

Universitatea Politehnica din Bucuresti

Facultatea de Electronica, Telecomunicatii si Tehnologia Informatiei

Sistemul de fisiere NTFS

Studenti: MAVRU Anca

IONESCU Liviu

Grupa: 432 A

Cuprins:

1.	Concepte generale.....	3
2.	Structura unei partitii NTFS.....	6
3.	Fisiere cu seturi multiple de date.....	8
4.	Criptarea si compresia NTFS.....	10
5.	Bibliografie.....	12

Sistemul de fisiere NTFS

1. Concepte generale (realizat de Mavru Anca)

Acronimul **NTFS** vine de la **New Technology File System**, denumirea in engleza pentru un sistem de fisiere dezvoltat initial pentru Windows NT si apoi extins si imbunatatit pentru Windows 2000.

NTFS dispune de trei versiuni: v1.2 folosita de NT 3.51 si NT 4, v3.0 folosita de Windows 2000 si v3.1 folosita de Windows XP.

Principalele facilitati pe care le ofera acest sistem de fisiere sunt:

- folosirea adreselor de disc de 64 biti;
- poate suporta partitii de pana la 2^{64} octeti;
- permite folosirea unei game largi de caractere in numele fisiereilor cum ar fi: caractere Unicode, spatii, puncte etc. si de asemenea se face distinctie intre literele mari si cele mici; numele de fisiere pot avea o dimensiune maxima de 255 de caractere;
- fisierele pot fi indexate;
- permite utilizarea fisiereilor cu seturi multiple de date.

Alte avantaje ale NTFS sunt date de toleranta la erori, securitate, comprimarea fisiereilor si alocarea unei cote parti din disc.

Toleranta la erori. Acest lucru se intampla deoarece NTFS corecteaza automat erorile de tip I/O ale hard-discurilor. Atunci cand sistemul de operare scrie un

fișier pe disc, o copie a fișierului respectiv este păstrată în memoria RAM. În cazul în care fișierul scris pe disc nu coincide cu copia din memoria RAM, datele nu sunt disponibile pentru citire iar NTFS returnează un mesaj de eroare.

NTFS dispune de numeroase opțiuni de **securitate**. Pentru accesul la directoare sau fișiere individuale, NTFS alocă permisiuni local sau la distanță. Insa cea mai importantă măsură de securitate folosită este Encrypting File System (sistemul de fișiere criptat) care previne accesul neautorizat la structurile sistemului de operare, dar și la fișiere sau directoare.

Comprimarea directoarelor sau a fișierelor individuale este realizată de NTFS în timp real.

NTFS poate economisi resurse prin **alocarea unei cote parti din disc**, adică prin faptul că îi permite administratorului de sistem să aloce unui utilizator accesul doar la porțiunea de disc de care acesta are nevoie și care este suficientă pentru aplicațiile sale.

Cu privire la dezavantaje, putem afirma că NTFS este un **sistem de fișiere consumator de resurse** iar acest lucru afectează în mod direct viteza de rulare a sistemului de operare.

O altă problemă este dată de faptul că NTFS utilizează pentru stocarea informațiilor Master File Table (MFT). Atunci când MFT crește în dimensiune și devine fragmentat viteza de accesare a informației scade. O soluție a acestei probleme este alocarea unui spațiu mai mic MFT-ului.

O altă facilitate consumatoare de resurse a sistemului NTFS este actualizarea automată a câmpurilor referitoare la data ultimei accesări a fișierelor și folderelor. Totuși, aceasta poate fi dezactivată prin intermediul registrelor.

În NTFS putem identifica următoarele tipuri de fișiere^[1]:

- **fișiere sistem**: sunt fișierele care conțin informații ce sunt folosite numai de către sistemul de operare (metadata).
- **fișiere cu seturi multiple de date (Alternate Data Streams – ADS)**: sunt fișiere care pe lângă setul de date principal (implicit), mai conțin și alte seturi distincte de date. Toate aceste seturi de date sunt reprezentate prin atribute de tip *Data*.
- **fișiere arhivate**: NTFS poate arhiva și dezarhiva fișierele „*onthe fly*”, adică în momentul efectuării operațiilor de scriere și respectiv, citire a datelor din ele. Acest mecanism este invizibil aplicațiilor ce utilizează astfel de fișiere.
- **fișiere criptate**: EFS (*Encrypted File System*) oferă suport pentru a stoca fișiere criptate pe un volum NTFS. Criptarea este transparentă pentru utilizatorii care au

cerut criptarea fisierului. Accesul celorlalti utilizatori nu este permis la aceste fisiere.

- **fisiere „rare” (sparse files)**: sunt fisiere în care informatia scrisa nu se gaseste într-o singura zona contigua, ci zonele in care s-au scris date alterneaza cu zone mari în care nu s-au scris („gauri”). NTFS permite setarea unui atribut special al acestor fisiere, prin care se indica sistemului de I/O sa aloce spatiu pe disc numai pentru zonele efectiv scrise din fisier.
- **fisiere de tip „hard-link”**: sunt fisiere speciale introduse de NTFS5. Aceste fisiere permit ca un fisier sa poate fi accesat prin mai multe cai fara ca datele efective sa fie duplicate. Daca stergem un fisier la care exista si o alta legatura, datele nu vor fi sterse de pe disc pana cand nu se sterg toate legaturile.

NTFS gestioneaza drepturile de acces prin intermediul unei **liste de control al accesului (ACL)**. Drepturile de acces se numesc permisiuni. ACL-urile contin informatii care definesc drepturile pe care fiecare utilizator sau grup de utilizatori le au asupra unui fisier.

Pentru optimizarea controlului asupra listelor de acces, au fost introduse incepand cu Windows 2000 **componentele de permisiuni**.

2. Structura unei partitii NTFS (realizat de Ionescu Liviu)

Formatarea unei partitii conform NTFS se face prin crearea unor fisiere de sistem, dintre care cel mai important este fisierul **Master File Table (MFT)**. Acest fisier poate fi privit ca o baza de date a sistemului deoarece contine informatii despre toate celelalte fisiere si directoare de pe partitia NTFS.

Figura 1^[2] prezinta structura unui volum NTFS la terminarea formatarii. Conform acestei figuri, **sectorul de boot** sau sectorul 0 este prima locatie pe o partitie NTFS. De asemenea, pentru programul de boot-are (programul de pornire al sistemului) sunt rezervate si sectoarele de la 1 la 16.

Fisierul **MFT** este primul fisier pe volumul NTFS. Pentru toate celelalte fisiere exista o intrare in MFT, chiar si pentru el insusi. Asa cum am mentionat mai sus, MFT se comporta ca o baza de date a sistemului deoarece pastreaza toate informatiile despre un fisier, cum ar fi: numele, dimensiunea, informatii de timp referitoare la fisier, permisiuni si date efective. Toate informatiile de mai sus sunt considerate atribute ale fisierului, prin urmare MFT este tratat ca fiind o colectie de atribute. „Un atribut este o secventa de octeti organizati in doua componente: componenta de descriere a atributului (header) si continutul sau. ” De obicei, atunci cand dimensiunea lor permite sa fie memorate in intrarea corespunzatoare din MFT, atributele de fisier sunt pastrate in MFT. Cu toate acestea, ele mai pot fi pastrate si in zone auxiliare de pe HDD, exterioare fisierului MFT.

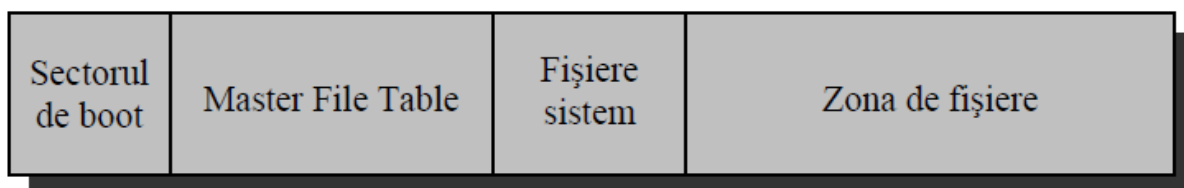


Figura 1. Structura unui volum NTFS^[2]

Atributele de fisier sunt folosite intern de catre NTFS, prin urmare utilizatorul nu are acces la ale si nu poate defini noi tipuri de atribute. In tabelul de mai jos sunt prezentate toate tipurile de atribute definite in prezent de sistemul de fisiere NTFS. Pe de alta parte, lista este totusi extensibila, in sensul ca in viitor vor putea fi definite si alte tipuri de atribute de fisier.

Tabelul 1. Tipuri de atribute ale fisierelor in NTFS^[3]

Tipul atributului	Descriere
<i>Standard information</i>	Include informații cum ar fi informații de timp și numărul de legături.
<i>Attribute Lists</i>	Listează locațiile tuturor înregistrărilor atributelor non-rezidente.
<i>File Name</i>	Un atribut care se poate repeta atât pentru denumiri scurte, cât și pentru denumiri lungi de fișiere. Numele lung al fișierului poate fi de până la 255 de caractere Unicode. Numele scurt este în format 8.3. Nume adiționale sau hard link-uri, necesitate de POSIX, pot fi incluse ca atribute de nume adiționale ale fișierului.
<i>Security Descriptor</i>	Denumeste proprietarul fișierului și utilizatorii care îl pot accesa.
<i>Data</i>	Conține datele din fișier. NTFS permite atribute multiple de date pentru fiecare fișier. Fiecare fișier are întotdeauna un atribut implicit de date.
<i>Object ID</i>	Un identificator unic în volum și utilizat de facilitatea de regăsire a legăturilor distribuite. Nu toate fișierele au identificatori de obiect.
<i>Logged Tool Stream</i>	Similar unui flux de date, dar operațiile sunt înscrise în fișierul log al NTFS întocmai ca și modificările de metadata. Folosit de EFS.
<i>Reparse Point</i>	Folosit pentru puncte de montare de pe disc și de asemenea și de drivere de filtrare ale IFS (<i>Installable File System</i>) pentru a marca anumite fișiere ca fiind speciale pentru acel driver.
<i>Index Root</i>	Folosit pentru a implementa directoare și alți indecși.
<i>Index Allocation</i>	Folosit pentru a implementa directoare și alți indecși.
<i>Bitmap</i>	Folosit pentru a implementa directoare și alți indecși (directoare f. mari)
<i>Volume Information</i>	Folosit doar de fișierul sistem \$Volume. Conține versiunea volumului.
<i>Volume Name</i>	Folosit doar de fișierul sistem \$Volume. Conține eticheta volumului.

NTFS se bazeaza pe principiul „totul este fisier”. Astfel el foloseste pentru accesul si managementul fisierelor structuri de date numite fisiere **metadata**.

3. Fisiere cu seturi multiple de date (realizat de Mavru Anca)

Asa cum putem observa in primul capitol, unul dintre avantajele pe care le ofera NTFS este lucrul cu fisiere cu seturi multiple de date. Aceasta facilitate permite ca unele date din fisier sa fie accesate ca o unitate separata.

Pentru a intelege mai bine acest proces, putem lua ca exemplu o aplicatie grafica care stocheaza pentru un fisier de tip imagine o versiune de calitate mai slaba, dar de dimensiune mult mai mica a imaginii (thumbnail) intr-un alt set de date al fisierului, diferit de cel care contine imaginea propriu-zisa.

Un set alternativ de date al unui fisier este tratat separat de NTFS, el putand fi identificat printr-un nume de forma: **nume_fisier:nume_set_alternativ**. Dimensiunile seturilor alternative de date ale unui fisier nu sunt evidentiata in dimensiunea acestuia. De asemenea, seturile de date alternative au aceleasi permisiuni pe care le are si fisierul propriu-zis.

Un set de date alternativ poate fi creat in doua moduri:

- prin apelul functiei **CreateFile**;
- din linia de comanda.

Mai jos avem un exemplu^[4] de creare de set de date alternativ din linia de comanda:

```
„echo „Setul principal de date”>Fisier.txt”
```

Prin aceasta comanda a fost creat un fisier cu numele Fisier.txt. Pentru a crea in acest fisier un set alternativ cu numele ADS folosim comanda:

```
„echo „Set alternativ de date”>Fisier.txt:ADS”
```

Observam ca setul adaugat nu apare intre fisierele din director si nici nu influenteaza dimensiunea fisierului principal.

Prin executia comenzilor urmatoare se citeste continutul setului principal si al celui alternativ:

```
„more<Fisier.txt”
```

```
„more<Fisier.txt:ADS”
```


Pentru a deschide acest set alternativ într-un editor de texte, de exemplu Word, numele setului trebuie sa contina o extensie, de exemplu: **Fisier.txt:ADS.doc**. Astfel el poate fi vizualizat si editat in editor prin comanda „**word Fisier.txt:ADS.doc**”.

In cazul in care seturile alternative de date contin fisiere executabile, acestea se executa cu comanda „**start./Fisier.txt:fis.exe**”.

Un posibil dezavantaj al folosirii seturilor alternative de date este protectia scazuta la virusi. Seturile alternative ofera virusilor posibilitatea de a se ascunde deoarece ele nu vad in lista de fisiere si nu modifica dimensiunea si marca de timp a fisierului principal.

Deoarece Windows nu ofera programe utilitare care sa detecteze seturile alternative de date si in mod implicit problemele acestora, cei de la Microsoft au creat un astfel de program utilitar care se numeste **Streams**.

4. Criptarea si compresia NTFS (realizat de Ionescu Liviu)

4.1. Criptarea

NTFS utilizeaza tehnologia de criptare **Encrypting File System (EFS)** pentru a memora fisierele criptate pe discuri. EFS este folosit si pentru protectia fisierelor deoarece interzice accesul neautorizat, cu ajutorul cheilor de criptare simetrice si publice. Utilizatorii EFS primesc un certificat digital cu o pereche de chei, una publica si una privata.

Criptarea este transparenta pentru utilizatorul autorizat. In schimb, daca un utilizator neautorizat incearca orice tip de operatie asupra fisierului, primeste un mesaj „Access denied”.

NTFS foloseste urmatoarele attribute specifice pentru criptare si decriptare:

- Owner SID – identificatorul de securitate al proprietarului fisierului;
- Group SID – identificatorul de securitate al grupului;
- Discretionary access control list (DACL) – specifica cine are acces la fisier;
- System access control list (SACL) – specifica ce operatie a carui utilizator a fi memorata în jurnalul de audit al securitatii.

Asa cum am mentionat in capitolele anterioare, o lista de control a accesului (ACL) este formata dintr-un antet si mai multe structuri ACE (intrari de control ale accesului). Totusi, este posibil ca ACL sa nu contina nici un ACE. In acest caz, nici un utilizator nu are acces la fisier.

De asemenea, daca DACL nu are nici o lista de acces, nimeni nu are acces la fisier. Im mod reciproc, daca DACL nu este prezent in descriptorii de securitate, fisierul poate fi accesat de orice utilizator.

4.2. Compresia

Obiectele NTFS sunt comprimate folosind un algoritm modificat LZ77. Principiul acestui algoritm este cautarea, într-un bloc de date, de sub-siruri care se repeta. Blocul este comprimat înlocuind sub-sirurile cu o referinta la prima aparitie a sub-sirului.

In practica se pot intalni trei situatii:

1. Datele din bloc sunt toate 00h. Acesta este memorat ca un bloc special (*sparse block* – bloc suprapus). In acest caz intrarea in lista blocului de date are lungimea X si LCN -1.
2. Datele din bloc sunt necomprimate. Situatia se intalneste cand compresia nu ar reduce dimensiunea blocului. Acest caz se recunoaste prin faptul ca intrarea in lista blocului de date are lungimea X si LCN ≥ 0 .
3. Datele din bloc sunt comprimate. Acest caz se recunoaste prin faptul ca intrarea in lista blocului de date are lungimea $L \leq X$ si LCN ≥ 0 .

5. Bibliografie

1. ^[1] **Capitolul 4.3.**, ADRIAN COLEȘA, IOSIF IGNAT, ZOLTÁN SOMODI– „Sisteme de Operare– chestiuni teoretice și practice”; Ed. U.T. Press, Cluj–Napoca, 2007;
2. ^[2] **Capitolul 4.2.**, ADRIAN COLEȘA, IOSIF IGNAT, ZOLTÁN SOMODI– „Sisteme de Operare– chestiuni teoretice și practice”; Ed. U.T. Press, Cluj–Napoca, 2007;
3. ^[3] **Capitolul 4.2.**, ADRIAN COLEȘA, IOSIF IGNAT, ZOLTÁN SOMODI– „Sisteme de Operare– chestiuni teoretice și practice”; Ed. U.T. Press, Cluj–Napoca, 2007;
4. ^[4] **Capitolul 4.5.**, ADRIAN COLEȘA, IOSIF IGNAT, ZOLTÁN SOMODI– „Sisteme de Operare– chestiuni teoretice și practice”; Ed. U.T. Press, Cluj–Napoca, 2007;