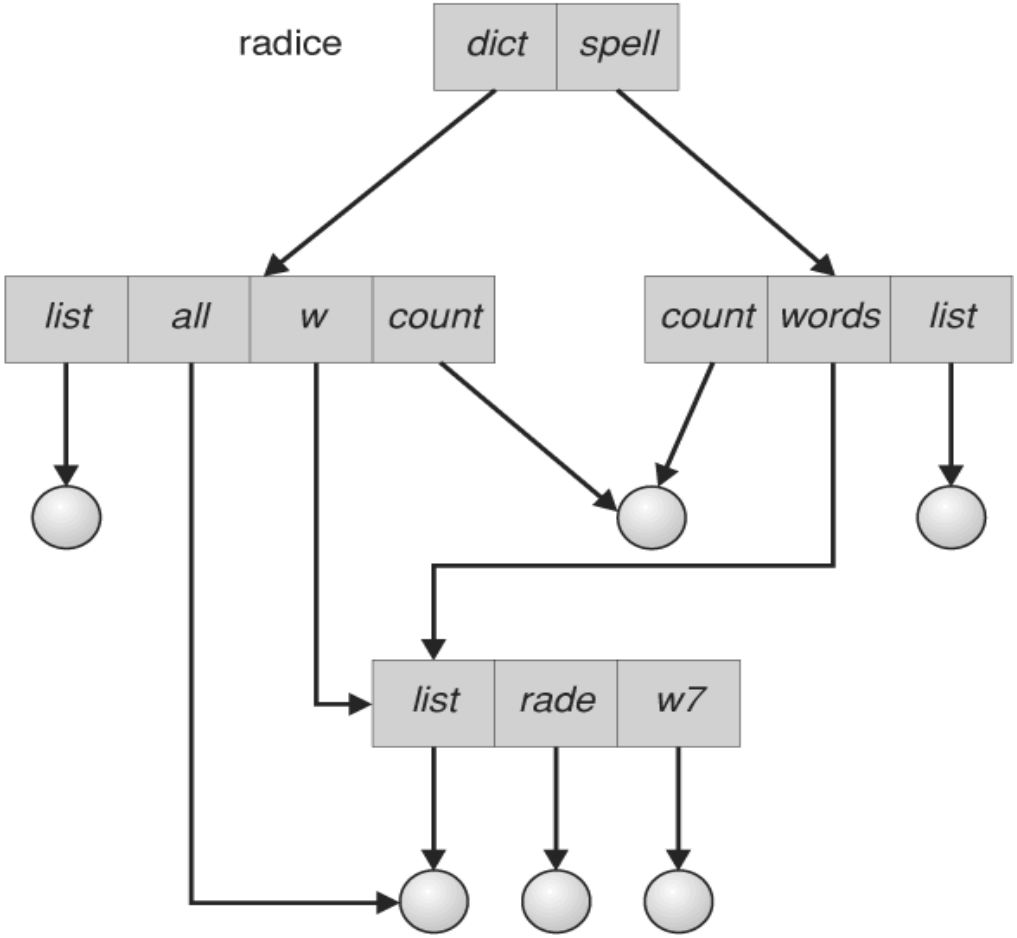


> **mkdir** count

Cancellazione di “mail” ⇒ Cancellazione di tutto il sottoalbero “mail”. Di solito richiede di cancellare prima il contenuto della directory.

Directory a grafo aciclico

- Ha sotto-directory e file condivisi.

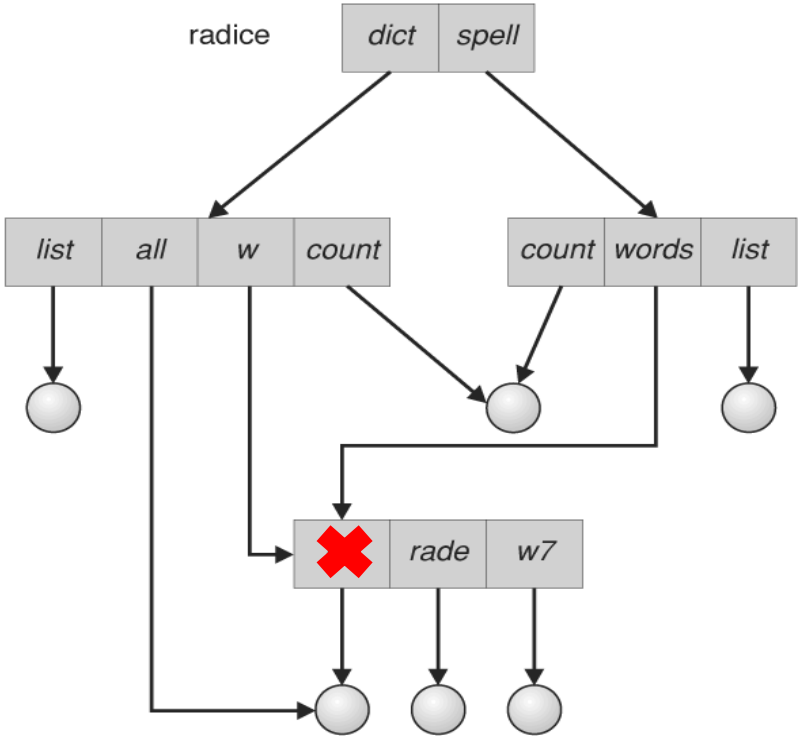


Directory a grafo aciclico

- Due nomi differenti per uno stesso file (*aliasing*)
- Se in ***dict*** si cancella ***list*** $\Rightarrow$ puntatore “appeso”.

Soluzioni:

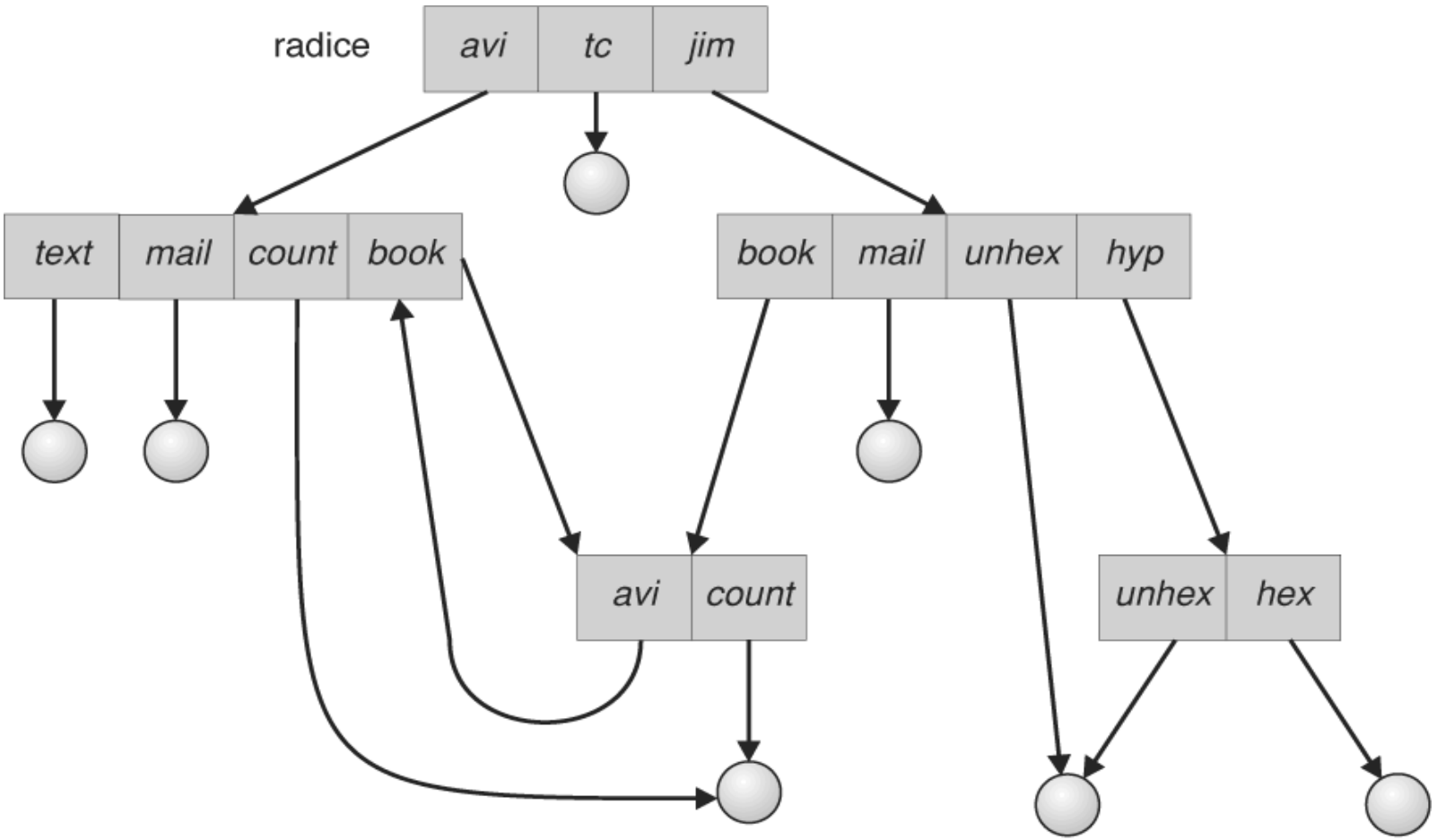
- Puntatori all’indietro, per cancellare tutti i puntatori all’elemento eliminato.
- Problema della dimensione (quanti puntatori ?).
- Soluzione basata su contatori.



Directory a grafo generale

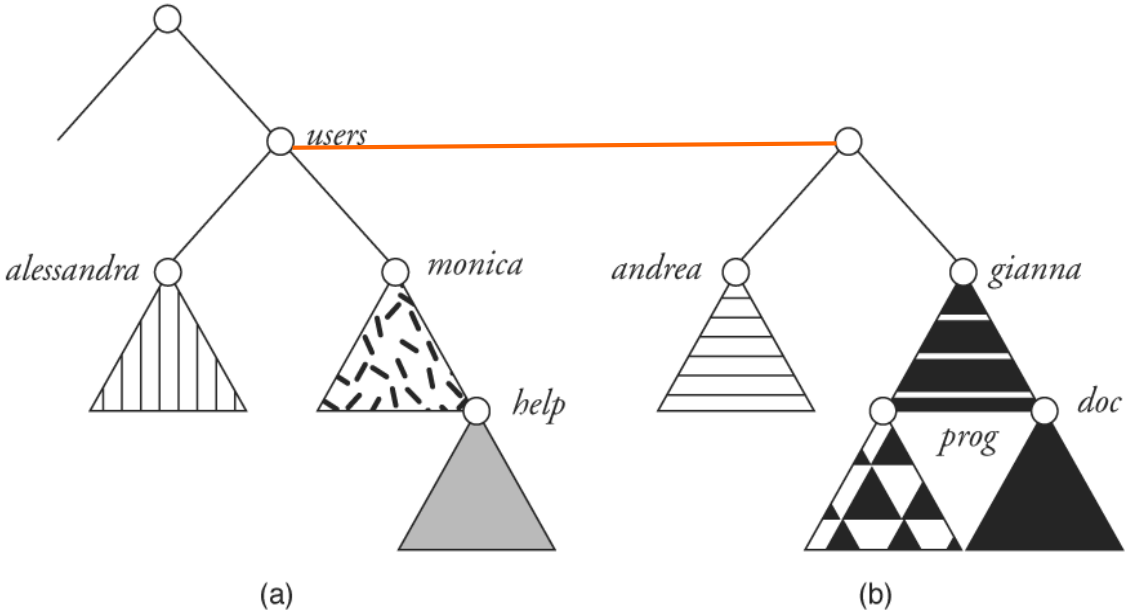
- Come garantire l’assenza di cicli?
  - Permettere link a file non a sotto-directory.
  - Garbage collection (costoso).
  - Ogni volta che viene creato un file si può usare un algoritmo per la rilevazione di cicli che li eviti. (costoso)

Directory a grafo generale



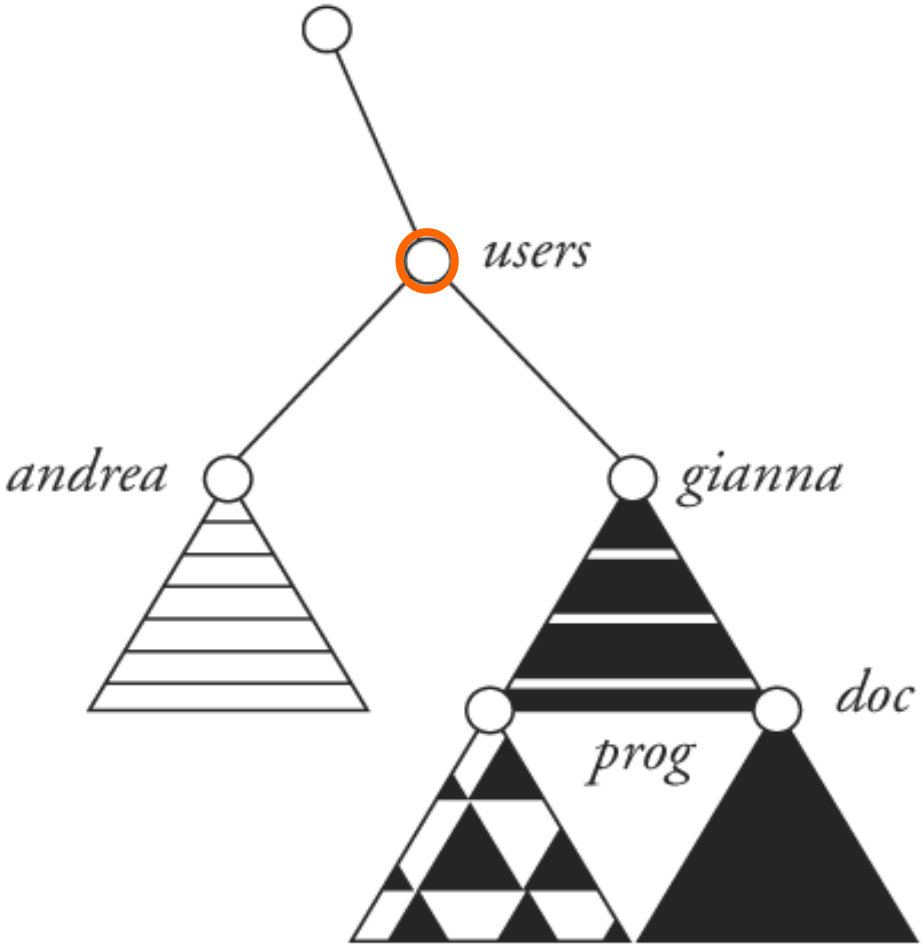
Montaggio di un file system

- 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***).



File System esistente Partizione non montata

Punto di montaggio



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.
- Tramite NFS si può configurare un file system composto con file presenti su un insieme di computer connessi in rete.

Protezione

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

- Tipi di 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

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

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