

**MA ALLA FINE...**

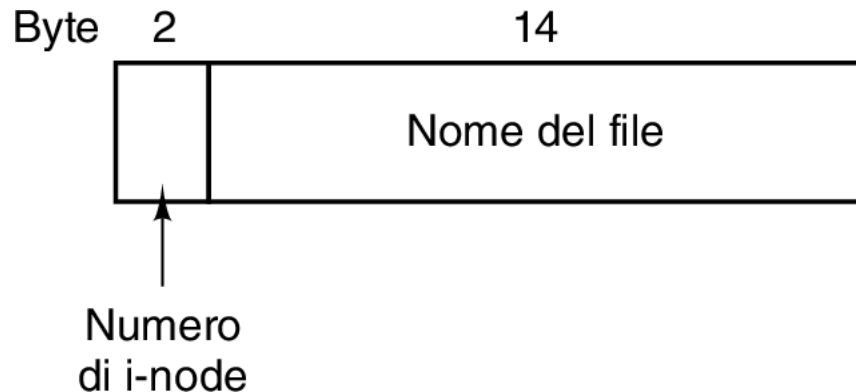


**QUALE FILE SYSTEM E' USATO?**

**E COME FACCIO A «MONTARE» UNA  
PARTIZIONE?**

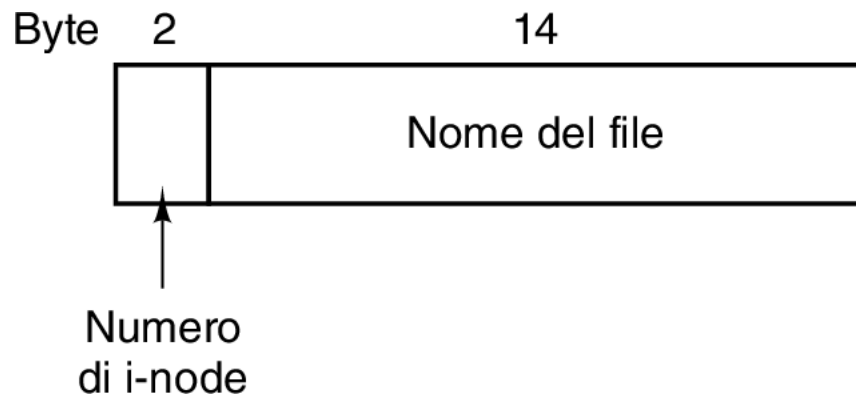
# INTRODUZIONE AL FILE SYSTEM V7 DI UNIX (1979!!!)

- **Origine e Influenza:** Derivato dal MULTICS, il file system V7 è stato implementato nel PDP-11 e ha contribuito significativamente alla fama di UNIX.
- **Struttura:** Formato da un albero gerarchico con la possibilità di formare un grafo aciclico orientato tramite link.
- **Nomi dei File:** Lunghezza massima di 14 caratteri, escludendo i caratteri '/' e NUL.
- **Limitazione Numerica:** Massimo di 64K file per file system, a causa del formato delle voci di directory.



# VOCI DI DIRECTORY E I-NODE IN UNIX V7

- **Voce di Directory:** Composta da un nome di file (14 byte) e un numero di i-node (2 byte).
- **i-node:** Contiene attributi del file, inclusi dimensioni, timestamp, proprietario, gruppo, e informazioni di protezione.
- **Gestione dei Link:** Contatore di link nell'i-node, che viene incrementato o decrementato con la creazione o rimozione di link.



# ALGORITMO DI RICERCA NEL FILE SYSTEM UNIX V7

- **Ricerca di File:** Processo per localizzare file tramite un percorso, partendo dalla directory radice o dalla directory corrente.
- **Esempio di Ricerca:** Ricerca di `/usr/ast/mbox` attraverso sequenze di letture di directory e individuazione di i-node.
- **Percorsi Relativi:** Gestiti allo stesso modo dei percorsi assoluti, iniziando dalla directory di lavoro corrente.
- **Gestione delle Directory Speciali:** Uso di `.` e `..` per indicare rispettivamente la directory corrente e la genitore.
- **Impatto su Sistemi Successivi:** Il design e le caratteristiche del file system V7 hanno influenzato lo sviluppo di file system UNIX successivi, tra cui quelli utilizzati in sistemi Linux moderni.



# EVOLUZIONE DEI FILE SYSTEM IN LINUX - DA EXT A EXT2

- **Ext (1992): Il Primo File System di Linux**

- Creato specificatamente per il kernel di Linux da Rémy Card.
- Superava i limiti del file system MINIX, con una capacità massima di 2 GB.
- Primo ad utilizzare il Virtual File System (VFS) nel kernel Linux.

- **Transizione a Ext2 (1993)**

- Ext2 introdotto per risolvere problemi di Ext, come la immutabilità degli i-node e la frammentazione.
  - **Immutabilità:** una volta creati, gli attributi principali di un i-node (come il suo numero identificativo) non cambiano per tutta la durata della vita del file a cui sono associati.
- Competizione con Xiafs, ma Ext2 prevale per la sua maggiore affidabilità a lungo termine.



# DALL'EXT3 ALL'EXT4 - INNOVAZIONE E STABILITÀ

- **Ext3: L'Introduzione del Journaling**

- Ext3 è stato sviluppato come successore di Ext2, aggiungendo il journaling per una maggiore integrità dei dati.

- **Nascita e Sviluppo di Ext4 (2006-2008)**

- Inizialmente estensioni retrocompatibili di Ext3, sviluppate tra il 2003 e il 2006.
- Scelta di biforcare Ext3 in Ext4 per evitare impatti sulla stabilità di Ext3.
- Ext4 marcato come stabile nel 2008, con rilascio nel kernel 2.6.28.

- **Adozione di Ext4 da Google**

- Google passa da Ext2 a Ext4 per il suo storage nel 2010.
- Android 2.3 adotta Ext4 al posto di YAFFS nel 2010.





# FILE SYSTEM EXT4 DI LINUX

- **Journaling e Affidabilità:**

- Introduce il journaling per prevenire la perdita di dati in caso di crash.
- Utilizza un Journaling Block Device (JBD) per le operazioni di log.

- **Miglioramenti rispetto a Ext2 e Ext3:**

- Aumento della dimensione massima dei file e dei file system.
- Introduzione degli "extent" per una gestione efficiente dei blocchi di memoria contigui.
  - Gli "extent" in ext4 sono strutture che indicano un intervallo contiguo di blocchi su disco, specificando l'indirizzo di inizio e la quantità di blocchi consecutivi.
  - Questo approccio semplifica la gestione di blocchi contigui, riducendo il numero di voci necessarie per descrivere la locazione dei dati su disco, specialmente per file di grandi dimensioni.
- Compatibile con ext2 ed ext3, ma con prestazioni e capacità superiori.

- **Configurazione e Opzioni:**

- Possibilità di configurare il journaling solo per i metadati o per l'intero disco.
- Supporto a file di dimensioni fino a 16TB e file system fino a 1EB.



# UNO SGUARDO AL FUTURO: BTRFS

- **Panoramica su Btrfs - Il File System Avanzato**

- **Btrfs: (pronuncia: «better F S»)**

- *Copy-on-Write*: Quando un file è duplicato, Btrfs condivide il file originale invece di crearne una copia, riducendo lo spazio occupato.
- *Prevenzione della Perdita di Dati*: quando i dati vengono modificati, Btrfs non sovrascrive i dati esistenti, ma invece scrive le modifiche in una nuova posizione sul disco

- **Caratteristiche Salienti di Btrfs**

- **Supporto per File Enormi**: Gestisce file fino a 16 exbibyte (18446.7 petabyte).
- **Archiviazione Efficiente**: Riduce il sovraccarico nei metadati dei file, ottimizzando così la gestione dello spazio e delle prestazioni.
- **Supporto RAID**: Compatibile con RAID 0, 1 e 1+0 per striping e mirroring dei dati.
- **Deframmentazione e Ridimensionamento Facili**: Operazioni eseguibili mentre il filesystem è attivo.
- **Allocazione Dinamica degli Inode**: Evita l'esaurimento degli inode, salvandoli per un gran numero di piccoli file.
- **Supporto per Snapshot**: Permette la creazione e il ripristino facili degli snapshot (backup) del filesystem.
- **Supporto Checksum**: Riduce il rischio di corruzione dei dati attraverso blocchi di dati verificati costantemente.
- **Ottimizzazione per SSD**: Migliora le prestazioni degli SSD, estendendone la durata.





# CONFRONTO TRA BTRFS E EXT3/EXT4

- **Btrfs: File System Avanzato**

- Progettato per l'era dei moderni dispositivi di archiviazione.
- Supporta snapshot e rollbacks, ottimale per backup e ripristini.
- Gestione nativa del RAID e miglioramento nell'integrità dei dati con checksum.
- Compressione dei dati e deduplicazione per un utilizzo efficiente dello spazio.

- **Ext3/Ext4: Affidabilità e Stabilità**

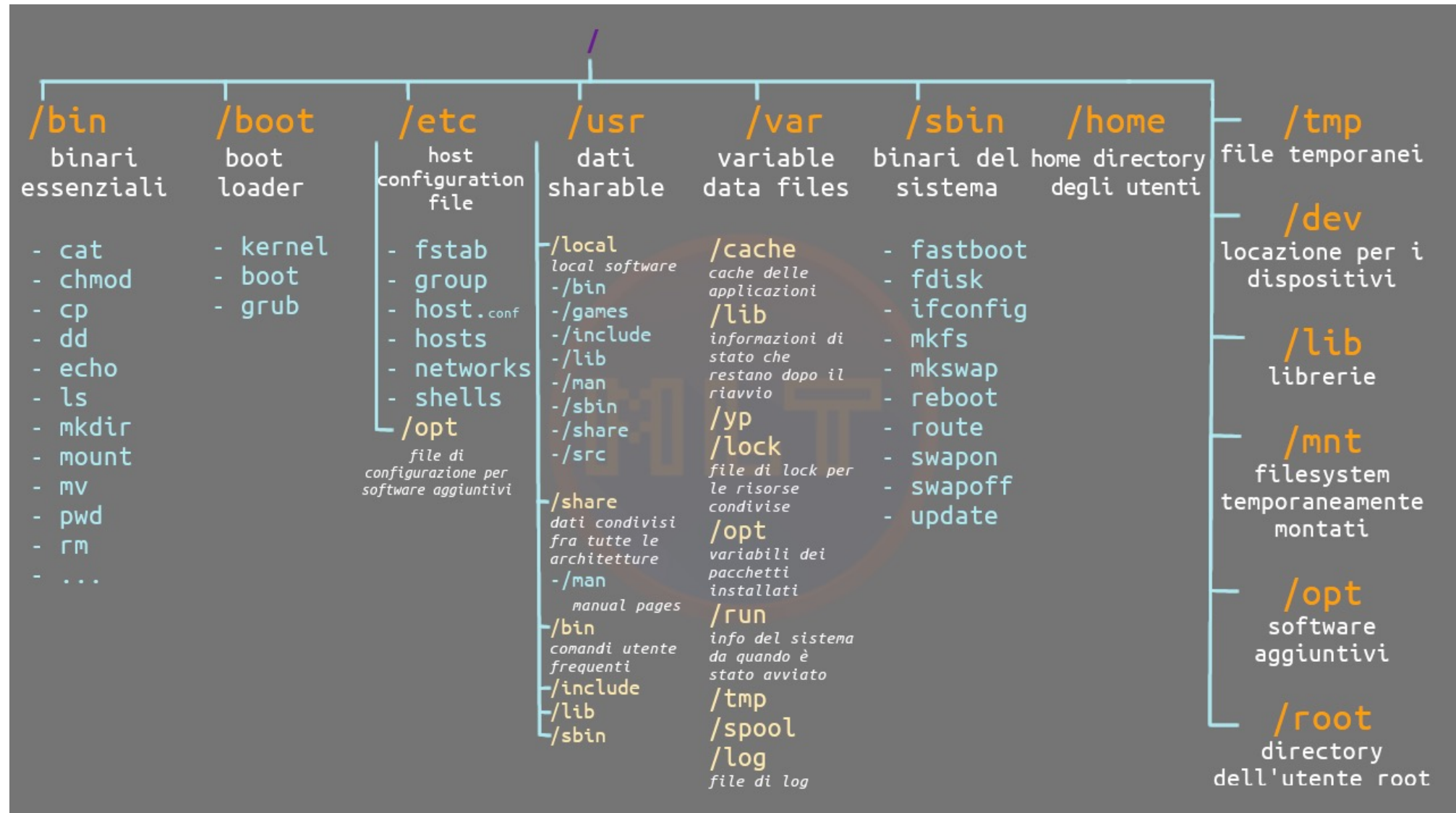
- Ext3: Affidabile con supporto journaling, ma limitato in termini di funzionalità avanzate.
- Ext4: Miglioramenti nell'efficienza, supporto per file di grandi dimensioni, riduzione della frammentazione.

- **Vantaggi e Svantaggi**

- Btrfs offre funzionalità avanzate ma è meno maturo e testato rispetto a Ext4.
- Ext3/Ext4 è noto per la sua stabilità e prestazioni, ma manca di alcune delle caratteristiche moderne di Btrfs.



# STRUTTURA DELLE CARTELLE IN LINUX



# STRUTTURA DELLE CARTELLE IN LINUX (2)

- **/bin:** contiene i binari dei principali comandi eseguibili dal sistema: `cat`, `ls`, `pwd`, ecc...
- **/boot:** contiene tutti i file necessari al Boot Loader per il processo di avvio del sistema. Inoltre, contiene il Kernel.
- **/etc:** contiene tutti i file di configurazione del sistema e di controllo
- **/usr:** contiene i pacchetti del sistema e le directory e le applicazioni dell'utente
- **/var:** contiene file temporanei, log, ecc...
- **/sbin:** ci sono i binari del sistema, quindi anche comandi come `ifconfig`, `mkfs`, `fdisk`, ecc...
- **/dev:** i file descrittori dei device, ovvero dei dispositivi/periferiche come gli hard disk interni. In linux anche i dispositivi vengono visti come file!
- **/home:** cartella principale dell'utente dove vengono salvati i file locali, contenente le cartelle: Documenti, Immagini, ecc...



# STRUTTURA DELLE CARTELLE IN LINUX (3)

- **/lib:** contiene le librerie essenziali, incluso il compilatore C e le relative librerie.
- **/media:** contiene cartelle che servono per gestire media rimovibili «montati» automaticamente dal sistema, come cd, floppy, ecc...
- **/mnt:** usata per eseguire il mount manuale dei filesystem temporanei dei dispositivi rimovibili come USB e cd.
  - Infatti, una volta attaccata una chiavetta usb, se non “montassimo” il suo filesystem per poterne leggere il contenuto, non saremmo in grado di visualizzarne le informazioni.
  - Solitamente c'è una cartella per ogni dispositivo montato. (Vedi dopo)
- **/opt:** abbreviazione di “Optional”, contiene sia gli add-ons di alcuni software, sia programmi che non sono necessari al sistema.
  - Molte persone usano questa cartella per creare sotto-directory in cui compilare e installare programmi.
- **/root:** home dell'utente root.



# VISUALIZZAZIONE DELLE PARTIZIONI E FILE SYSTEM CON MOUNT E ALTRI COMANDI

- **Scoprire Partizioni e File System Disponibili:**

- **Comando** `lsblk`

- Mostra un elenco dei dispositivi di blocco, inclusi dischi e partizioni.
    - Esempio: `lsblk` per visualizzare tutte le partizioni e i dispositivi.

- **Comando** `fdisk -l`

- Elenca dettagliatamente tutte le partizioni sui dischi, comprese quelle non montate.
    - Richiede privilegi di root: `sudo fdisk -l`

- **Visualizzazione dei File System Montati:**

- **Comando** `mount -l`

- Elenco dei file system attualmente montati con le loro opzioni di montaggio e etichette.
    - Utile per vedere rapidamente dove e come sono montate le partizioni e i file system.

- **Importanza di Conoscere le Partizioni:**

- Fondamentale per gestire correttamente lo spazio su disco e l'organizzazione dei dati.
  - Cruciale per operazioni di backup, ripristino e manutenzione del sistema.



# MONTAGGIO DI PARTIZIONI E FILE SYSTEM CON IL COMANDO MOUNT

- **Montaggio di una Partizione/File System:**

- Sintassi Base:

- `mount [opzioni] <dispositivo> <directory>`.

- **Esempio:** `sudo mount /dev/sda1 /mnt/mydisk` per montare /dev/sda1 in /mnt/mydisk.

- Creazione della Directory di Montaggio:

- La directory di destinazione deve esistere prima del montaggio (Esempio: `mkdir /mnt/mydisk`).

- **Opzioni di Montaggio Comuni:**

- Specifica del Tipo di File System: `-t <tipo>`, es. `-t ext4`.

- Opzioni Aggiuntive: `-o <opzioni>`, es. `-o ro` per montaggio in sola lettura.

- **Smontaggio di un File System:**

- Comando `umount`:

- Utilizzato per smontare in modo sicuro un file system o una partizione.

- **Esempio:** `sudo umount /mnt/mydisk`.

- **Note Finali:**

- Queste operazioni richiedono privilegi di root.

- Smontaggio corretto è essenziale per prevenire la perdita di dati.

