

**POLITECNICO DI BARI**  
**CORSO DI LAUREA IN INGEGNERIA ELETTRONICA**  
**CORSO DI SISTEMI OPERATIVI**  
**ANNO ACCADEMICO 1998 / 99**

***FILE SYSTEM DI  
MICROSOFT WINDOWS NT***

**TARRICONE ANTONIO, FABRIZIO**

# ***SOMMARIO***

<b>1. Caratteristiche dell'NTFS</b>	<b>1</b>
1.1 Gestione dei volumi di grande capacità.....	1
1.2 Suddivisione dei file in stream .....	2
1.3 Nomi dei file in Unicode .....	2
1.4 Supporto di POSIX.....	2
1.5 Sicurezza e riservatezza dei dati .....	2
1.6 Ricerca veloce dei file .....	4
1.7 Recuperabilità .....	5
1.8 Ridondanza e correzione degli errori.....	5
<b>2. Struttura dell'NTFS</b>	<b>6</b>
2.1 Cluster.....	6
2.2 Tabella principale dei file .....	6
2.2.1 Indicizzazione dei file delle directory .....	8
2.2.2 Metadati e metafile .....	10
2.2.3 Compressione dei dati.....	11
<b>3. Recuperabilità</b>	<b>13</b>
3.1 Analisi.....	14
3.2 Ripetizione.....	15
3.3 Annullamento .....	15
<b>4. Ridondanza e correzione degli errori</b>	<b>16</b>
4.1 Gestione dei volumi.....	16
4.1.1 Volume set .....	16
4.1.2 Stripe set .....	17
4.2 Volumi con correzione degli errori .....	18
4.2.1 Mirror set (RAID level 1) .....	18
4.2.2 Stripe set con parità (RAID level 5) .....	19
4.2.3 Riserva di settori .....	19
4.3 Recupero dei cluster difettosi da parte dell'NTFS .....	20
<b>Bibliografia</b>	<b>21</b>

# ***1. CARATTERISTICHE DELL'NTFS***

Fino al 1988, la Microsoft supportava due distinti file system: la file allocation table (FAT) per DOS e per Windows, l'high performance file system (HPFS) per OS/2.

La FAT fu originariamente sviluppata per floppy disk dalla capacità molto bassa, e i suoi limiti si incominciarono ad avvertire quando si diffusero gli hard disk.

Con l'HPFS si superarono alcune limitazioni della FAT: si ridussero i tempi d'accesso alle directory molto estese, e si incluse la gestione di hard disk dalla capacità di 4 GB. In seguito l'HPFS fu ulteriormente migliorato, e poté supportare hard disk dalla capacità di 2 TB.

Tuttavia, nessuno dei due file system era adatto ad applicazioni critiche che richiedevano la possibilità di **recupero, sicurezza e correzione degli errori**.

Queste limitazioni spinsero la Microsoft a sviluppare un nuovo file system per il suo nuovo Windows NT, il *new techonogy file system* (NTFS).

## ***1.1 GESTIONE DEI VOLUMI DI GRANDE CAPACITÀ***

Il file system divide ogni volume in *unità elementari di allocazione* di eguale grandezza.

Nel file system FAT, tali unità d'allocazione sono dette *cluster*.

La FAT utilizza, per indirizzare i cluster, una struttura da 16 bit, perciò al massimo possono essere gestiti  $2^{16} = 65536$  cluster.

Dal momento che, il cluster è il più piccolo blocco indirizzabile nel volume, file più piccoli di un cluster non possono esistere. Di conseguenza la FAT non può gestire più di 65536 file.

Nell'HPFS le unità d'allocazione coincidono con i settori fisici del disco e sono dette *run*.

L'HPFS utilizza per indirizzare le unità d'allocazione una struttura a 32 bit con segno, quindi al massimo possono essere gestiti  $2^{31} = 2147483648$  run.

L'NTFS utilizza, per l'indirizzamento dei cluster, una struttura da 64 bit, perciò può gestire al massimo  $2^{64}$  cluster.

La dimensione dei cluster può essere variata all'atto della formattazione del volume, come può fare anche la FAT.

## **1.2 SUDDIVISIONE DEI FILE IN STREAM**

In un volume NTFS i file sono composti da *attributi*. L'istanza di ogni attributo è detta *stream*.

Sono attributi di un file il nome, il proprietario, la data e l'ora dell'ultima modifica, i dati veri e propri e così via.

Per accedere ad un file, è necessario specificare il nome dell'attributo sul quale si vuole operare. Ad esempio:

```
miofile.dat:stream2
```

L'attributo contenente i dati veri e propri non ha nome, quindi se si vuole accedere a questi, basta specificare solo il nome del file.

## **1.3 NOMI DEI FILE IN UNICODE**

L'*Unicode* è uno schema di codifica dei caratteri a 16 bit, e permette di rappresentare ogni simbolo delle principali lingue.

L'NTFS permette di specificare i nomi dei file e delle directory utilizzando questo codice. Inoltre, ogni nome può essere composto al massimo da 255 caratteri, e può contenere spazi e più occorrenze del "punto".

## **1.4 SUPPORTO DI POSIX**

L'NTFS supporta pienamente le caratteristiche del file system di POSIX:

- distinzione fra le maiuscole e le minuscole;
- indicazione dell'ora e della data dell'ultimo accesso al file;
- *hard link*, ossia riferimenti multipli allo stesso file.

## **1.5 SICUREZZA E RISERVATEZZA DEI DATI**

La sicurezza dei dati è essenziale per tutti gli utenti che elaborano informazioni private o riservate, come banche, ospedali, ed enti della difesa.

Questi utenti devono avere la garanzia che i loro dati siano protetti dagli accessi non autorizzati.

Prima che un processo possa aprire un handle per un oggetto, il sistema di protezione di Windows NT verifica che esso abbia le autorizzazioni per farlo.

Ogni oggetto in Windows NT ha un *descrittore di sicurezza*, e tutti gli utenti che eseguono il logon del sistema, devono fornire una password di identificazione.

Il sistema di protezione, grazie alla password di identificazione ed al descrittore di sicurezza, verifica se un processo di un certo utente può accedere all'oggetto.

Per quel che riguarda i volumi, le directory ed i file, il descrittore di sicurezza contiene per ogni *identità* (gruppo di utenti) una serie di *permessi*.

Le possibili identità sono descritte nella Tabella 1.

Identità	Significato
Local groups nel dominio contenente il server	Gruppi nel dominio contenente il server invisibili agli altri.
Global groups nel dominio contenente il server	Gruppi nel dominio contenente il server visibili agli altri.
Individual users nel dominio contenente il server	Singoli utenti nel dominio del server.
Global groups nei domini fidati	Gruppi nei domini fidati.
Individual users nei domini fidati	Singoli utenti nei domini fidati.
Everyone	Tutti gli utenti attuali e futuri.
System	Sistema operativo locale.
Network	Tutti gli utenti attuali e futuri che accedono dalla rete.
Interactive	Tutti gli utenti attuali e futuri che accedono direttamente dal server.
Creator owner	L'utente che ha creato il file o la directory.

**Tabella 1** – *Identità.*

I permessi si suddividono in permessi individuali, elencati nella Tabella 2, e permessi standard, elencati nella Tabella 3 e nella Tabella 4, che sono una composizione di permessi individuali.

R	Read
W	Write
X	Execute
D	Delete
P	Change Permissions
O	Take Ownership

**Tabella 2** – *Permessi individuali.*

No Access	-
Read	RX
Change	RWXD
Full Control	RWXDPO

**Tabella 3** – *Permessi standard per i file.*

	<b>Operazioni consentite</b>	<b>Operazioni consentite sui nuovi file</b>
No Access	-	-
List	RX	Non specificato
Read	RX	RX
Add	WX	Non specificato
Add & Read	RWX	RX
Change	RWXD	RWXD
Full Control	RWXDPO	RWXDPO

**Tabella 4** – *Permessi standard per le directory.*

Il sistema di sicurezza di Windows NT è molto sofisticato ed è conforme ai requisiti del livello C2 stabiliti dal Dipartimento della Difesa del Governo degli USA per i sistemi operativi affidabili, al livello C2 dell'NCSC (United States National Computer Security Center), al livello F-C2/E3 della commissione ITSEC (UK Information Technology Security Evaluation and Certification)

## **1.6 RICERCA VELOCE DEI FILE**

L'NTFS memorizza l'elenco dei file per ogni directory in ordine alfabetico. Questo gli permette di ricercare più velocemente i file, quando le directory sono molto estese.

## **1.7 RECUPERABILITÀ**

Sui personal computer si è sempre cercato di velocizzare le operazioni di I/O sui dischi, trascurando l'affidabilità.

Con la diffusione dei PC nell'ambito aziendale l'affidabilità è diventata sempre più importante, infatti, se un volume diventa inaccessibile, la velocità delle operazioni non ha più importanza.

Per rispondere a queste esigenze, l'NTFS fa in modo che ogni operazione sia considerata come una *transazione*. Questo meccanismo fa in modo che ogni transazione avviata debba essere portata a termine, oppure, se un'avaria del sistema la interrompesse, tutte le parti già completate devono essere annullate (*abort* e *roll back*). In questo modo il contenuto del volume è sempre coerente.

## **1.8 RIDONDANZA E CORREZIONE DEGLI ERRORI**

Per le applicazioni che non possono rischiare la perdita dei dati, la ridondanza offre un altro livello di protezione.

Windows NT fornisce il supporto del *RAID* (*Redundant Arrays of Inexpensive Disks*) livello 1 e livello 5.

Con il RAID livello 1 si possono duplicare i dati su più volumi (*mirror*), in modo che, se una copia non è più accessibile, c'è l'altra.

Con il RAID livello 5 i dati sono suddivisi in *spezzoni* e ognuno di questi è distribuito su più volumi, in più in ogni spezzone vengono inclusi i codici di parità che permettono di ricostruire le informazioni ospitate su un volume andato in avaria, a partire dagli altri.

In un volume NTFS formattato come "*a prova d'errore*" (*fault tollerant*), quando l'NTFS cerca di leggere un cluster che cade su settori difettosi, automaticamente è attuata la seguente procedura:

- ricerca di una copia valida dei dati richiesti (RAID livello 1 e livello 5);
- allocazione di un nuovo cluster;
- memorizzazione nel nuovo cluster della copia valida;
- marcatura del cluster difettoso.

Se il volume non fosse stato fault tollerant, l'NTFS avrebbe marcato lo stesso il cluster come difettoso, ma non sarebbe stato in grado di recuperare una copia valida dei dati.

## **2. STRUTTURA DELL'NTFS**

### **2.1 CLUSTER**

Il *cluster* è l'unità fondamentale allocabile su un volume.

La dimensione dei cluster, o *fattore di raggruppamento*, è determinata all'atto della formattazione del volume, è sempre una potenza del 2 di settori fisici, e può essere stabilita dall'utente.

Cluster più grandi permettono la riduzione della frammentazione, ma comportano un maggior spreco di spazio sul volume.

L'NTFS fa riferimento solo ai cluster. Questo gli permette di essere indipendente dalla dimensione dei settori, e di gestire dischi non standard.

Ogni cluster è indicato da un *numero logico* (LCN: *Logical Cluster Number*) da 64 bit. In pratica, si tratta di un numero progressivo, assegnato ai cluster partendo dall'inizio del volume fino alla fine.

Dal momento che il numero massimo di cluster indirizzabili è fisso, quando la capacità dei dischi aumenta, è necessario incrementare il fattore di raggruppamento.

L'NTFS suddivide i file in cluster logici. Ogni cluster logico è indicato da un *numero virtuale* (VCN: *Virtual Cluster Number*). In pratica, il primo cluster logico di un file ha VCN 0, il secondo 1, e così via.

Cluster logici contigui non occupano necessariamente cluster fisici contigui.

### **2.2 TABELLA PRINCIPALE DEI FILE**

La *tabella principale dei file* (MFT: *Master File Table*) costituisce la struttura sulla quale l'NTFS basa la gestione dei volumi. Ogni record della MFT ha la dimensione di 1 KB.

I principali attributi di un file sono elencati in Figura 1, tuttavia un file non li deve necessariamente avere tutti. Inoltre un file può avere attributi definiti dall'utente, come può avere diversi attributi dati. In questo caso gli attributi dati devono avere un nome.

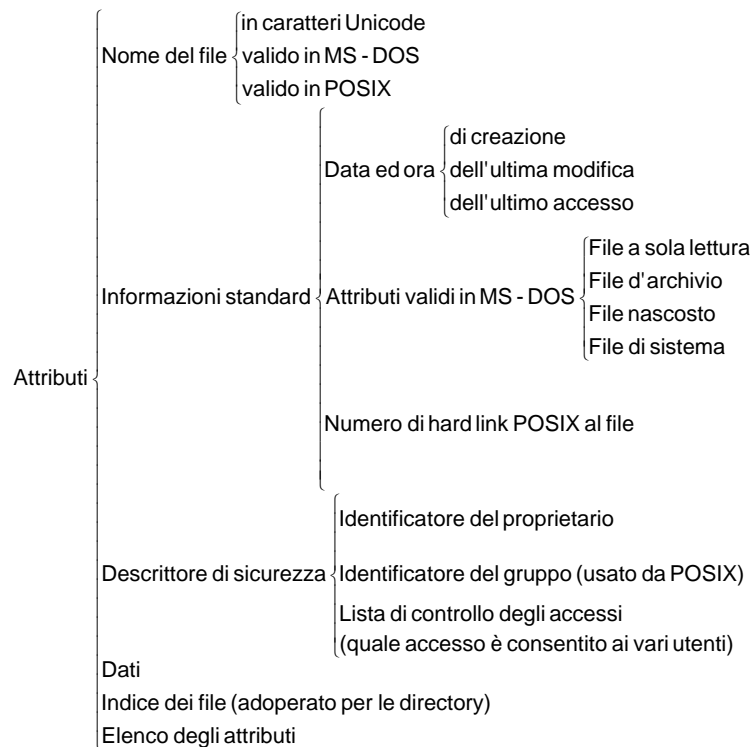
I record della MFT sono composti dagli attributi del file.

Un file non molto grande e con pochi attributi, può essere interamente memorizzato in un record della MFT.

Se gli attributi sono troppi e non possono essere contenuti in un record della MFT, se ne adoperano più di uno: il primo di questi è detto *record di*



*base* e contiene l'attributo “elenco degli attributi”. L’elenco degli attributi contiene i puntatori ai record della MFT che ospitano gli attributi restanti.



**Figura 1** – Attributi principali.

L’istanza di un attributo, quando è di piccole dimensioni, può essere memorizzata interamente in un campo di un record della MFT. In questo caso si dice che l’attributo è *residente*.

Se l’istanza di un attributo è troppo grande per essere memorizzata interamente in un record della MFT, viene allocata sul volume un’area esterna, detta *run* o *extent*, contenente lo stream. In questo caso si dice che l’attributo *non è residente*.

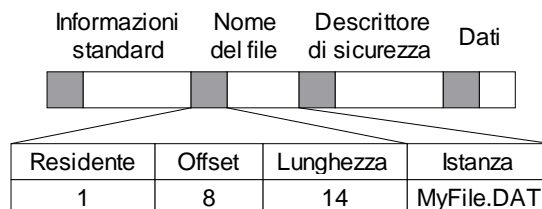
La dimensione dei run è di 2 o 4 KB, a seconda che la grandezza dei cluster sia minore o maggiore di 4 KB. Non è detto che i cluster componenti il run siano contigui. In casi particolari la dimensione dei run può essere maggiore di 2 o 4 KB.

Se la dimensione di uno stream eccede la capacità del run, saranno allocati più di uno.

L’istanza di ogni campo della MFT inizia sempre con un *header* (*intestazione*) che indica se lo stream è residente o no, e nel primo caso contiene la distanza da se stesso dell’inizio dell’istanza, oltre che la lunghezza di quest’ultima (Figura 2).

Se l’attributo non è residente, l’header contiene le informazioni necessarie per localizzare i run che contengono lo stream (Figura 3). Queste informazioni consistono nel *fattore di bloccaggio*, ossia il numero di cluster

logici contenuti in un cluster fisico, il VCN del primo cluster logico contenuto nel run ed il corrispondente LCN, la lunghezza in cluster fisici del run.

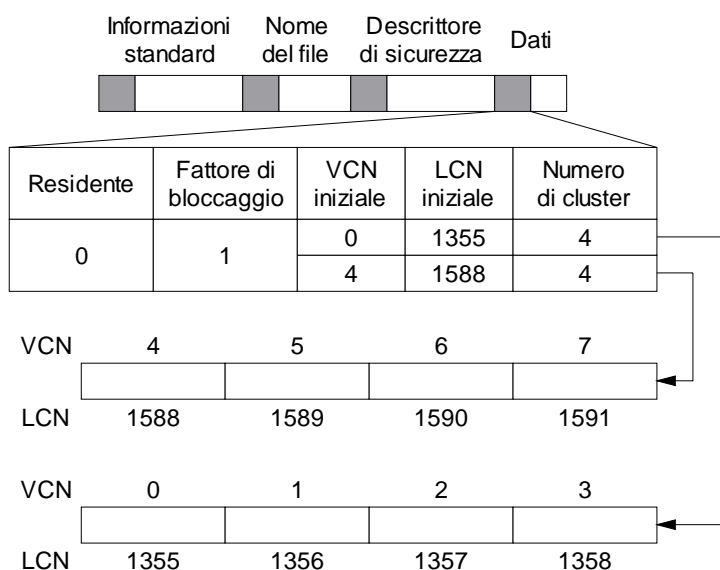


**Figura 2** – Esempio di header di un attributo residente.

Osserviamo che, per l'attributo dati il fattore di bloccaggio è sempre unitario, non potendo l'NTFS conoscere la sua struttura logica.

Fra gli attributi standard, solo quelli che possono aumentare di dimensione possono essere non residenti.

Nell'eventualità che gli attributi non residenti, siano così frammentati da richiedere molti LCN per rintracciare tutti i run, si adoperano più record della MFT.



**Figura 3** – Esempio di header di un attributo non residente.

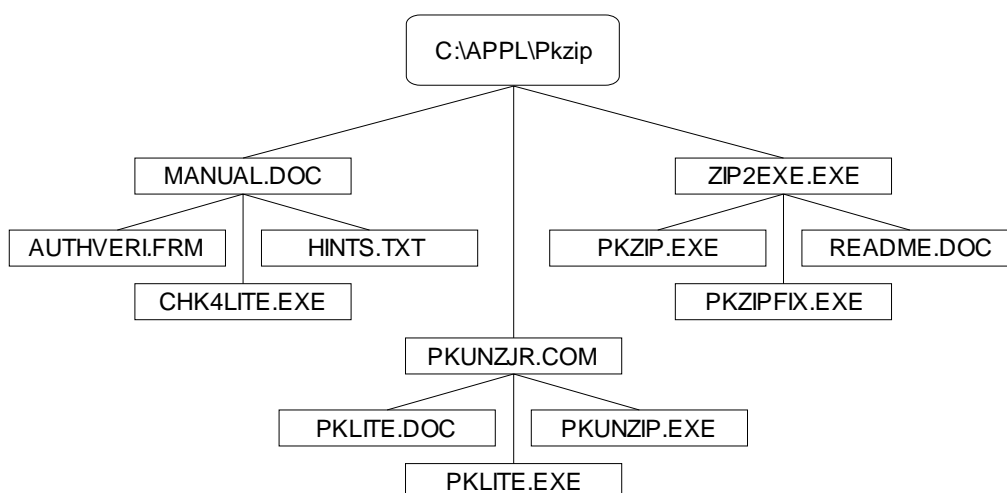
### 2.2.1 INDICIZZAZIONE DEI FILE DELLE DIRECTORY

L'NTFS organizza l'indice dei file e delle sotto directory contenute in una directory, sfruttando una struttura ad albero a due livelli. I file sono

sistemati nei vari nodi in modo che, percorrendo la struttura in *ordine differito*, si incontrano i nomi in ordine alfabetico crescente. Ad esempio, l'NTFS organizzerà i nomi dei file della directory in Figura 4 come in Figura 5.

C:\APPL\Pkzip
ZIP2EXE.EXE
AUTHVERI.FRM
CHK4LITE.EXE
HINTS.TXT
MANUAL.DOC
PKLITE.DOC
PKLITE.EXE
PKUNZIP.EXE
PKUNZJR.COM
PKZIP.EXE
PKZIPFIX.EXE
README.DOC

**Figura 4** – Esempio del contenuto di una directory.



**Figura 5** – Organizzazione ad albero di una directory.

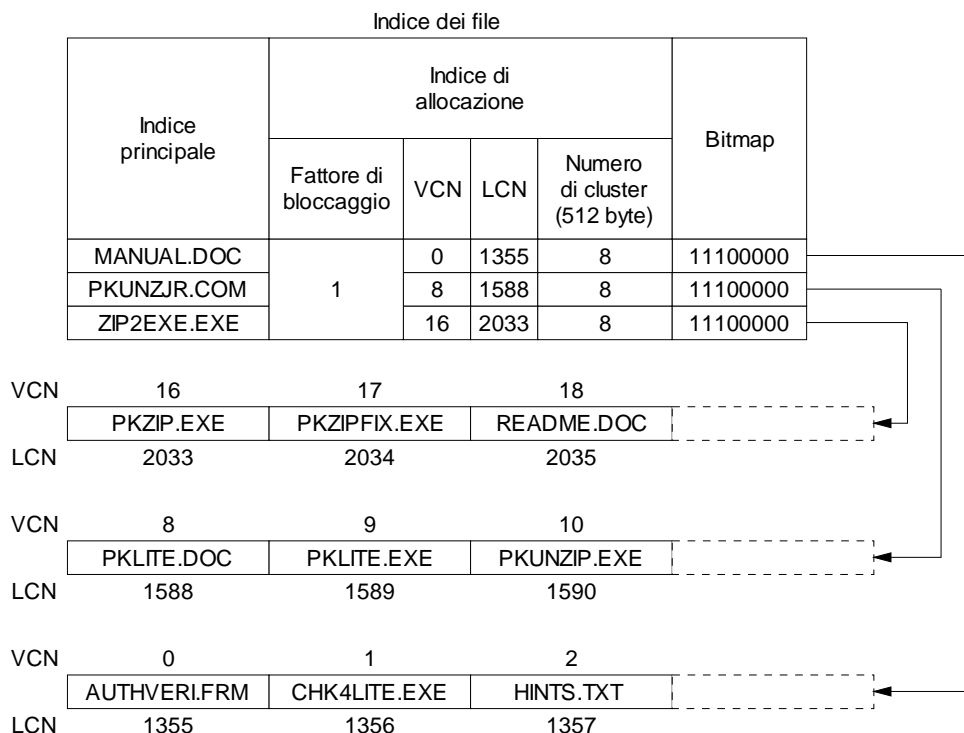
I record della MFT che descrivono le directory hanno un particolare attributo: “indice dei file”. Questo attributo è in realtà l’unione di tre diversi attributi: “indice principale”, “indice d’allocazione”, “bitmap”.

L’indice principale è sempre residente e contiene il primo livello dell’albero.

L’indice d’allocazione, invece, contiene i riferimenti ai run contenenti l’elenco dei figli di ciascun nodo del primo livello.

In questo caso i run hanno sempre la dimensione di 4 KB, indipendentemente da quella dei cluster, sono detti *buffer indice*, e i cluster componenti il run devono essere contigui.

Per ogni file, non è memorizzato soltanto il nome, come è mostrato nella Figura 6 per semplicità, ma è memorizzato il riferimento al record della MFT che lo descrive, la dimensione, la data e l'ora di creazione. Queste ultime informazioni, sebbene possano essere ricavate dalla MFT, sono duplicate per permettere una navigazione più veloce fra le directory.



**Figura 6** – Organizzazione dell'indice di una directory (in realtà la struttura si estende orizzontalmente e non verticalmente).

Infine, la bitmap indica per ogni buffer indice quali VCN sono utilizzati e quali sono liberi.

In questo caso, un cluster logico contiene le informazioni per un file, perciò il fattore di bloccaggio dipende dalla dimensione dei cluster fisici.

Al crescere delle dimensioni della directory, non aumenta il numero di livelli dell'albero, ma aumenta l'estensione orizzontale di ognuno di questi. Se dovesse aumentare l'estensione del primo livello, sarebbero adoperati più record della MFT. Se dovesse aumentare l'estensione del secondo livello, sarebbero adoperati più buffer indice.

### **2.2.2 METADATI E METAFIILE**

Nell'NTFS i dati memorizzati su un volume, sono contenuti in file, compresi i *metadati*, ossia le informazioni usate per implementare la struttura del file system.

Questa scelta permette una facile espansione dei volumi senza la necessità di riformattarli.

Il nome di ogni *metafile* inizia sempre con il simbolo \$, ed i principali sono descritti nella Tabella 5.

Metafile	Contenuto
\$MFT	Tabella principale dei file (MFT).
\$MFTMirr	Copia dei record della MFT contenenti le informazioni sui metafile. Questa copia parziale della MFT è posta al centro del disco, è adoperata se per qualche motivo una parte della MFT non può più essere letta.
\$LogFile	File di log contenente una traccia di tutte le operazioni che alterano la struttura del volume e non sono ancora state completate. E' adoperato dall'NTFS per recuperare il volume dopo un'avaria del sistema.
\$Bitmap	Bitmap indicante i cluster liberi e quelli occupati.
\$Boot	Codice per avviare il sistema (bootstrap).
\$BadClus	Elenco dei cluster difettosi.
\$Volume	Contiene il nome del volume, la versione dell'NTFS con cui è stato formattato, ed un flag che indica se si è verificato un danneggiamento della struttura del disco.
\$AttrDef	Indica gli attributi supportati dal volume, e se possono essere indicizzati e recuperati da un'operazione di recupero.

*Tabella 5 – Descrizione dei metafile.*

### **2.2.3 COMPRESSIONE DEI DATI**

L'NTFS supporta la compressione dei dati utente. E' in grado di comprimere singoli file, singole directory, o l'intero volume.

Non ci occuperemo dell'algoritmo di compressione adoperato, ma solo di come l'NTFS distingue i file compressi da quelli no.

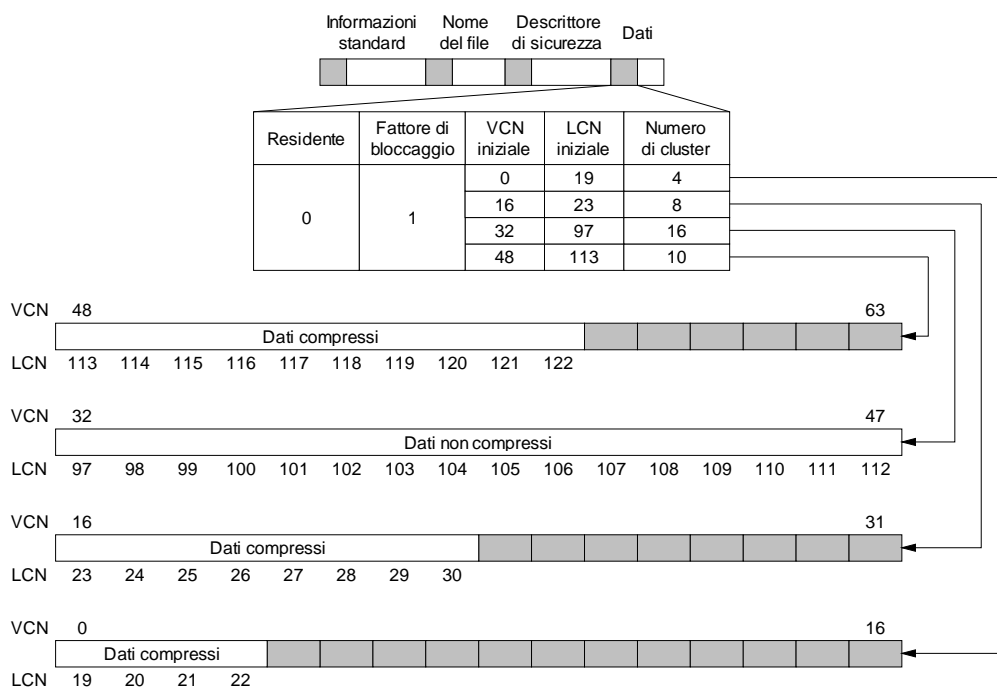
Quando si richiede all'NTFS di comprimere un file, questo lo suddivide in un run di 16 cluster detti *unità di compressione*, e cerca di comprimere ognuna di queste indipendentemente dagli altri.

L'NTFS comprime un run solo se può risparmiare almeno un cluster, se no non procederà alla compressione.

L'NTFS riesce a distinguere i run compressi da quelli no grazie all'header del campo dati. Se un cluster logico di un run non risulta mappato su nessun cluster fisico, allora il run è compresso.

Osserviamo la Figura 7 e consideriamo il riferimento al primo run: sommando 4 (numero di cluster) a 0 (VCN iniziale) otteniamo 4 e non 16 come riportato nel riferimento del run successivo, questo ci indica che il primo run è compresso.

Consideriamo il riferimento al terzo run: sommando 16 (lunghezza del run) a 32 (VCN iniziale) otteniamo 48 come riportato nel riferimento al run successivo, questo ci indica che il terzo run non è compresso.



**Figura 7** – Record della MFT relativo ad un file compresso.

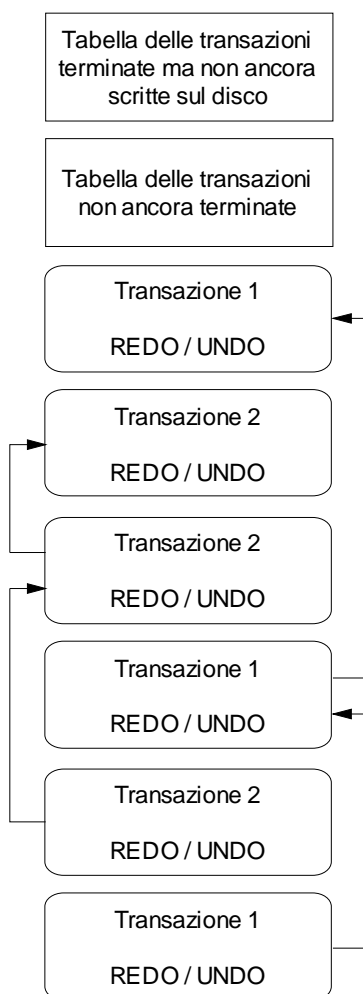
### 3. RECUPERABILITÀ

La **recuperabilità** dell'NTFS consiste nel fare in modo che il contenuto del volume sia sempre coerente. Questo non evita la perdita dei dati, ma fa sì che il volume sia sempre accessibile.

La tecnica adoperata dall'NTFS per implementare la recuperabilità, è stata sviluppata per l'elaborazione delle *transazioni*, ed è nota come *write ahead logging*.

L'NTFS, per monitorizzare le transazioni sul volume, adopera due tabelle:

- tabella delle transazioni non ancora terminate, più brevemente tabella delle transazioni;
- tabella delle transazioni terminate non ancora scritte sul disco, più brevemente tabella delle pagine alterate.



**Figura 8** – Struttura del file di log.

La tabella delle transazioni contiene la descrizione di tutte le sotto operazioni già eseguite per ogni transazione ancora attiva. Queste sotto operazioni saranno annullate (*roll back*) al riavvio del sistema in caso di crash.

La tabella delle pagine alterate contiene la descrizione di tutte le sotto operazioni già svolte per le transazioni terminate (*committed*). Questa tabella è necessaria perché Windows NT fornisce il caching del disco, perciò anche se una transazione è stata completata, non è certo che il volume sia stato aggiornato. Queste sotto operazioni saranno ripetute al riavvio del sistema in caso di crash.

Queste due tabelle sono allocate in memoria. Per evitare la loro perdita a causa di un'avaria, NTFS ne fa una copia nel *file di log*. L'area del file di log nella quale sono memorizzate le tabelle, è detta *area di checkpoint*.

Prima di eseguire una sotto operazione di una transazione, NTFS aggiorna le tabelle in memoria, e poi scrive nel file di log un *record di aggiornamento* contenente sia le informazioni per annullarla sia quelle per ripeterla. I record di aggiornamento di una transazione sono tutti concatenati fra loro.

Per eseguire il recupero di un volume, NTFS esegue tre fasi: analisi, ripetizione, annullamento.

Osserviamo che anche le operazioni di logging sono sottoposte a caching, tuttavia il gestore della cache sa che quando deve scaricare i dati sul disco, deve aggiornare per primo il file di log.

### **3.1 ANALISI**

Nella fase di analisi NTFS legge l'area di checkpoint del file di log. Per ogni transazione descritta nelle tabelle, è determinata la posizione dell'ultimo record di aggiornamento (i record di aggiornamento di una transazione sono concatenati fra loro).

Se un record di aggiornamento segnala che una transazione è conclusa, NTFS trasferisce la descrizione di questa dalla tabella delle transazioni a quella delle pagine alterate.

Sfruttando i record di aggiornamento, NTFS può aggiornare le due tabelle e quindi procedere alla ripetizione delle sotto operazioni componenti le transazioni concluse, e all'annullamento di quelle componenti le transazioni non concluse.

Può accadere che, sebbene una transazione sia conclusa, il gestore della cache non abbia fatto in tempo ad aggiornare il file di log. In questa eventualità la transazione sarà annullata.



### ***3.2 RIPETIZIONE***

Nella fase di ripetizione NTFS, per ogni transazione conclusa, analizza le varie sotto operazioni, e verifica che siano effettivamente state eseguite sul volume. Ogni operazione non effettuata è ripetuta.

La fase di ripetizione inizia dalle sotto operazioni più vecchie.

### ***3.3 ANNULLAMENTO***

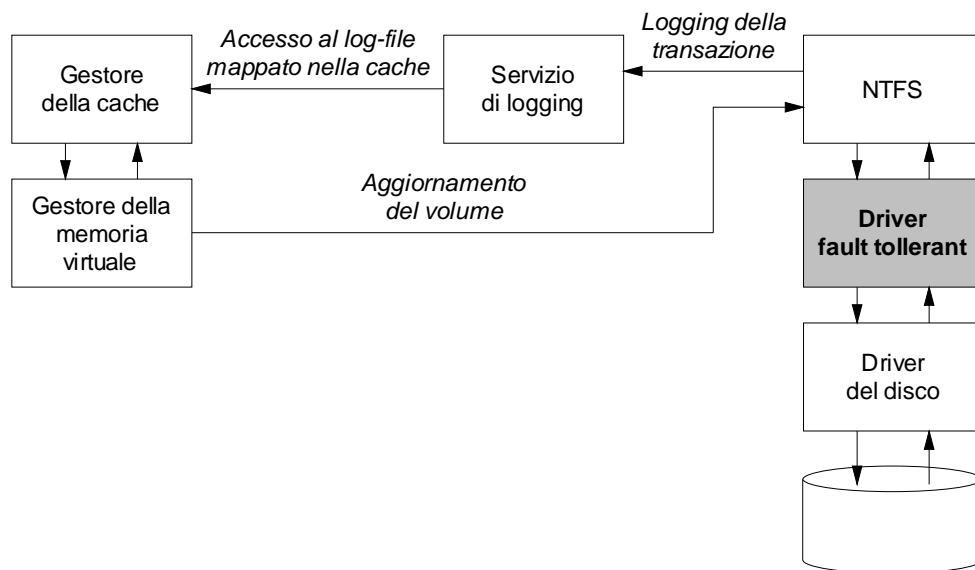
Analogamente alla fase di ripetizione, durante l'annullamento NTFS analizza le varie sotto operazioni delle transazioni non concluse, e verifica se sono state eseguite sul volume. Ogni sotto operazione eseguita sul volume, è annullata.

La fase di annullamento inizia dalle sotto operazioni più recenti.

Osserviamo che anche la fase di recupero è sottoposta a logging, in quanto il sistema, in caso di avaria, deve essere in grado di annullare ogni operazione di annullamento e ripetizione non conclusa, e ripetere quelle terminate.

## 4. RIDONDANZA E CORREZIONE DEGLI ERRORI

Il supporto della correzione degli errori è fornito all'NTFS dal *Fault Tollerance Disk Driver (FtDisk)*. FtDisk è un driver che si pone fra l'NTFS stesso ed il driver del disco, inoltre può funzionare con qualsiasi file system supportato da Windows NT.



*Figura 9 – Interazione fra le componenti.*

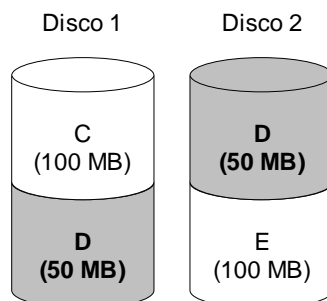
### 4.1 GESTIONE DEI VOLUMI

FtDisk implementa alcune funzionalità per la gestione dei volumi non direttamente collegate alla correzione degli errori, come i *volume set* e gli *stripe set*. Queste funzionalità organizzano i volumi in modo da aumentare l'efficienza delle operazioni su disco.

#### 4.1.1 VOLUME SET

Un volume set, o *volume composito*, è un singolo volume composto da aree (al massimo 32) che possono risiedere sullo stesso disco o su dischi diversi (Figura 10).

I volumi composti sono utili quando si hanno molte aree libere di piccola capacità, che non conviene adoperare singolarmente, oppure quando si vuole creare un grande volume a partire da supporti di piccola capacità.



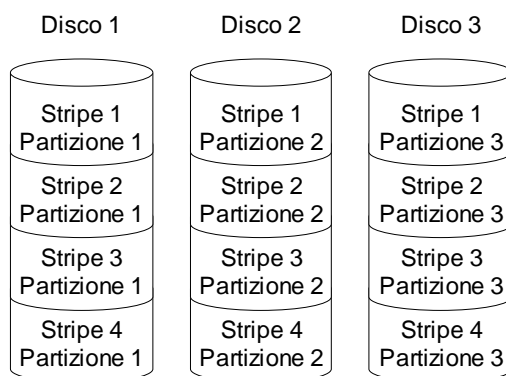
**Figura 10** – Volume set (il volume composto è D).

Se un volume composto è stato formattato per l'NTFS, può essere in seguito ampliato senza influire sui dati già memorizzati. Questo è il gran vantaggio di avere sotto forma di file tutti i dati necessari alla gestione del volume (metadati). I file, infatti, possono essere ampliati dinamicamente, mentre, le tabelle adoperate dalla FAT, hanno una dimensione e una posizione fissa.

FtDisk nasconde la configurazione fisica del volume ai file system installati. Quando il file system ha bisogno di eseguire un'operazione sul disco, la richiesta la fa al FtDisk.

#### **4.1.2 STRIPE SET**

Uno stripe set è una serie di partizioni, una per disco, combinate in un solo volume logico (Figura 11).



**Figura 11** – Stripe set.

La Figura 12 mostra come appaiono i settori all'NTFS.



**Figura 12** – Numerazione logica dei settori.

La dimensione di ogni *stripe* è di 64 KB. Questa dimensione relativamente piccola, fa sì che i dati tendano a ripartirsi in modo omogeneo fra i dischi.

Dal momento che si può accedere simultaneamente su dischi diversi, i tempi di latenza per le operazioni sono drasticamente ridotti.

## **4.2 VOLUMI CON CORREZIONE DEGLI ERRORI**

Le metodologie di gestione dei volumi sono utilizzate da FtDisk per implementare due diversi schemi di memorizzazione con ridondanza: *mirror set* e *stripe set con parità*.

### **4.2.1 MIRROR SET (RAID LEVEL 1)**

In un mirror set, il contenuto di una partizione (*partizione primaria*) è duplicato in un'altra (*partizione speculare*) della stessa dimensione della primaria, ma residente su un altro disco.

Quando l'NTFS scrive su un volume, FtDisk automaticamente fa una copia dei dati sia nella partizione primaria, che in quella speculare.

Se la partizione primaria non è più leggibile, FtDisk soddisfa le richieste accedendo a quella speculare.

I mirror set migliorano anche il throughput del sistema, infatti, FtDisk ripartisce automaticamente le richieste in lettura fra la partizione primaria e quella speculare, secondo il carico su ognuna di esse.

### 4.2.2 STRIPE SET CON PARITÀ (RAID LEVEL 5)

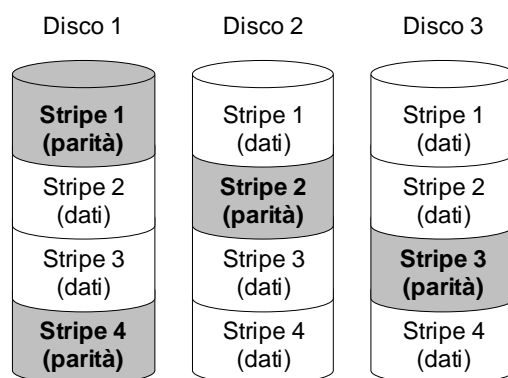
Gli stripe set con parità adoperano una loro partizione per memorizzare i codici di parità delle altre.

Il byte *i*-esimo della partizione di parità è ottenuto calcolando lo XOR fra i byte *i*-esimi delle altre partizioni.

Se un disco andasse in avaria, FtDisk potrebbe ricostruire le partizioni non più accessibili sfruttando le altre.

Le partizioni di parità vengono ricalcolate e riscritte sul disco ogni volta che gli stripe sono modificati. Per questa ragione le partizioni di parità non sono ospitate tutte sullo stesso disco, perché se così fosse, questo sarebbe sempre in uso, costituendo un collo di bottiglia.

FtDisk determina la partizione che conterrà i codici di parità, calcolando il modulo fra in numero dello stripe ed il numero di dischi. Se ad esempio abbiamo tre dischi, lo stripe 1 avrà i codici di parità ospitati sul disco 1, lo stripe 2 li avrà sul disco 2, lo stripe 3 li avrà sul disco 3, lo stripe 4 li avrà sul disco 1, e così via (Figura 13).



*Figura 13 – Disposizione delle partizioni di parità.*

### 4.2.3 RISERVA DI SETTORI

La ridondanza non è utilizzata solo per recuperare i dati dopo l'avaria totale di un disco, ma anche per recuperarli da un singolo settore che diventi difettoso.

FtDisk adopera una tecnica nota come *riserva di settori* per rimpiazzare dinamicamente i dati persi quando un settore del disco non è più leggibile.

Questa tecnica si avvale di una particolare funzionalità di alcuni dischi rigidi, che mettono a disposizione un gruppo di settori di “riserva”.

Se FtDisk riceve un messaggio d'errore dal disco, recupera i dati che stavano sul settore danneggiato, leggendoli dalla partizione speculare o

ricalcolandoli sfruttando i codici di parità, e li copia in uno di riserva. Tutto questo avviene in modo trasparente al file system, e senza il suo intervento.

I dischi rigidi che mettono a disposizione i settori di riserva, sono quelli SCSI.

Se si verifica un errore per settore difettoso su un disco non SCSI, o su uno che ha esaurito la riserva, FtDisk recupera sempre una copia valida dei dati, e avverte il file system che non c'è più ridondanza.

### ***4.3 RECUPERO DEI CLUSTER DIFETTOSI DA PARTE DELL'NTFS***

L'NTFS è in grado di sostituire dinamicamente un cluster contenente settori danneggiati, e interviene quando FtDisk non è in grado di sostituire il settore difettoso, o quando FtDisk non è installato, o quando non è stato attivato il supporto di ridondanza dei dati.

Quando l'NTFS riceve una segnalazione d'errore per un settore difettoso, aggiorna la tabella dei cluster rovinati, ne alloca uno nuovo, e poi aggiorna la mappa VCN/LCN del file che cade sul cluster rovinato. Se FtDisk è installato ed è attivato il supporto per la ridondanza, l'NTFS è in grado di ripristinare la ridondanza, copiando nel nuovo cluster i dati recuperati da FtDisk. Se FtDisk non è installato, o non è attivato il supporto di ridondanza, i dati memorizzati nel settore rovinato sono persi.

Per aumentare il grado di protezione dei metadati da mal funzionamenti, l'NTFS ha su ogni volume due copie del metafile \$Boot, che contiene il codice per il bootstrap, e una copia dell'area della MFT contenente i riferimenti ai metafile (\$MFTMirr).

# ***BIBLIOGRAFIA***

Inside Microsoft Windows NT  
David A. Salomon  
Mondadori Informatica – Microsoft Press