

Gestione dei file su Windows

Index

- [Introduction](#)
- [Caratteristiche principali di FAT](#)
- [File Allocation Table](#)
 - [Funzionamento](#)
 - [Struttura](#)
 - [Boot sector](#)
 - [FAT](#)
 - [Regione Root Directory](#)
 - [Regione dati](#)
 - [Limitazioni](#)
- [NTFS](#)
 - [Formato del volume](#)
 - [Regione boot sector](#)
- [MTF](#)
 - [Record](#)
 - [Files](#)
 - [Record base](#)
 - [Record con estensioni](#)

Introduction

In Windows esistono due tipi di file-system:

- FAT (vecchio, da MS-DOS) → allocazione concatenata, con blocchi (*cluster*) di dimensione fissa
 - NTFS (nuovo) → allocazione con **bitmap**, con blocchi (*cluster*) di dimensione fissa
-

Caratteristiche principali di FAT

FAT è l'acronimo di *File Allocation Table* ovvero una tabella ordinata di puntatori alla memoria.

Come file-system risulta essere molto limitato per gli usi attuali ma andava bene per i vecchi dischi (soprattutto per i floppy), seppur rimanga ancora usato sulle chiavette USB.

L'approccio però è non scalabile e dunque la FAT stessa può occupare molto spazio

File Allocation Table

Ogni riga della FAT è un puntatore ad un **cluster** (blocco) del disco. Ogni cluster ha dimensione variabile (può cambiare da partizione a partizione, ma resta fisso all'interno di una partizione) tra 2 e 32 KB, ed è costituito da settori di disco contigui.

Come conseguenza si ha che la tabella cresce con la grandezza della partizione. I puntatori sono valori di 12, 16 o 32 bit (FAT-12, FAT-16, FAT-32).

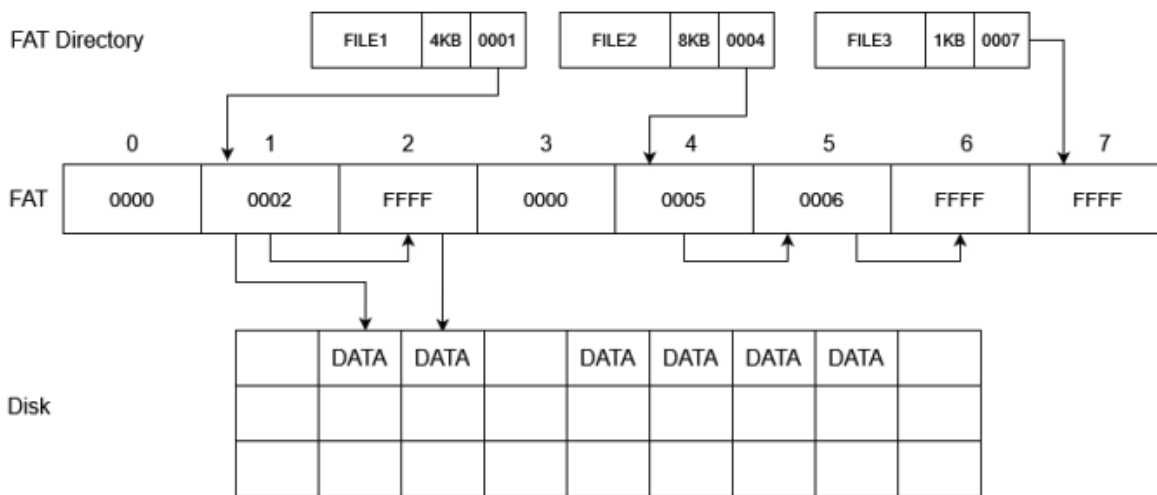
La parte della FAT relativa ai file aperti deve essere sempre mantenuta interamente in memoria principale; questa parte consente infatti di identificare i blocchi di un file e accedervi seguendo sequenzialmente i collegamenti nella FAT (un file è semplicemente una catena di indici, risulta essere un misto tra allocazione concatenata e indicizzata)

In questo modo però risulta avere overhead di spazio non indifferente infatti ci sta un puntatore (32 bit) per riga, e tante righe quanti cluster del disco. Con un disco da 100 GB e cluster da 1 KB, la FAT ha 100 milioni di righe, dunque $4B \cdot 1.000.000 = 400 \text{ MB}$ per la FAT

Funzionamento

Data la tabella FAT di puntatori, il puntatore i-esimo:

- se è tutti zero → l'i-esimo cluster del disco è libero
- se è tutti uno → valore speciale, vuol dire che è l'ultimo blocco del file
- se non è zero e non è un valore speciale → indica il cluster dove trovare il prossimo pezzo del file, oltre che la prossima entry della FAT per questo file



Ci sta una corrispondenza uno ad uno con i cluster nel disco. Ad esempio il FILE3 punta alla settima entry del FAT, in cui ci stanno memorizzati tutti 1; il che vuol dire che il settimo cluster è l'unico in cui ci sono i dati relativi al FILE3

Struttura

Boot sector	File allocation table 1	File allocation table 2 (duplicate)	Root directory	Other directories and all files
-------------	-------------------------	-------------------------------------	----------------	---------------------------------

Boot sector

Questa regione contiene le informazioni necessarie per l'accesso al volume ovvero il tipo e puntatore alle altre sezioni del volume, e il bootloader del sistema operativo in BIOS/MBR

FAT

Questa regione contiene due copie della file allocation table (utile in caso la tabella principale si corrompa) che sono sincronizzate ad ogni scrittura su disco

Permette, come già detto, di mappare il contenuto della regione dati, indicando a quali directory/file i diversi cluster appartengono

Regione Root Directory

E' una *directory table* che contiene tutti i file entry per la directory root di sistema (che ha dimensione fissa e limitata in FAT12 e FAT16, 265 entries). Contiene dunque tutti i metadati dei file (nome, timestamp, dimensione, ecc.)

In FAT32 è inclusa nella regione dati, insieme a file e directory normali, e non ha limitazioni sulla dimensione

Regione dati

E' la regione del volume in cui sono effettivamente contenuti i dati dei file e directory. Le directory, chiaramente, seguono la struttura FAT vista prima, i files sono semplicemente i dati contenuti nei vari cluster

Limitazioni

- Supporta file di dimensione massima 4GB (32 bit nel campo dimensione file nelle directory)
 - Non implementa journaling
 - Non consente alcun meccanismo di controllo di accesso ai file/directory
 - Ha un limite di dimensione delle partizioni a 2TB (2^{32} settori da 512GB)
-

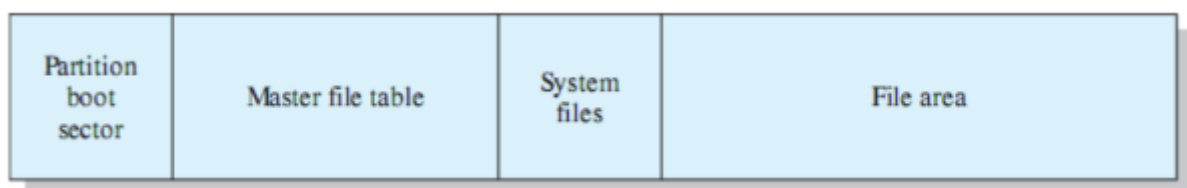
NTFS

L'NTFS (*New Technology File System*) è il file-system adottato a partire da Windows NT in poi.

Questo file system utilizza **UNICODE** per l'encoding dei nomi dei file, che possono raggiungere lunghezza massima di 255 caratteri

A differenza del precedente FAT, qui i file sono definiti da un insieme di attributi rappresentati come un *byte stream*, inoltre supporta hard e soft link e implementa il journaling

Formato del volume



Regione boot sector

Questa regione è basata sull'equivalente FAT, seppur alcuni campi sono in posizioni diverse, per il resto, valgono le stesse condizioni del FAT

MTF

La MTF (*Master File Table*) è la principale struttura dati dell'NTFS ed è unica per ciascun volume (differentemente dal FAT)

Questa viene implementata come un file composto da una **sequenza lineare di record** (massimo 2^{48}), da 1 a 4KB. Ogni record descrive un file che è identificato da un puntatore di 48 bit, mentre i rimanenti 16 bit dei 64 totali sono usati come numero di sequenza

Record

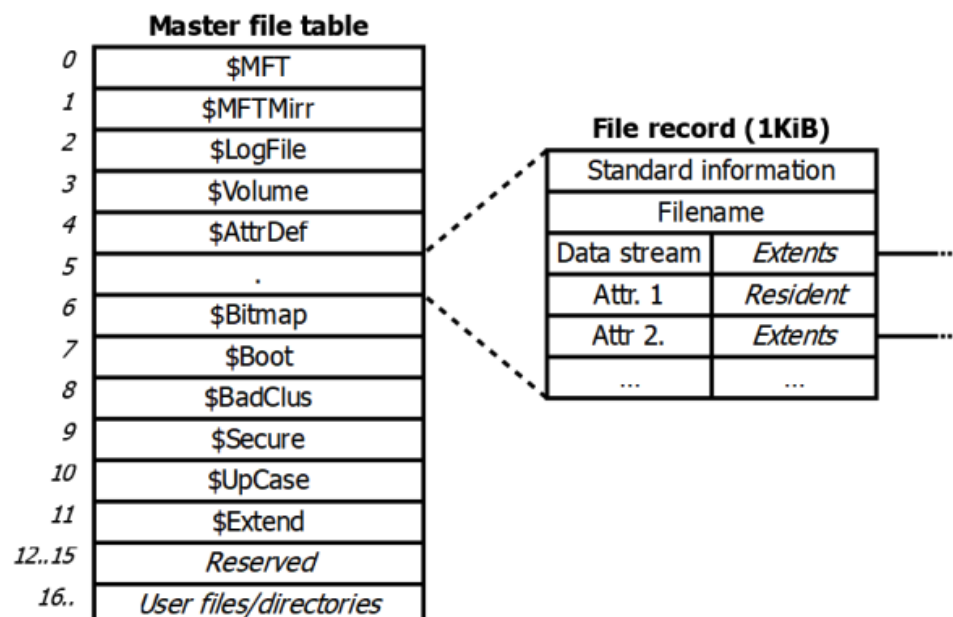
Ogni record contiene una lista di attributi: (attributo, valore)

- **attributo** → intero che indica il tipo di attributo (il contenuto di un file (\$DATA))
- **valore** → sequenza di byte

Se il valore dell'attributo è incluso direttamente nel record si parla di attributo *residente* mentre se è puntato dal record si parla di attributo *non residente*. Ad esempio, se il valore dell'attributo è troppo grande e non sta record sarà non residente (es. \$DATA)

I primi 27 record sono riservati per i metadati del file system:

- record 0 → descrive l'MFT stesso, in particolare, tutti i file nel volume
 - record 1 → contiene una copia dei primi record dell'MFT in modo non residente
 - record 2 → contiene le informazioni di journaling (metadata-only)
 - record 3 → contiene le informazioni sul volume (id, label, versione FS, ecc.)
 - record 4 → è una tabella degli attributi usati nell'MFT
 - record 5 → rappresenta la directory principale del volume e contiene i puntatori ai record della MFT che rappresentano file e directory nella root del volume
 - record 6 → definisce la lista dei blocchi liberi usando una **bitmap**
- Dal record 28 in poi ci sono i descrittori dei file normali



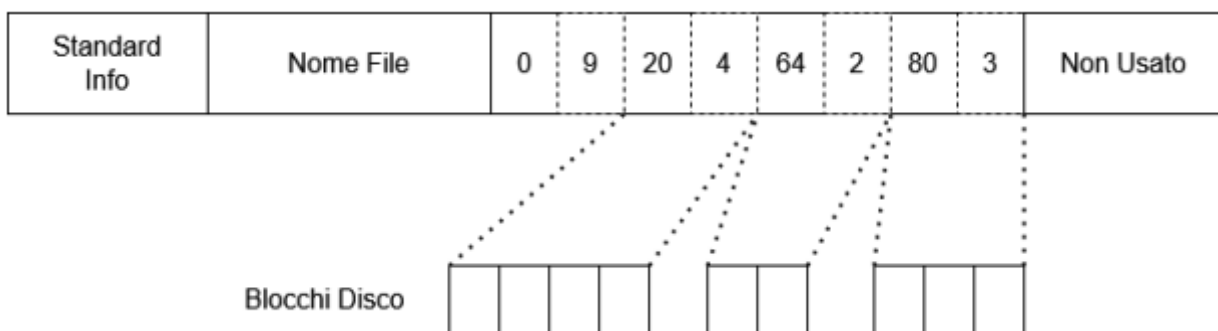
Files

L'NTFS cerca sempre di assegnare ad un file sequenze contigue di blocchi (quando possibile).

Per file piccoli (< 1KB) i dati sono salvati direttamente nel record dell'MFT, mentre per i file grandi, il valore dell'attributo indica la **sequenza ordinata dei blocchi sul disco dove risiede il file** (come l'allocazione indicizzata)

Per ogni file, esiste un record base nell'MFT, ma quando un file non utilizza tutti lo spazio disponibile nel record per la registrare i puntatori, i restanti byte vengono impostati a 0

Record base



In questo esempio si ha che il file è memorizzato in 9 cluster totali su 3 diverse porzioni di memoria contigue (es. 4 cluster a partire dal 20esimo) e si nota come un descrittore (record) singolo sia sufficiente per l'intera run (file)

Questa modalità di rappresentazione permette in potenza di descrivere file di dimensione illimitata, dipende tutto dalla contiguità dei blocchi; infatti analizziamo questi due esempi

- blocchi da 1KB; 1 file da 20GB, 20 sequenze da 1.000.000 di blocchi. Ognuna richiede $20 + 1$ coppie di 64 bit: $21 \cdot 2 \cdot 8B = 336B$
- blocchi da 1KB; 1 file da 64KB, 64 sequenze da 1 di blocco. Ognuna richiede $65 \cdot 2 \cdot 8B = 1040B$

Record con estensioni

Per file di grandi dimensioni possono essere necessari più record.

L'NTFS, in questi casi, usa una tecnica simile agli i-node di Unix/Linux; si ha infatti il record base che diventa un puntatore, invece che alla memoria, ad altri N record secondari (eventuale spazio rimanente nel record base contiene le prime sequenze del file)

Se i record estesi non rientrano in MFT per mancanza di spazio, vengono trattati come attributo non residente, e quindi salvati in un file dedicato con un apposito record salvato nell'MFT

File di k sequenze di blocchi, con due record di estensione

