

File system

Il file system è l'astrazione che permette l'interazione dei programmi con i dispositivi di storage. Fornisce:

naming memorizzazione dei dati con nomi significativi e strutture gerarchiche;

condivisione mantiene metadati con proprietari e permessi;

affidabilità gestire corruzione dei dati, perdite di corrente, usura...

miglioramento delle prestazioni riduzione del costo di operazioni dispendiose (scrittura, seek, erasing...) con caching e disposizione appropriata dei dati sul dispositivo;

I dati sono organizzati in una gerarchia di file e directory:

file insieme di dati, presentati come array di byte ma non necessariamente contigui sul disco, con nome, diritti, modalità di accesso (e.g. sequenziale o casuale) e altri metadati;

directory indici di file e directory, consentono la traduzione da path a identificatore di file. Spesso implementate come file speciali;

link *hard* puntano ai metadati, *simbolici/soft* a un nome. Linux non permette hard link a directory per mantenere la struttura di DAG del file system.

API: create/unlink, link, mkdir/rmdir, open/close, read/write, seek, mmap/munmap, fsync. Spesso sono usate tramite funzioni di libreria che fanno buffering per ridurre il numero di syscall.

Il file system deve fornire:

- un indice per determinare i blocchi (gruppi di settori, per ridurre l'overhead rispetto a gestirli singolarmente) che compongono i file;
- una mappa dello spazio libero;
- strategie per migliorare la località.

Application	FAT	FFS	NTFS	ZFS
Library	linked	tree (fixed, list)	tree (dynamic)	tree (COW, dynamic)
File System	Index structure	assymmetric		
Block Cache	Index structure granularity	block	extent	block
Block Device Interface	Free space management	FAT array	bitmap in file (file)	space map (log- structured)
Device Driver	Locality heuristics	bitmap (fixed)		
Memory-Mapped I/O, DMA, Interrupts		defrag. block groups	best fit	write-anywhere
Physical Device		reserve space	defrag.	block groups