

Előadás_#13.

1. Az NT alapú operációs rendszerek fájlrendszere az NTFS

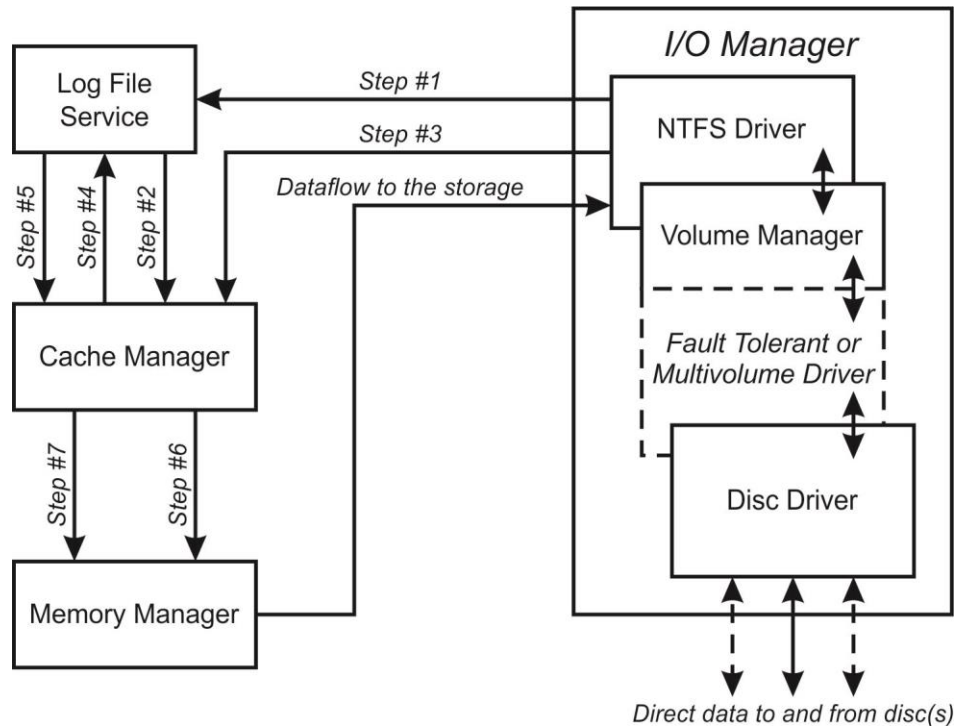
Korlátok a korábbi fájlrendszerek és az NTFS összehasonlításában:

- FAT12 alatt a fájl név 8.3 szerkezetű, egy fájl maximális mérete nem lehet több mint 32MB. A maximális partíció mérete is 32MB. A fájlok maximális száma egy partíción 4077db.
- FAT16 alatt a fájl név 8.3 szerkezetű. A maximális partíció mérete alapvetően 2GB (de pl. XP alatt 4GB). A fájlok maximális száma egy partíción 65.517db.
- A FAT32 támogatja a hosszú (255 karakter LFN / Long Filename) fájlneveket. Egy fájl maximális mérete nem lehet több mint 4GB mínusz 1Bájt. A maximális partíció mérete 2TB. A fájlok maximális száma egy partíción 268.435.427db.
- Az NTFS támogatja a hosszú (255 karakter) fájlneveket. Egy fájl maximális mérete nem lehet több mint 16TB. A maximális partíció mérete 256TB. A fájlok maximális száma egy partíción $2^{32}-1$ db. A korábbi (Microsoft által is támogatott) fájlrendszerek a „Convert” parancs segítségével NTFS-re konvertálhatók.

[pl.: Az „F:” meghajtót ez a parancs konvertálja FAT32-ről NTFS-re egy CMD ablakban: `convert f: /FS:NTFS`]

Az NTFS fájlrendszer működéséhez a három alapvető komponens az NTFS Driver, a Log File Service és a Cache Manager. A következő ábra nem csupán ennek a három komponensnek az együttműködését mutatja be, hanem bővebb áttekintést is enged, mivel a réteges felépítésű struktúra további opcionális rétegek (az ábrán szaggatott téglalapban) beépülését is lehetővé teszi. Az opcionális rétegek jellemzően többlet funkciókat valósítanak meg. További komponens még az adatok épségéért felelős Volume Manager, és a fizikai tárolókkal a közvetlen kapcsolatot biztosító Disc Driver.

Valós életben létező opció a szoftveres RAID támogatás, amely több lemez használatával képes a tárterületet és/vagy az adatbiztonságot illetve a sebességet is megnövelni. Például RAID 0 használata esetén több kisebb fizikai tároló összekapcsolásával egy nagy tárolót lehet kialakítani, RAID 1 használata esetén pedig az adatok duplikált tárolását teszi a rendszer lehetővé. A magasabb RAID szintek (1, 5, 6, 10, 50, 60) az adatbiztonság emelése mellett, az olvasás párhuzamosításával még sebesség növelést is biztosítanak.

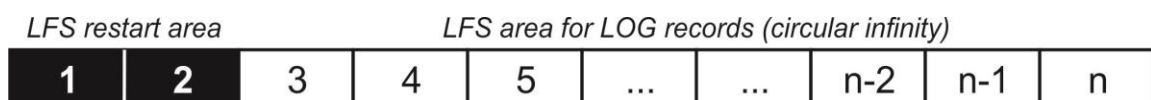


A fájl írási művelet kiszolgálásának lépései (Step #1-7):

1. Az NTFS Driver jelzi az LFS-nek, hogy fájl írás művelet következik, és egyben utasítja is, hogy készítsen a tranzakcióról egy úgynevezett „log rekord”-ot.
2. Az LFS hozzáír egy rekordot a cache-ben lévő log fájlhoz.
3. Az NTFS Driver a fájlt beírja a CM-be. (Nagyméretű fájlok nem írhatók ki egy lépésben, hanem csak a tároló blokkméretének megfelelő műveletenként.)
4. A CM jelzi, hogy az írás befejeződött, minden adat továbbításra vár.
5. Az LFS meghatározza, hogy melyik adatokat kell a cache-ből továbbítani: a log fájlt és a megváltoztatott teljes fájlt (vagy annak csak az aktuális blokkját).
6. A CM (a többi komponens segítségével) a lemezre írja a log fájlt.
7. A CM (a többi komponens segítségével) a lemezre írja a megváltoztatott fájlt (vagy annak csak az aktuális blokkját).

A log fájl szerkezete

A log file egy körkörös szervezésű tömb, mely egyrészt a fájlrendszer újra indításához szükséges információkat tartalmazza két kitüntetett rekordban – azaz két példányban – illetve további rekordjaiban az elvégzett tranzakciókról tartalmaz információkat. Ezen információk segítségével nyílik lehetőség egyes tranzakciók újbóli elvégzésére, illetve visszavonására. A megszakadt, illetve le nem zárt tranzakciók sikeres befejezését is közel száz százalékos valószínűséggel oldja meg az NTFS fájlrendszer.



Az NTFS fájlrendszer további előnyei

- Az alternatív, vagy többszörös adatfolyamok (ADS / Alternative Data Stream) teszik lehetővé azt, hogy egy fájl több adatfolyamhoz is kapcsolódhasson. Például a "001.txt" nevű fájl "001.txt:secret.txt" néven egy alternatív adatfolyamot is magába foglalhat. Az ADS név ismeretének hiányában ez az extra tartalom nem érhető el. Az alternatív adatfolyamok jelenlétére a fájl mérete közvetlenül nem is utal, így a szabad tárhelyet észrevétlenül csökkenti. Az alternatív adatfolyam elveszik, amennyiben az azt tartalmazó fájlt egy nem ADS képes fájlrendszer (pl. FAT32) használó tárolóra másoljuk.
- Unicode fájl nevek használata.
Maximum 255 karakter hosszúak lehetnek a fájlnevek – akár a pont és a szóköz karakterek is használhatók – a kis- és nagybetűket a fájlrendszer megkülönbözteti.
- Az indexelési lehetőség segítségével az adott tulajdonságú fájlok közvetlenül érhetők el.
- Dinamikus hibás szektor kezelés, menet közbeni tiltással. Amennyiben egy szektor hibássá válik, a rendszer azt a későbbiekben nem fogja adattárolásra felhasználni. A meghibásodott szektor tartalmát – amennyiben az még olvasható – a rendszer egy másik, ép szektorba másolja át.
- A kvóták segítségével nyílik lehetősége a rendszer adminisztrátorának arra, hogy az egyes felhasználók számára egyedi méretű lemezterületeket jelöljön ki. A felhasználók így természetesen már nem a fizikai tároló méretét látják a szabad- illetve a foglalt helyek ellenőrzésekor, hanem csupán a számukra kiosztott részt.
- Az NTFS az LZ77 algoritmus (amely a ZIP fájl formátumban is használatos) használatával a fájlok tömörítésére is képes. A tömörítés, azaz a be- illetve kicsomagolás egyaránt erőforrás igényes művelet, azaz csak akkor célszerű élni ezzel a lehetőséggel, ha megfelelően erős és viszonylag „kihasználatlan” CPU áll ehhez rendelkezésre. Szerverek esetében – ahol a CPU-k terhelése a kliensek igényeinek megfelelően tág határok között változhat – nem célszerű a tömörítéssel további erőforrásokat elvonni a rendszertől.
- Fájlok illetve könyvtárak – a felhasználó szempontjából átlátszó – titkosítását teszi lehetővé az EFS (Encrypting File System) használata. Ez a képesség csak az 5.0 (azaz a Windows 2000) és az azt követő verziókban érhető el, a korábbi verziókban nem áll rendelkezésre. Természetesen ez esetben sem szabad megfeledkezni arról, hogy a titkosítás is jelentős CPU erőforrásokat képes lekötöni.

Adattípusok és adatszerkezetek

- A kötet (Volume) egy elsődleges háttértár (HDD vagy SSD) egy logikai partíciója. A kötet formázása tulajdonképpen a köteten használatos logikai fájlrendszer kialakítása.
- Az adattárolás alapegysége, a helyfoglalási egység, a szektorcsoport (Cluster) [ZBR esetén nyilván zónacsoport]. A csoportképzés formázáskor kerül beállításra, a kettő hatványai szerint.
- Logical Cluster Number (LCN)
Az LCN az egy kötetben nyilvántartott Clusterek sorszámait jelenti. A Cluster-eket 0-tól [az LCN 0 a kötet első Cluster-e, azaz a Boot Sector] n-ig folyamatosan számozzuk, az érték gyakorlati maximuma $2^{64}-1$. A rendszer így Cluster-ek sorozataként látja a meghajtót. Az LCN számot a Cluster méretével megszorozva kapjuk meg az adott LCN számhoz tartozó adat fizikai címét a meghajtón.
- Virtual Cluster Number (VCN)
A VCN az egy adott adatfolyamhoz (Stream) tartozó Cluster-ek számait jelenti, melyek így az adott fájlt reprezentálják. A VCN 0 az adott fájl első Cluster-ét jelenti. A VCN számok a potenciális töredezettség miatt nem feltétlenül mutatnak a fizikailag is egymást követő Cluster-ekre. Egy adott fájl fizikai címét a meghajtón a VCN és a LCN segítségével találja meg az NTFS fájlrendszer. Ehhez pontosan az LCN-ek VCN-ekhez történő hozzárendelését és nyilvántartását kell felhasználni, vagyis azt, hogy a fájl egy adott LCN-e, melyik VCN-el érhető el. A hozzárendelést természetesen fájlanként kell nyilvántartani.
- NTFS Metaadatok (Metadata)
Az NTFS Metadata a Master File Table rendszer szintű bejegyzéseinek az összefoglaló neve.

Master File Table (MFT)

A Master File Table azoknak a fájl rekordoknak sorozata, amelyek a fájlrendszer kezeléséhez, a benne tárolt fájlok eléréshez szükségesek. Az NTFS értelmezésében mindent, ami a lemezen található, azt fájlként kell definiálni és kezelni. A következő – az Master File Table-t bemutató – táblázatnak is értelem szerűen minden sora egy-egy fájlt jelent. A bejegyzésben szereplő fájlokat két kategóriába soroljuk: Metadata és Normal. A felhasználó közvetlenül csak a Normal kategóriához fér hozzá.

Egy fájl kezelésekor az alapadatokon, azaz a nevéen, kiterjesztésén és méretén túl a fájl egyéb jellegzetességei, tulajdonságai, azaz attribútumai (Attribute) is fontos szerepet töltenek be. Az NTFS fájlrendszerben egy fájl az attribútumai szerint lehet írásvédett fájl, rejtett fájl, archív fájl, rendszer fájl, tömörített fájl, index fájl, illetve az attribútumokban tárolódik a létrehozási és a módosítási idő is.

A hatékony, nem pazarló helykihasználás miatt a Master File Table nem az egész fájl rekordot, hanem annak csak az első 1kB-os darabját tartalmazza, így a Master File Table gyakorlatilag egy 1kB-os bejegyzéseket tartalmazó táblázat. A táblázat minden bejegyzése egy-egy fájl azonosít. A fájl egyedi azonosítója ezek után az a sorszám lesz, ami megmondja, hogy az Master File Table hányadik bejegyzése tartozik hozzá. A bejegyzések minden, a fájlra vonatkozó információt tartalmaznak, azonban szerkezetük nem kötött.

Az Master File Table első 16db Metadata bejegyzése a rendszerfájlok, melyek neve „\$” jellel kezdődik. Ezek a felhasználó elől rejtve vannak, ezeket az operációs rendszer használja. Ezt követi 8db jövőbeli fejlesztésre fenntartott, azaz előzetesen lefoglalt (Reserved) bejegyzés, majd ezek után következnek a felhasználói fájlok és könyvtárak, azaz a Normal fájlok listája.

Az ábrán csak a \$Root tartalma szerepel kibontva, de mindegyik Metadata bejegyzés saját adatszerkezettel és további egyedi attribútumokkal rendelkezik.

| Master File Table | | <i>Metadata</i> | <i>Function</i> |
|--------------------------|-------|-------------------------|--|
| | 0 | \$MFT | MFT, an index on every file |
| | 1 | \$MFT mirror | Part image of MFT (usually first 4 records) |
| | 2 | \$LogFile | Transactional logging file |
| | 3 | \$Volume | Serial number, creation time, dirty flag |
| | 4 | \$AttrDef | Attribute definitions |
| | 5 | \$Root | Root directory called '.' is an ordinary directory |
| | 6 | \$Bitmap | Volume's cluster map (free vs. in-use) |
| | 7 | \$Boot | Volume boot record |
| | 8 | \$BadClus | Bad cluster list on the volume |
| | 9 | \$Secure | Security descriptions used by the volume |
| | 10 | \$UpCase | For short and long filename comparison |
| | 11 | \$Extend | Directory for more metafiles |
| | 12 | \$Extend\\$\\$Reparse | Reparse Point file |
| | 13 | \$Extend\UsnJrnl | Log changing file |
| | 14 | \$Extend\Quota | Quote management file |
| | 15 | \$Extend\ObjId | Object ID file |
| | 16-23 | <i>Reserved</i> | <i>Reserved</i> |
| | 24- | User Files, Directories | User Files, Directories |
| | n-1 | User Files, Directories | User Files, Directories |
| | n | User Files, Directories | User Files, Directories |

| | |
|----------------------|----------|
| Standard information | |
| Filename | |
| Data stream | Extents |
| Attribute #1 | Resident |
| Attribute #2 | Extents |
| ... | ... |

Az attribútumok tárolási módjai

- Rezidens tárolás

Ebben az esetben közvetlenül az attribútum fejléce (Header) után tárolódik az attribútum értékét tartalmazó információ. Jellemzően a kisebb méretű attribútumok tárolódnak ezen a módon, melyek képesek a Master File Table rekordjába beférni.

Rezidens módon tárolt attribútum például normál esetben a fájlnev, az időbélyeg, illetve a biztonsági és a jogosultsági információk.

- Nem rezidens tárolás

Ebben az esetben az attribútum fejléce (Header) után csak az attribútum értékét tartalmazó adatok helye, pontosabban címe van tárolva. Jellemzően azok az attribútumok tárolódnak ezen a módon, melyek a nagy méretük vagy potenciális méretnövekedésük miatt eleve nem férnek be a Master File Table egy rekordjába, így tárolásukhoz további Cluster-eket kell felhasználni. A vonatkozó Cluster-ek nyilvántartásához egy indextáblára van szükség, mely a szükséges VCN-LCN hozzárendelést tartalmazza. A nem rezidens attribútum tárolási mód velejárója a tördelődés.

Nem rezidens módon tárolt attribútum használatos például a titkosított, kódolt, illetve tömörített állományok esetén.

Állományok illetve könyvtárak azonosítása egy 64-bites referenciaszámmal történik, mely két részből áll. Az első 48-bit mutatja meg azt, hogy hányadik Master File Table rekordról van szó, míg a következő 16-bit azt mutatja meg, hogy az adott állományt hányszor használták.

2. Hasznos linkek

[Mi az NTFS \(Microsoft, angol nyelvű\)](#)

[Hogyan működik az NTFS \(Microsoft, angol nyelvű\)](#)

[Az NTFS és a FAT típusú fájlrendszerek összehasonlítása \(Microsoft\)](#)