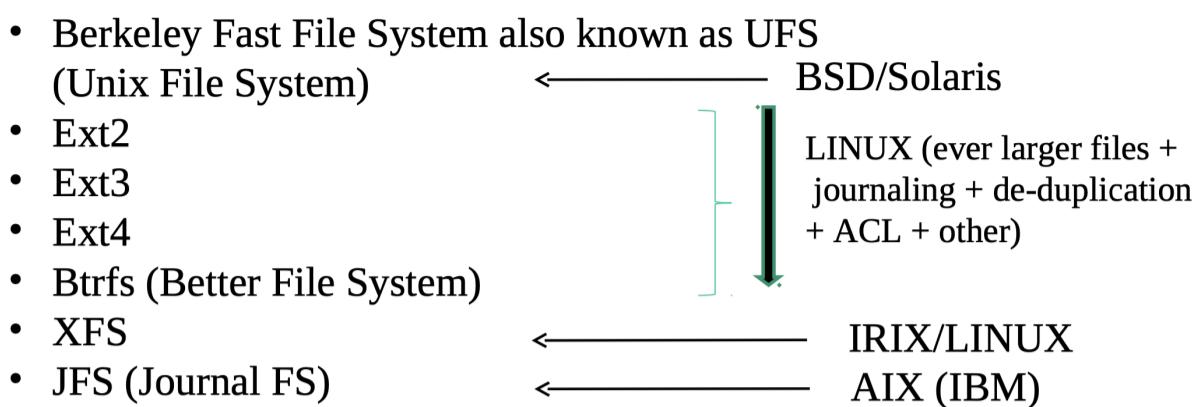


# UNIX file systems



## Caratteristiche di base dei file system UNIX

Quando noi abbiamo un file system UNIX, i file che vengono mostrati alle applicazioni come sequenze di records, in realtà sono tali per cui ciascun record quanto è grande? È un unico byte.

Un file è una sequenza di singoli byte. Se vogliamo registrare all'interno di un file informazioni più ampie di un singolo byte, utilizziamo una zona fatta da più record, ovviamente.

I file system unix ci forniscono un metodo di accesso diretto: SU QUESTA SEQUENZA DI BYTE NOI POSSIAMO POSIZIONARCI O RIPOSIZIONARCI IN UN QUALSIASI PUNTO.



Size of ciascun record: 1 byte

**Gli attributi per la gestione dei singoli file ossia il RECORD DI SISTEMA, viene denominato I-NODE.**

**Nel metodo di allocazione, i file utilizzano lo schema di allocazione indicizzata, quindi è un nodo di indice.**

**I vari dati associati ad uno specifico file, per sapere dove i blocchi di questo file sono collocati nell'hard drive, il record di sistema mi mantiene direttamente gli indici.**

**L'indicizzazione è a livelli multipli!!! Abbiamo alcuni indici diretti per sapere dove una parte di un file è posizionata, poi se il file è più grande e abbiamo bisogno di andare avanti e raggiungere una parte finale che non è direttamente indicizzata, utilizziamo indici indiretti. Anche i valori di I-NODE devono essere riportati sull'hard drive dove i blocchi del file sono presenti, non posso perderli. Alla fine sono dati che vanno ad indicare alcune informazioni strutturali sul file system, quindi non possono essere mantenuti soltanto in maniera volatile.**

**Se noi lavoriamo su un hard drive dove abbiamo più partizioni - suddiviso in tanti contenitori di informazioni separati - la partizione dove manteniamo il file system unix è fatta in questo modo qua: (residente in una zona della memoria di massa)**



**Nei blocchi manteniamo i record del file - ci sarà una zona di blocchi occupati e una zona libera - e così via. L'altra zona è composta dai metadati, tutto l'I-NODES, il vettore avrà degli elementi liberi e non (si manterranno i record di sistema dei file correntemente allocati all'interno del file system).**

**Quando creiamo un nuovo file, e poi vedremo una system call che fa questo, dobbiamo controllare il vettore i-nodes e vedere una zona libera che ci permette di inserire i dati associati a quello specifico file.**

Mantenere tutti i record di sistema in quella zona non significa dire che poi non vengano caricati in memoria di lavoro per eseguire le operazioni. Quindi vengono comunque portate in CACHE I-NODES per lavorarci sopra, come per esempio l'altra zona viene portata nel buffer cache.

Nel momento in cui abbiamo un i-nodes per rappresentare un certo file, sappiamo che esso è in posizione X nella sequenza, sappiamo qual è l'indice di quell'I-NODE.

Sapere all'interno di una directory, qual è l'I-NODE che dobbiamo utilizzare per un file mantenuto all'interno della directory, è facile: in questa directory andiamo ad indicare che un certo file F sta utilizzando l'I-NODE con indice numero due.

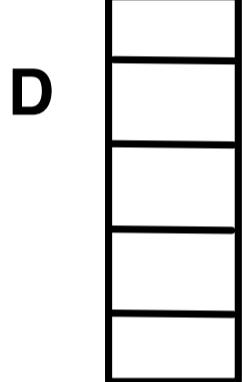
I permessi d'accesso sono lì, le informazioni che ci dicono dove sono i dati del file SONO PROPRIO QUI, all'interno di questo I-NODE.

- Organizzazione gerarchica dell'archivio
  - 1) la directory associa ad ogni file il numero di i-node corrispondente
  - 2) struttura di directory-entry: numero di i-node – spiazzamento per la entry – successiva – lunghezza del nome – nome

La struttura di directory ha 4 byte almeno per indicare qual è l'i-node dell'elemento contenuto all'interno della directory.

Poi abbiamo X byte per il nome del file, poi la lunghezza del nome, e informazioni del tipo che ci dicono qual è la entry successiva all'interno della struttura di directory.

In ciascun elemento io vado a dire dov'è posizionato l'elemento successivo, e gli elementi di una directory hanno una taglia differente.



Questo directory è un file ma poi questi dati possono essere spartiti su più record diversi chiaramente, ma i record spartiti su più blocchi del dispositivo di memoria di massa.

Il primo elemento del file System Unix è una bit-Map che ci dice quali sono i blocchi liberi dove noi possiamo scrivere i dati, poi possiamo usare questi blocchi per ampliare un file esistente o per registrare nuovi file, e poi aggiornare il valore di quei bit. Ma quei bit ci vanno ad indicare quali sono gli I-NODES liberi all'interno di questo array.

Quand'è che un file-System è saturo?

il file system è rappresentato su dispositivo tramite questo unico blocco di informazioni. Ma quand'è che è pieno?

O quando la zona dei blocchi è piena, ma anche quando la zona del vettore I-NODES non è piena. Possiamo registrare altri I-NODES senza poter aggiungere dati di nuovi file.

Oppure nella sezione dei blocchi abbiamo anche spazio, ma nel vettore i-nodes non abbiamo più spazio per ospitare gli attributi dei file. Quindi il numero dei file massimo che potevamo gestire lo abbiamo già raggiunto.

