



# Operační systémy

3. ročník

---

Tomáš Michalek

*tomas.michalek@spsehavirov.cz*

1.3.2 (2022)

SPŠE Havířov

# **NTFS**

## **vnitřní struktura**

## 1. NTFS

Vznik a vývoj

Vlastnosti a organizace oddílu

Volume Boot Record

Master File Table

    Záznam

    Atributy

    Analýza příkladového záznamu

    MFT Zone

    MFTMirr

Systémové metasoubory

    \$LogFile

\$Secure

\$BadClus

\$Volume

\$Extend

\$AttrDef

\$UpCase

\$Bitmap

Kořenový adresář (.)

Alternativní datové toky (ADS)

Nevýhody NTFS

## 2. Cvičení

Úkol: Vytvoření a připojení VHD

Úkol: zobrazení vnitřní struktury NTFS

Úkol: zobrazení záznamu MFT

Úkol: Prozkoumání ADS

Bonusové úkoly

### 3. Kontrolní otázky

**NTFS**

---

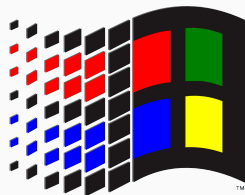
Během 90. let společnost *Microsoft* začala silně pociťovat jednu z největších slabin svých operačních systémů *MS-DOS* a *Windows 3.x* – závislost na souborové systému **FAT**.

Microsoft si kladl za cíl vytvořit vysoce kvalitní, výkonný, spolehlivý a bezpečný operační systém tak, aby byl schopen plně konkurovat operačním systémům jako Unix<sup>1</sup>.

Systém FAT nedostačoval požadavkům na *správu, management a spolehlivost*, které na něho byly kladené při nasazení v rozsáhlých firemních prostředích. Proto se Microsoft z příchodem *Windows NT* rozhodl vytvořit zcela nový souborový systém, který už nevycházel z FAT. Tak se zrodil nový **NTFS**.

---

<sup>1</sup>Nutnost splnění standartu POSIX.



MICROSOFT  
WINDOWSNT

**Obrázek 1:** Logo Microsoft Windows NT z roku 1993.

## NTFS

### New Technology File System

V době kolem vzniku NTFS spolupracoval Microsoft z firmou *IBM* na jejich operačním systému *OS/2*. Společně vytvořili souborový systém *HPFS*<sup>2</sup> a po rozvázání spolupráce s IBM se některé poznatky z vývoje HPFS promítly právě do nově vzniklého NTFS<sup>3</sup>.

NTFS byl navržen tak aby měl následující vlastnosti:

- **Spolehlivost**
- **Bezpečnost** a řízení přístupů (**oprávnění**)
- Efektivní ukládání dat
- **Navýšení limitů** pro velikosti oddílů i souborů
- **Dlouhé názvy souborů**

---

<sup>2</sup>High Performance File System

<sup>3</sup>V té době politika Microsoftu nerada viděla věci, které nebyly vytvořené "přímo" Microsoftem. Proto většina názvů, které se v odvětví standardně používají se ve Windows jmenuje jinak.



Souborový systém NTFS postupně procházel vývojem, který reagoval na různé nové potřeby (jako například nástup SSD disků a s nimi spojená technologie TRIM).

Rok	Označení	Verze Windows	Vlastnost	Max. velikost
1993	NTFS 1.0	Windows NT 3.1	Žurnalování	2TB
1994	NTFS 1.1 <sup>a</sup>	Windows NT 3.5	Dlouhá jména FAT	2TB
1996	NTFS 1.2	Windows NT 3.51	Komprimované soubory, ACL	2TB
2000	NTFS 3.0	Windows 2000	Kvóty, šifrování, Reparse, ...	2TB
2001	NTFS 3.1	Windows Server 2003	Extended MFT, Stínové kopie, ...	256TB
2005		Windows Vista	Symbolické odkazy, ...	256TB
2006		Windows Server 2008	SMART reader, self-healing	256TB
2009		Windows Server 2008 R2	SSD TRIM, nativní VHD	256TB

**Tabulka 1:** Postupný vývoj NTFS

<sup>a</sup>Některá literatura označuje verze stejně jako verze kernelu zapsané v **NTFS.sys** (NTFS 4.0, NTFS 5.0, ...)

- NTFS využívá **transakce**
  - operace a interakce s souborovým systémem jsou rozdělené na několik **dílčích akcí**.
  - Např. zápis na disk je rozdělen na dílčí akce:
    - přenos dat do řadiče
    - vyhledání volného místa na disku
    - vlastní zápis dat
    - uložení informací o poloze zapsaných dat do tabulky logické struktury
  - Podstata transakce spočívá v tom, že se **bud' provede, nebo se neprovede vůbec**.
    - pokud dojde k havárii některého z kroků – neprovede se nic
    - nemůže tedy dojít např. ke ztrátě clusteru jako u FAT
- **Vylepšená správa dat**
  - **Není omezen počet složek v kořenu disku.**
  - Použití B-Stromu (b-tree) pro efektivní uložení dat.
  - Při vyhledávání je minimalizován počet přístupů na disk.

- **Žurnalování**

- všechny zápisy na disk se zároveň zaznamenávají do speciálního souboru tz. *žurnálu* (\$Logfile).
- Pokud uprostřed zápisu systém havaruje, je následně možné podle záznamů všechny rozpracované operace dokončit, nebo anulovat a tím systém uvést do konzistentního stavu.

- **Přemapování clusterů<sup>4</sup>**

- **Komprese dat**

- zakomponovaná přímo do NTFS
- transparentní pro operační systém a uživatele
- *Windows Explorer* zobrazuje komprimované soubory **modrou barvou** (ikona modrých šipek k sobě).

---

<sup>4</sup>Přemapování clusterů probíhá při zápisu.

- **Oprávnění pro složky a soubory**

- Popisují co může a nemůže uživatel provádět s daty ve složce (souvislost s právy uživatelů).

- **Přípojný bod svazků<sup>5</sup>**

- **Diskové kvóty**

- **Šifrování dat**

- zakomponovaná přímo na úrovni souborového systému
- transparentní pro operační systém a uživatele
- *Windows Explorer* zobrazuje šifrované soubory **zelenout barvou** (ikona zámku).

---

<sup>5</sup>Adresář se chová jako disk nebo disk se připojí jako adresář do diskové struktury.

Oddíl sformátovaný jako NTFS má několik zásadních vlastností:

- **Všechno je soubor.**
- Vše je na disku uloženo jako `little endian`
- Celý oddíl je dostupný pro data (soubory)<sup>6</sup>.
  - **Cluster 0** začíná na začátku oddílu.
- **Celý systém je řešen jako obří databáze**
  - ve které jeden záznam v ní odpovídá souboru
- Speciální soubory pro popis NTFS:
  - jejich názvy se začínají znakem `$`
  - označujeme je názvem **metasoubory** (metafiles)
  - *Windows Explorer* tyto soubory nezobrazuje
  - Zobrazit je můžeme například příkazem:  
`dir \ah <NAZEV>`



**Obrázek 2:** Oddíl NTFS

<sup>6</sup>Kromně souborů potřebných systému, které mají z pravidla stejné umístění.

- Prostor je rozdělen na **clustery** (podobně jako u FAT).
  - nejmenší adresovatelná jednotka z pohledu NTFS.
  - typicky se skládá z 2, 4, 8, 16 nebo 32 sektorů.
  - **maximálně 64kB**<sup>7</sup>
    - hledání rovnováhy mezi fragmentací oddílu a nevyužitým místem v clusteru.
- až 64-bitové adresování clusterů:
  - teoretická hranice až **16EB** při clusteru 64KB<sup>8</sup>.
  - pro disk s MBR je maximální velikost svazku 2TB
  - pro disk s GPT je aktuální maximální velikost 256TB (pro 64KB cluster)



**Obrázek 3:** Oddíl NTFS

<sup>7</sup>Windows 10 (1903) přidal nové velikosti: 128K až 2M

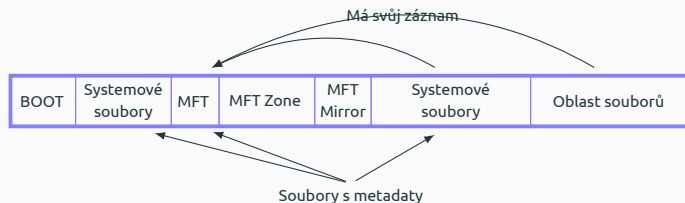
<sup>8</sup>tj. asi  $17 * 10^9$  TB

- logické číslování clusterů (**LCN**)
  - Offset na oddílu:  $Index_{clusteru} * Velikost_{clusteru}$
- clustery souboru identifikovány virtuálním číslováním (**VCN**):
  - číslovány sekvenčně od 0, ale ve skutečnosti **musí jít za sebou**.
  - součástí souboru mapování  $VCN -> LCN$ .



**Obrázek 4:** Oddíl NTFS

Naformatováním disku jako NTFS se prostor rozdělí podobně jak je tomu na obrázku 5.



**Obrázek 5:** Typická organizace oddílů NTFS



# NTFS – Volume Boot Record (\$Boot)

Při formatování oddílu NTFS, formátovací program na něm vyhradí (alokuje) prvních 16 sektorů (maximální velikost 8KB) pro metasoubor **\$Boot**, který slouží k zavedení systému. Boot sektor oddílu obsahuje dvě hlavní struktury:

- **BIOS parameter block**
  - Obsahuje informace o oddílu, název svazku, velikosti a umístění metasouborů.
- **Volume Boot Code**
  - Malý blok kódu programu obsahující instrukce pro zavedení systému.

## Záloha

Pro případ poškození je metasoubor \$Boot zálohovaný ještě na konci oddílu.



**Obrázek 6:** NTFS boot sektor

Ofset	Velikost	Popis	Hodnota
0x00	3	Instrukce skoku	0xEB'52'90
0x03	8	<b>OEM ID</b>	NTFS
0x0B	2	Byty v sektoru	512
0x0D	1	Sektorů v klastru	8
...			
0x28	8	Sektorů celkem	571391
0x30	8	<b>Start \$MFT</b>	3
0x38	8	<b>Start \$MFTMirr</b>	35711
...			
0x1FE	1	Konec sektoru	0x55'AA

**Tabulka 2:** Vybrané hodnoty ze souboru \$Boot



**Obrázek 7:** NTFS boot sektor

## MFT

### Master File Table

Každý **soubor** v oddílu NTFS **je reprezentován pomocí záznamu** ve speciálním souboru **\$MFT**. **První 16 záznamů je rezervováno** pro tabulku speciálních informací.

První záznam je popisuje samotnou tabulku MFT.



**Obrázek 8:** NTFS \$MFT

## Master File Table:

- vytváří se během formátování
- **každý soubor** (adresář) má zde svůj **vlastní záznam** (větu)
  - zaznamenává organizace dat v clusterech
  - velikost jednoho záznamu je pevná a obvykle **1KB (1024)**
  - malé soubory a složky<sup>9</sup> jsou uloženy plně v MFT jako je tomu na obrázku **12**.

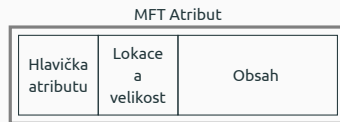


**Obrázek 9:** NTFS \$MFT

<sup>9</sup>Typicky menší jak 512 bytů

Každý záznam v MFT se skládá z:

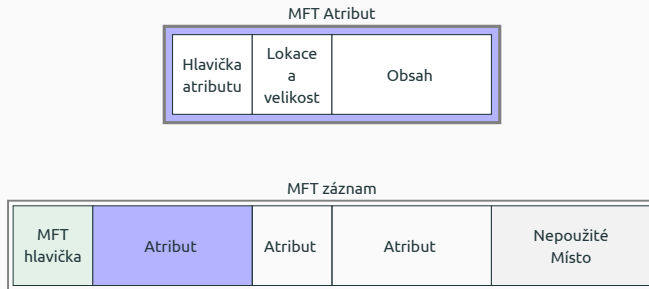
- **MFT hlavičky** - prvních **42** bytů
- **Atributů** - zbylé byty
  - Každý atribut se skládá z:
    - **hlavička** (16 bytů)
    - **umístění a velikost** obsahu (8 nebo  $56^{10}$  bytů)
    - **obsah** (různá velikost) - detaily atributu



**Obrázek 10:** Struktura atributu

---

<sup>10</sup>Pokud obsah je uložen v jiném místě oddílů (příznak "Non-Resident").



**Obrázek 11:** Struktura záznamu je složená z atributů a hlavičky

Typ	Název	Poznámka
0x00	Prázdné místo	
0x10	\$STANDARD_INFORMATION	základní informace
0x30	\$FILE_NAME	název souboru a jeho základní metadata
0x40	\$OBJECT_ID	unikátní GUID objektu
0x50	\$SECURITY_DESCRIPTOR	řízení přístupů
0x60	\$VOLUME_NAME	název oddílu
0x70	\$VOLUME_INFORMATION	verze a stav <sup>11</sup> oddílu
0x80	\$DATA	data souboru
0x90	\$INDEX_ROOT	informace o složce
		...
0xFFFF'FFFF	Konec atributů	

**Tabulka 3:** Vybrané atributy definované v \$AttrDef (viz Russon et al., 2005)

---

<sup>11</sup>Vlajka Dirty označuje špatné ukončení systému a nutnost provést chkdsk.

Díky tomu, že je malý soubor uložený přímo v \$MFT je přístupová doba k němu velmi rychlá<sup>12</sup>.

Složky jsou podobně jako soubory uložené v MFT, ale místo dat mají záznam informací o indexu. Podobně jako je tomu u malých souborů, malé záznamy složek mohou být plně uloženy v MFT.

\$MFT záznam				
(0x10) Standardní informace	(0x30) Název souboru nebo složky	(0x50) Bezpečnostní popisovač <sup>13</sup>	(0x80) Data nebo (0x90) index	...

**Obrázek 12:** Záznam v \$MFT pro malý soubor nebo adresář

<sup>12</sup>Pro porovnání ve FAT musíme nejdříve zjistit, zda-li soubor existuje a pak ho "poskládat" průchodem přes FAT alokační tabulku. U NTFS máme začátek dat k dispozici hned.

<sup>13</sup>Security descriptor



# NTFS – Příklad MFT záznamu

Offset	00 01 02 03 04 05 06 07	08 09 0A 0B 0C 0D 0E 0F	ASCII	Unicode
1F40E800	46 49 4C 45 30 00 03 00	10 C5 20 00 00 00 00 00	FILEÖ...Ä ....	..0. .
1F40E810	01 00 01 00 38 00 01 00	E0 01 00 00 00 04 00 00	...8...ä....	..8.Ä.E.
1F40E820	00 00 00 00 00 00 00 00	06 00 00 00 26 00 00 00	.....r...	.....&.
1F40E830	0B 00 00 00 00 00 00 00	10 00 00 00 60 00 00 00	.....	.....
1F40E840	00 00 00 00 00 00 00 00	48 00 00 00 18 00 00 00	.....H.....	.....H...
1F40E850	9C D5 92 59 D5 25 D8 01	69 F4 64 8C D5 25 D8 01	..Ö.YÖ%.iöd.Ö%.	..ü...ü
1F40E860	69 F4 64 8C D5 25 D8 01	25 A2 55 6D D7 25 D8 01	iöd.Ö%.%4umx%.	..ü...ü
1F40E870	20 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	.....
1F40E880	00 00 00 00 08 01 00 00	00 00 00 00 00 00 00 00	.....	..Ě.....
1F40E890	00 00 00 00 00 00 00 00	30 00 00 00 88 00 00 00	.....ö.....	.....0....
1F40E8A0	00 00 00 00 00 00 05 00	6E 00 00 00 18 00 01 00	.....n.....	.....n...
1F40E8B0	05 00 00 00 00 00 05 00	9C D5 92 59 D5 25 D8 01	.....Ö.YÖ%.	.....ü
1F40E8C0	9C D5 92 59 D5 25 D8 01	0E E3 26 5F D5 25 D8 01	..Ö.YÖ%..ä&_Ö%.	..ü...ü
1F40E8D0	9C D5 92 59 D5 25 D8 01	00 00 00 00 00 00 00 00	..Ö.YÖ%.....	..ü....
1F40E8E0	00 00 00 00 00 00 00 00	20 00 00 00 00 00 00 00	.....	.....
1F40E8F0	16 00 73 00 6F 00 75 00	62 00 6F 00 72 00 5F 00	..s.o.u.b.o.r..	..soubor..
1F40E900	70 00 72 00 6F 00 5F 00	61 00 6E 00 61 00 6C 00	p.r.o..a.n.a.l.	pro_anal
1F40E910	79 00 7A 00 75 00 2E 00	74 00 78 00 74 00 00 00	y.z.u...t.x.t..	yzu.txt.
1F40E920	40 00 00 00 28 00 00 00	00 00 00 00 00 00 04 00	@...(.....	@.(xxxx
1F40E930	10 00 00 00 18 00 00 00	8E 15 11 32 C7 91 EC 11	.....2C.i.	.....
1F40E940	BF 61 5C BA EF AB 21 6E	80 00 00 00 90 00 00 00	za\ei\ln.....	.....
1F40E950	00 00 18 00 00 00 01 00	72 00 00 00 18 00 00 00	.....r.....	.....r...
1F40E960	54 6F 74 6F 20 6A 65 20	73 6F 75 62 6F 72 2C 20	Toto je soubor,	.....
1F40E970	68 74 65 72 79 20 6D 61	74 65 20 7A 61 20 75 68	ktery mate za uk	..9.....
1F40E980	6F 6C 20 70 72 6F 7A 68	6F 75 6D 61 74 20 76 20	ol prozkoumat v	.....46
1F40E990	70 72 6F 67 72 61 6D 75	20 70 72 6F 20 61 6E 61	programu pro ana	.....
1F40E9A0	6C 79 7A 75 0D 0A 4E 54	46 53 2E 20 56 79 70 69	lyzu.NTFS. Vypi	.....
1F40E9B0	73 74 65 20 70 6F 7A 6E	61 74 68 79 20 68 20 6A	ste poznatky k j	.....
1F40E9C0	65 68 6F 20 7A 61 7A 6E	61 6D 75 20 76 20 4D 46	eho zaznamu v MF	.....56.
1F40E9D0	54 2E 00 00 00 00 00 00	FF FF FF FF 82 79 47 11	T.....ýýýý.yG.	.....
1F40E9E0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....	.....

Obrázek 13: Vybraný záznam fyzicky na disku

## NTFS – Příklad MFT záznamu

Name	Offset	Value
Signature (must be 'FILE')	000	FILE
Offset to the update sequence	004	0x30
Update sequence size in words	006	3
\$LogFile Sequence Number (LSN)	008	2 147 600
Sequence number	010	1
Hard link count	012	1
Offset to the first attribute	014	0x38
▼ Flags	016	01 00
In use	:0	1
Directory	:1	0
Real size of the FILE record	018	480
Allocated size of the FILE record	01C	1 024
Base FILE record	020	0
Next attribute ID	028	6
ID of this record	02C	38
Update sequence number	030	0B 00
Update sequence array	032	00 00 00 00
➤ <b>Attribute \$10</b>	<b>038</b>	
➤ <b>Attribute \$30</b>	<b>098</b>	
➤ <b>Attribute \$40</b>	<b>120</b>	
➤ <b>Attribute \$80</b>	<b>148</b>	
End marker	1D8	0xFFFFFFFF

**Obrázek 14:** Hlavička záznamu MFT

## NTFS – Příklad MFT záznamu

▼ Attribute \$10	038	
Attribute type	038	0x10
Length (including header)	03C	96
Non-resident flag	040	0
Name length	041	0
Name offset	042	0x00
▼ Flags	044	00 00
Compressed	:0	0
Encrypted	:14	0
Sparse	:15	0
Attribute ID	046	0
Length of the attribute	048	72
Offset to the attribute data	04C	0x18
Indexed flag	04E	0
Padding	04F	0
▼ \$STANDARD_INFORMATION	050	
File created (UTC)	050	19.02.2022 21:12
File modified (UTC)	058	19.02.2022 21:13
Record changed (UTC)	060	19.02.2022 21:13
Last access time (UTC)	068	19.02.2022 21:26
> File Permissions	070	20 00 00 00
Maximum number of versio...	074	0
Version number	078	0
Class Id	07C	0
Owner Id	080	0
Security Id	084	264
Quota Changed	088	0
Update Sequence Number	090	0

Obrázek 15: Atribut \$STANDARD\_INFORMATION

# NTFS – Příklad MFT záznamu

▼ Attribute \$30	098	
Attribute type	098	0x30
Length (including header)	09C	136
Non-resident flag	0A0	0
Name length	0A1	0
Name offset	0A2	0x00
▼ Flags	0A4	00 00
Compressed	:0	0
Encrypted	:14	0
Sparse	:15	0
Attribute ID	0A6	5
Length of the attribute	0A8	110
Offset to the attribute data	0AC	0x18
Indexed flag	0AE	1
Padding	0AF	0
▼ \$FILE_NAME	0B0	
Parent directory file record ...	0B0	5
Parent directory sequence n...	0B6	5
File created (UTC)	0B8	19.02.2022 21:12
File modified (UTC)	0C0	19.02.2022 21:12
Record changed (UTC)	0C8	19.02.2022 21:12
Last access time (UTC)	0D0	19.02.2022 21:12
Allocated size	0D8	0
Real size	0E0	0
► File attributes	0E8	20 00 00 00
(used by EAs and reparse)	0EC	0
File name length	0F0	22
File name namespace	0F1	0
File name	0F2	soubor_pro_analyzu.txt

**Obrázek 16:** Atribut \$FILE\_NAME

▼ File attributes	0E8	20 00 00 00
Read-Only	:0	0
Hidden	:1	0
System	:2	0
Archive	:5	1
Device	:6	0
Normal	:7	0
Temporary	:8	0
Sparse File	:9	0
Reparse Pointe	:10	0
Compressed	:11	0
Offline	:12	0
Not Content Indexed	:13	0
Encrypted	:14	0
Directory	:28	0
Index View	:29	0

**Obrázek 17:** Atribut \$FILE\_NAME – detail

## NTFS – Příklad MFT záznamu

▼ Attribute \$40	120	
Attribute type	120	0x40
Length (including header)	124	40
Non-resident flag	128	0
Name length	129	0
Name offset	12A	0x00
▼ Flags	12C	00 00
Compressed	:0	0
Encrypted	:14	0
Sparse	:15	0
Attribute ID	12E	4
Length of the attribute	130	16
Offset to the attribute data	134	0x18
Indexed flag	136	0
Padding	137	0
▼ \$OBJECT_ID	138	
GUID Object Id	138	8E 15 11 32 C7 91 EC 11 BF 61 5C BA EF AB 21 6E

Obrázek 18: Atribut \$OBJECT\_ID

## NTFS – Příklad MFT záznamu

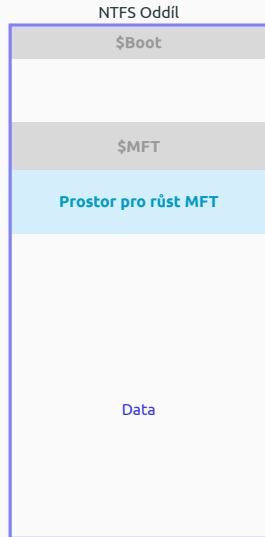
▼ Attribute \$80	148	
Attribute type	148	0x80
Length (including header)	14C	144
Non-resident flag	150	0
Name length	151	0
Name offset	152	0x18
▼ Flags	154	00 00
Compressed	:0	0
Encrypted	:14	0
Sparse	:15	0
Attribute ID	156	1
Length of the attribute	158	114
Offset to the attribute data	15C	0x18
Indexed flag	15E	0
Padding	15F	0
▼ \$DATA	160	
Data	160	54 6F 74 6F 20 6A 65 20 73 6F 75 62 6F 72 2C 20 6B 74 65 72 7...

Obrázek 19: Atribut \$DATA

Pro rychlejší a efektivnější práci systém (při formátování) **vyhradí prostor pro růst \$MFT souboru - \$MFT zone**. Zpravidla šlo o **12.5%** celkové kapacity, ale lze toto nastavení změnit (typicky na 25% nebo 50%).

Od *Windows Vista* bylo toto nastavení z důvodu kapacit disku nepraktické, a tak systém alokuje pro *\$MFT zonu* **200MB prostor**, který může být umístěn kdekoliv na disku.

Neznamená to, že by do takto vyhrazeného prostoru nešlo zapsat data souborů, ale je to vodítko pro systém, aby tak učinil co nejpozději s ohledem na volné místo v jiných částech disku.



**Obrázek 20:** MFT může růst díky prealokovanému prostoru

Druhým záznamem v MFT je *MFT mirror record*, který odkazuje na soubor \$MFTMirr.

**Tento soubor slouží jako záloha prvních <sup>14</sup> záznamy pro případ korupce MFT.** (Zalohované jsou záznamy pro \$MFT, \$MFTMirr, \$LogFile and \$Volume)

Podobně jako v případě souboru \$MFT je i lokace \$MFTMirr uložena v *boot záznamu*.

**Soubor \$MFTMirr se naléza přibližně v polovině oddílu NTFS<sup>15</sup>.**



**Obrázek 21:** Umístění \$MFTMirr

<sup>14</sup>Některé zdroje chybně uvádějí 16, viz: Sedory, 2018

<sup>15</sup>Podobnou zálohu má i boot sektor, který má svou zálohu v logickém středu disku.



Inode	Název souboru	Popis
0	\$MFT	Index všech souborů (a složek)
1	\$MFTMirr	Záloha prvních 4 záznamů MFT
2	\$LogFile	Záznamy o transakcích (žurnalování)
3	\$Volume	seriové číslo, vytvoření, vlajka <i>dirty</i>
4	\$AttrDef	Definice atributů
5	.	kořenový adresář oddílu
6	\$Bitmap	Mapa klastrů oddílu (1 = používaný vs 0 = volný)
7	\$Boot	boot záznam oddílu
8	\$BadClus	seznam vadných klastrů na oddílu
9	\$Secure	bezpečností popisovače pro oddíl
		...

**Tabulka 4:** Systémové metasoubory (viz Russon et al., 2005)

Inode	Název souboru	Popis
		...
10	<b>\$UpCase</b>	tabulka mapování velkých a malých čísel pro vyhledávání <sup>16</sup>
11	<b>\$Extend</b>	složka (\$ObjId, \$Quota, \$Reparse, \$UsnJrnl)
12-15	Nepoužívané	vyhrazeno pro budoucí rozšíření (in-use)
16-23	Nepoužívané	
<Any>	<b>\$Extend\ \$Quota</b>	kvóty oddílů
<Any>	<b>\$Extend\ \$Reparse</b>	zkratky, přípojný body, ...
<Any>	<b>\$Extend\ \$UsnJrnl</b>	žurnalovací záznam šifrování

**Tabulka 5:** Systémové metasoubory - pokračování

---

<sup>16</sup>NTFS používá Unicode pro ukládání znaků. Mapování mezi nimi není tak jednoduché jako v případě ASCII.

- Jak již bylo zmíněno všechny přístupy k objektům na disku jsou v pohledu NTFS transakce.
- Soubor **\$LogFile** zaznamenává průběh transakcí:
  - obsahuje pouze detaily nedávných transakcí
  - omezení velikosti souboru – řešeno jako kruhový buffer
  - po naplnění souboru se přepíše první záznam nových záznamem
- cílem zajištění konzistence POUZE **systémových souborů**



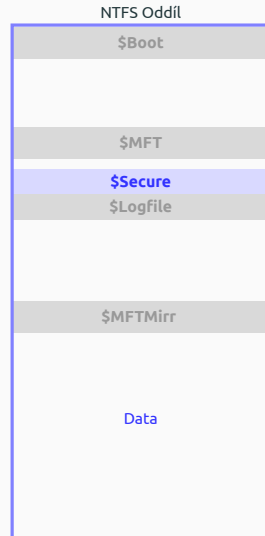
**Obrázek 22:** Umístění \$LogFile na disku

Systémový soubor **\$Secure** popisuje:

- **vlastníka** pro každý soubor (\$SSI - Standard Information ID)
- **pravidla přístupu** ACL<sup>17</sup> (\$SDH - Security Descriptor Hash)

## Okénko do minulosti

Ve *Windows NT* každý soubor měl vlastní atribut \$Security\_Descriptor, který ukládal tyto informace. Většina souborů ale měla stejné hodnoty tohoto atributu, proto se pro ně vyčlenil speciální soubor, aby nedocházelo k duplikaci dat.

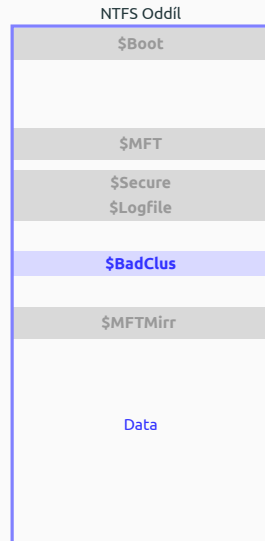


**Obrázek 23:** Umístění \$Secure na disku

<sup>17</sup>Access Control List

Soubor **\$BadClus sleduje vadné clustery** na disku.

- Cluster je označený jako vadný, pokud alespoň jeden sektor je vadný.
- **V souboru \$Bitmap se cluster označí jako používaný (1)**, aby zamezil budoucím pokusům o zápis na jeho místo.
- \$BadClus má velikost rozměru NTFS oddílu:
  - Soubor je „**sparse file**“.
  - Ve *sparse* souborech se spočítají 0, ale neukládají se fyzicky na disk.
- \$BadClus nezabírá místo na disku:
  - pokud cluster je označen jako špatný, data jsou místo toho zapsané v souboru \$BadClus se stejným offsetem, jako by tomu bylo na disku.



**Obrázek 24:** Umístění souboru \$BadClus na disku

Soubor **\$Volume** obsahuje informace o aktuálním NTFS oddíle:

- **název NTFS oddílu**
- číslo **verze** NTFS (1.x nebo 3.x)
- **vlajky** pro operace, které se mají provést při boot-u jako například:
  - **0x0001 (dirty)** - špatné ukončení relace.
  - 0x0002 - změň velikost logovacího souboru
  - upgraduj na novou verzi

## Příznak „dirty“

Příznak „dirty“ signalizuje OS, že minulý ukončení systému neproběhlo správně. Při příštím startu musí systém provést **chkdsk \f**.



**Obrázek 25:** Umístění \$Volume na disku

Složka **\$Extend** obsahuje volitelné metasoubory (extensions - rozšíření):

- **\$Quota** - metasoubor specifikující kvóty
  - **\$0** - záznamy o uživatelích, kteří mají nastavené kvóty (má vlastník souboru kvótu?)
  - **\$Q** - záznam pro všechny uživatele v systému (jak velká je kvóta?)
- **\$ObjId** - seznam všech souborů, které mají atribut \$Object\_ID:
  - Nejčastěji využívané pro MS Office dokumenty; odkazy v souborech tak mohou být přejmenované bez toho, aby dokument ztratil k nim přístup.



**Obrázek 26:** Umístění \$Extend na disku

- **\$Reparse** seznam všech *reparse* bodů:
  - umožňuje připojit část systému jako oddíl
- **\$UsnJrnl** zvaný „žurnál změn“:
  - krátkodobé logování změn v systému (přes jednotlivé aplikace)
  - podobný jako \$LogFile
  - využívá se k inkrementální zálohy, antivirový scan, ...
  - většinou se jeví jako prázdný – obsahuje data jen po dobu, kdy je potřebuje program



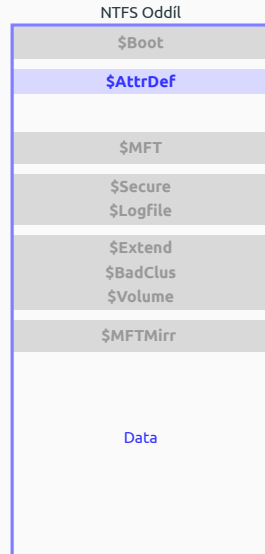
**Obrázek 27:** Umístění \$Extend na disku



Jak již bylo zmíněno, soubory a složky v NTFS se skládají z *atributů*. Jejich definice se nalézá v metasouboru **\$AttrDef**. (viz tabulka 3)

Pro každý atribut obsahuje několik informací:

- **název atributu** (attribute name)
- **ID**
- pravidla pro použití (v současnosti nepoužívané)
- **vlajky**
  - 0x02 - indexované (indexed)
  - 0x40 - vždy residentní (always resident)
  - 0x80 - může být neresidentní (non-resident)
- **minimální velikost**
- **maximální velikost**

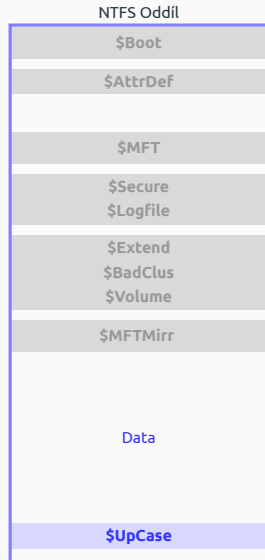


**Obrázek 28:** Umístění \$AttrDef na disku

Metasoubor **\$UpCase** umožňuje efektivní třídění a vyhledávání v NTFS. Pro práci s různými jazykovými sadami (code pages), ukládá NTFS názvy souborů ve znakové sadě *Unicode*. Soubor *\$UpCase* tedy obsahuje všechny *velká* písmena pro zakovou sadu Unicode (128KB).

### Dle čeho třídíme?

Samotný soubor má vždy ponechané velikosti písmen, ale názvy souborů jsou převedene na VELKE\_PISMENA pro třídění, když se pro ně vytváří záznam do složky.



**Obrázek 29:** Umístění \$UpCase na disku

# NTFS – \$Bitmap

Metasoubor **\$Bitmap** popisuje aktuální stav alokace každého clusteru na oddíle. Jednotlivé bity pak mohou nabývat hodnot:

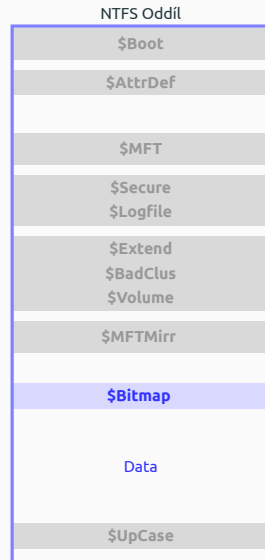
- **dostupný (0)**
- **alokovaný (1)**

## Defragmentace

Defragmentovací utility se snaží, aby jednotlivé byty byly 0x00 nebo 0xFF. Jiné hodnoty jsou vnímány jako „díry“, které se musí defragmentovat.

## Zarovnání

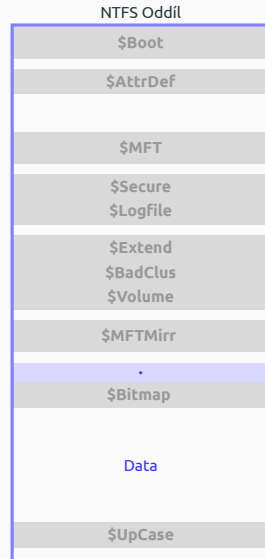
Protože soubor \$Bitmap je zarovnaný jako násobky 8 (ale oddíly ne) - proto na konci souboru může být sekce, která odpovídá prostoru „za oddílem“. Vždy má hodnoty 1 (alokovaný).



**Obrázek 30:** Umístění \$Bitmap na disku

Kořenový adresář **nemá**, na rozdíl od FAT, **omezení na počet položek v něm**. Obsahuje odkazy na soubory a podsložky na oddíle.

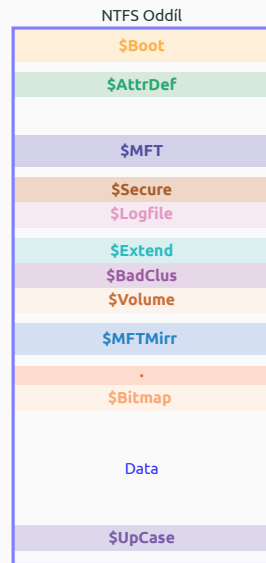
Je tvořen jako B-strom (balancovaný strom) tak, aby umožňoval efektivní ukládání, třídění a čtení informací.



**Obrázek 31:** Umístění kořenového adresáře

# NTFS – Systémové metadata – shrnutí

- popisují jednotlivé části NTFS systému
- jsou skryté před uživatelem
  - přístup k nim je umožněn „raw“ čtením disku
  - normální systémové API k nim nemá přístup
- až na několik výjimek (\$Boot) se mohou vyskytovat na různých oddílu (historické změny NT, XP, 7, 10)
- stále procházejí vývojem:
  - Např. \$Quote byl přesunut do \$Extend\Quote
  - Nové metadata jako \$Deleted, ...



**Obrázek 32:** Přehled metasouborů  
NTFS

# Alternativní datové toky (ADS)

Komně povinné částí v NTFS - kterou je zápis údajů o souboru do MFT (do souboru \$MFT) lze v NTFS uložit také další data - **alternativní datové toky**:

- nezobrazují se ve *Windows Explorer*
- v příkazové řádce je možnost je zobrazit příkazem `dir /r`
- jejich **velikost není započítaná** do velikosti **souboru**.
- při kopírování na FAT, posílání e-mailem a podobně je **kopírován pouze „standardní“ datový tok**.

## Čtení a zápis ADS (cmd)

1. `C:\>ECHO "moje_tajnostka" > muj_soubor.txt:psst`
2. `C:\>MORE < muj_soubor.txt:psst`

- **Fragmentace**

- NTFS od svého počátku trpí nepříjemnou vlastností – s oblibou fragmentuje soubory.
- Proto je od *Windows 7* implicitně nastavena defragmentace na 1x týdně.

- **Limitovaná komprese**

- NTFS není schopné komprimace pro klustry větší jak 4kB.

- **Neexistující oficiální dokumentace**

- Jako proprietární systém Microsoftu neexistuje celková dokumentace.
- Většina zásadních informací o vnitřím fungování NTFS byla získána zpětným inženýrstvím.

- **Omezená kompatibilita s jinými OS**

- Pro \*nixové systémy dostupný ovládač NTFS-3G.
- Apple MacOS má od verze 10.13 podporu pro čtení NTFS.

# Cvičení

---

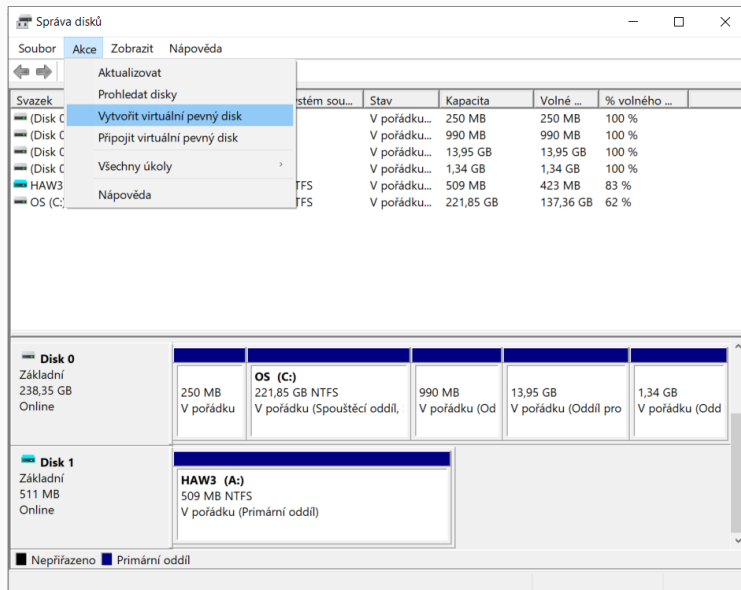


## Zadání

Vytvořte a připojte virtuální disk v systému Windows:

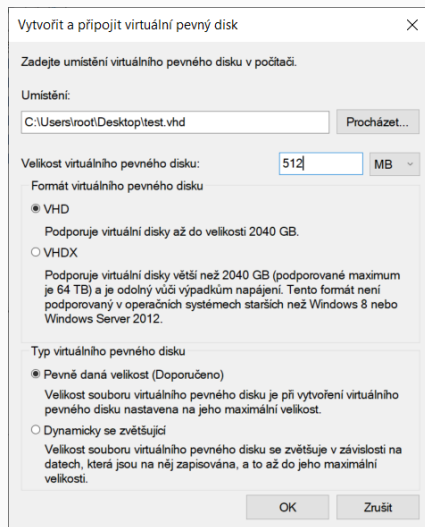
1. Otevřete program „**Správa disků**“ (Win + X).
2. Vytvořte (na ploše) virtuální disk „3{a,b}\_{prijmeni}.vhd“.
3. Nastavte parametry na:
  - Master Boot Record
  - Statická velikost (512MB)
  - Typ: VHD

# Úkol: Vytvoření a připojení VHD (řešení)



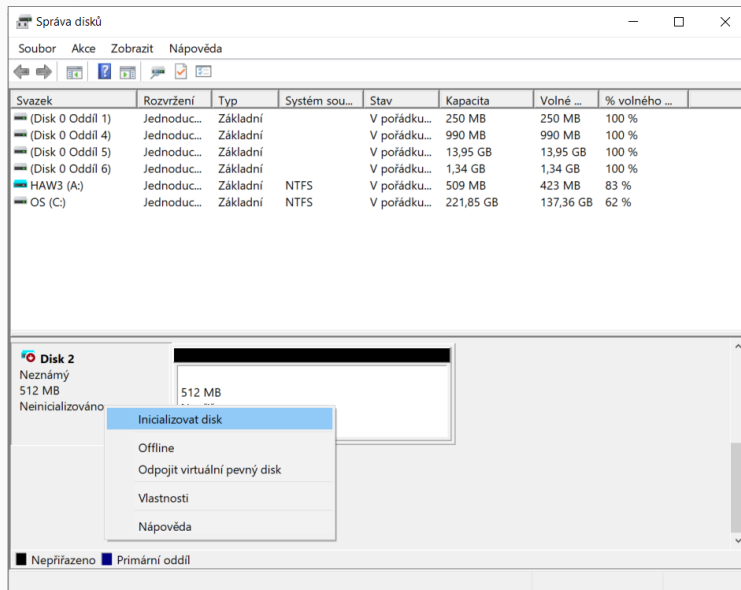
**Obrázek 33:** Zvolíme z menu „Akce“ vytvoření nového virtuálního disku

# Úkol: Vytvoření a připojení VHD (řešení II)



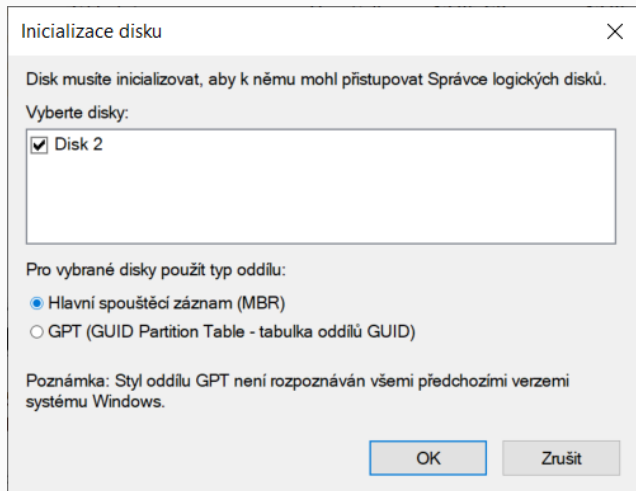
**Obrázek 34:** Zvolíme správné nastavení a umístění virtuálního disku

# Úkol: Vytvoření a připojení VHD (řešení III)



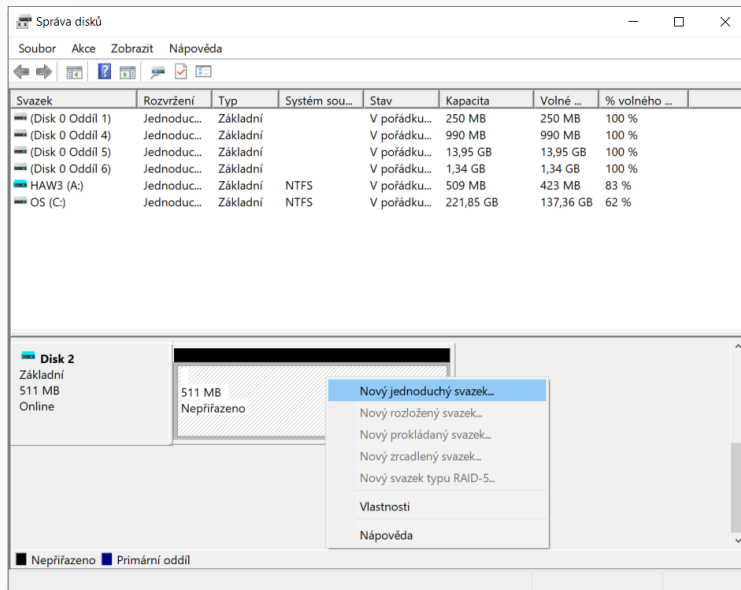
**Obrázek 35:** Inicializujeme virtuální disk

## Úkol: Vytvoření a připojení VHD (řešení IV)



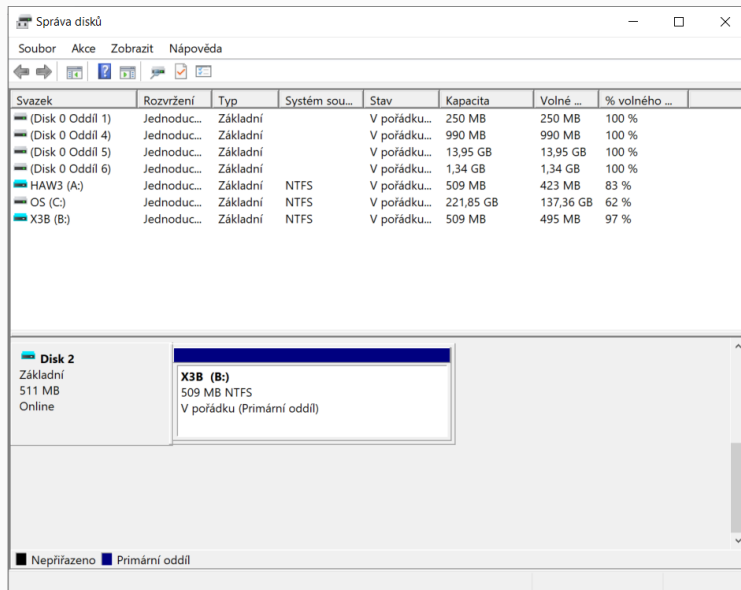
**Obrázek 36:** Inicializujeme virtuální disk

# Úkol: Vytvoření a připojení VHD (řešení V)



Obrázek 37: Naformátujeme virtuální disk

# Úkol: Vytvoření a připojení VHD (řešení VI)



Obrázek 38: Výsledkem je připojený virtuální disk

## Zadání

Získejte a prozkoumejte základní informace o vnitřní struktuře NTFS oddílu:

1. Připojte disk „`haw_3_task_analyse_ntfs.vhd`“ do systému.
2. Spusťte program „**Disk Editor for NTFS**“ a vyberte disk HAW3
3. Zobrazte MFT tabulku
4. Zobrazte obsah kořenového adresáře



# Úkol: zobrazení vnitřní struktury NTFS (řešení)

The screenshot displays the 'Runtime's DiskExplorer for NTFS' application window. The main pane shows a list of sectors, with the 'Valid Boot Sector' (Sector 128) selected. The details pane on the right shows the NTFS signature, bytes per sector, sectors per cluster, media descriptor, sectors per FAT, sectors per track, heads, and hidden sectors. The bottom status bar shows the drive path, volume, and region.

Runtime's DiskExplorer for NTFS

File Goto Link Edit View Tools Help

Sector Boot sector (NTFS)

x00000080 Valid Boot Sector

128 NTFS Signature: NTFS Physical drive #: x80 128  
Bytes per sector: x0200 512 Sectors in volume: x000000FE7FF 1042431  
Sectors per cluster: x08 8 1st MFT cluster: x0000A9AA 43434  
Media descriptor: xF8 248 1st MFT mirror cluster: x00000002 2  
Sectors per FAT: x0000 0 Clusters/file record: x000000FE 246  
Sectors per track: x003F 63 Clusters/index block: x00000001 1  
Heads: x00FF 255 Volume serial number: x74240A19 1949518937  
Hidden sectors: x000000000080 128

x00000081 Invalid Boot Sector

129 NTFS Signature: 0 0 T M Physical drive #: x00 0  
Bytes per sector: x4700 18176 Sectors in volume: x000000000000 0  
Sectors per cluster: x00 0 1st MFT cluster: x00000000 0  
Media descriptor: x00 0 1st MFT mirror cluster: x00000000 0  
Sectors per FAT: x0033 51 Clusters/file record: x00000000 0  
Sectors per track: x0030 48 Clusters/index block: x00000000 0  
Heads: x0M0 54272 Volume serial number: x00000000 0  
Hidden sectors: x00/124x0000 603979776

x00000082 Invalid Boot Sector

130 NTFS Signature: 2 12 8 Physical drive #: x66 102  
Bytes per sector: x0E87 3767 Sectors in volume: x040F00023C3E 14519995952083  
Sectors per cluster: x08 11 1st MFT cluster: x3A3E8366 977175398  
Media descriptor: x66 102 1st MFT mirror cluster: x886FFD30 2338782512  
Sectors per FAT: x96A3 3x\*43 Clusters/file record: x3E8B6607 1049323015  
Sectors per track: x6602 26114 Clusters/index block: xA166024A 2707817034  
Heads: x4F11 18081 Volume serial number: xEDE8022E 3991405102  
Hidden sectors: x125F0b36602 2809009497602

x00000083 Invalid Boot Sector

131 NTFS Signature: fPigJ@ Physical drive #: xC3 195  
Bytes per sector: x050F 34063 Sectors in volume: x040F00087883 14519995993101  
Sectors per cluster: x10 24 1st MFT cluster: x0A00367 130472903  
Media descriptor: x49 73 1st MFT mirror cluster: x067C3C0 2154283968  
Sectors per FAT: x6606 3x\*18 Clusters/file record: x6001C35 100670597  
Sectors per track: x1E51 2761 Clusters/index block: x6760661E 1734370846  
Heads: x6707 26119 Volume serial number: x10538D66 273911142  
Hidden sectors: x78x4x13FB8B 9829743932097

(Sector Offset)=x00000080-x00 (128 0) Selection=x00000080-x00-x00000080-x00

Drive: HD129: (2nd hard drive), 1 048 576 (x00100000) sectors 1st partition (NTFS) 509 MB, sectors 128-1 042 559

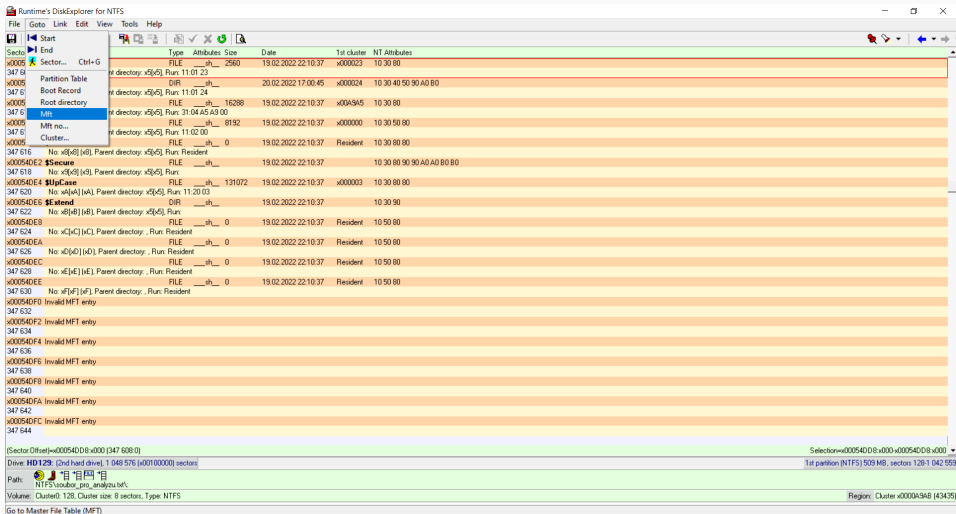
Path: NTFS

Volume: Cluster0: 128, Cluster size: 8 sectors, Type: NTFS Region: Boot sector

Memory in use: 788376 View: R/O [Unlicensed Evaluation Copy]

Obrázek 39: Úvodní obrazovka oddílu v aplikaci Disk Editor

# Úkol: zobrazení vnitřní struktury NTFS (řešení II)



Obrázek 40: Menu Disk Editoru

# Úkol: zobrazení vnitřní struktury NTFS (řešení III)

Runtime's DiskExplorer for NTFS

File Goto Link Edit View Tools Help

Sector	Name	Type	Attributes	Size	Date	1st cluster	NT Attributes
x00054D00	\$MFT	FILE	_____	262144	19.02.2022 22:10:37	x00A9AA	10 30 80 80
347 600	No: x[1][x0], Parent directory: x[5][5], Run: 31:40 AA A9 00						
x00054D02	\$MFTMir	FILE	_____	4096	19.02.2022 22:10:37	x0000002	10 30 80
347 602	No: x[1][x1], Parent directory: x[5][5], Run: 11:01 02						
x00054D04	\$LogFile	FILE	_____	4341760	19.02.2022 22:10:37	x00A581	10 30 80
347 604	No: x[2][x2], Parent directory: x[5][5], Run: 32:24 04 01 A5 00						
x00054D06	\$Volume	FILE	_____	0	19.02.2022 22:10:37	Resident	10 30 60 70 80
347 606	No: x[3][x3], Parent directory: x[5][5], Run: Resident						
x00054D08	\$AttrDef	FILE	_____	2560	19.02.2022 22:10:37	x0000023	10 30 80
347 608	No: x[4][x4], Parent directory: x[5][5], Run: 11:01 23						
x00054D0A		DIR	_____		20.02.2022 17:00:45	x0000024	10 30 40 50 90 A0 B0
347 610	No: x[5][x5], Parent directory: x[5][5], Run: 11:01 24						
x00054D0C	\$Bitmap	FILE	_____	16380	19.02.2022 22:10:37	x00A9A5	10 30 80
347 612	No: x[6][x6], Parent directory: x[5][5], Run: 31:04 A5 A9 00						
x00054D0E	\$Boot	FILE	_____	8192	19.02.2022 22:10:37	x0000000	10 30 50 80
347 614	No: x[7][x7], Parent directory: x[5][5], Run: 11:02 00						
x00054D10	\$BadClus	FILE	_____	0	19.02.2022 22:10:37	Resident	10 30 80 80
347 616	No: x[8][x8], Parent directory: x[5][5], Run: Resident						
x00054D12	\$Secure	FILE	_____		19.02.2022 22:10:37		10 30 80 90 90 A0 B0 B0
347 618	No: x[9][x9], Parent directory: x[5][5], Run:						
x00054D14	\$UpCase	FILE	_____	131072	19.02.2022 22:10:37	x0000003	10 30 80 80
347 620	No: x[A][xA], Parent directory: x[5][5], Run: 11:20 03						
x00054D16	\$Extend	DIR	_____		19.02.2022 22:10:37		10 30 90
347 622	No: x[B][xB], Parent directory: x[5][5], Run:						
x00054D18		FILE	_____	0	19.02.2022 22:10:37	Resident	10 50 80
347 624	No: x[C][xC], Parent directory: . Run: Resident						
x00054D1A		FILE	_____	0	19.02.2022 22:10:37	Resident	10 50 80
347 626	No: x[D][xD], Parent directory: . Run: Resident						
x00054DEC		FILE	_____	0	19.02.2022 22:10:37	Resident	10 50 80
347 628	No: x[E][xE], Parent directory: . Run: Resident						
x00054DEE		FILE	_____	0	19.02.2022 22:10:37	Resident	10 50 80
347 630	No: x[F][xF], Parent directory: . Run: Resident						
x00054DF0	Invalid MFT entry						
347 632							
x00054DF2	Invalid MFT entry						
347 634							
x00054DF4	Invalid MFT entry						
347 636							

[Sector Offset]=x00054D00:x000 [347 600:0]

Selection=x00054D00:x000 x00054D00:x000

Drive: HD 129: [2nd hard drive], 1 048 576 [x001000000] sectors

1st partition (NTFS) 509 MB, sectors 128-1 042 559

Path: NTFS\

Volume: Cluster0: 128, Cluster size: 8 sectors, Type: NTFS

Region: Cluster x0000A9AA [43434]

Memory in use: 790764 [Views: R/O [Unlicensed Evaluation Copy]

Obrázek 41: Metasoubory uložené v MFT

# Úkol: zobrazení vnitřní struktury NTFS (řešení III)

Runtime's DiskExplorer for NTFS

File Goto Link Edit View Tools Help

NTFS v5 DIR: \*\* \_\_th\_\_ modified 20.02.2022 17:00:45, starting at cluster x000024, Parent dir x5[5]

Interpretation of data:

Name	MIB	Size	Date	Attr	Subst	
\$MFT	x4[x4]	0				Save View
\$MFTMir	x6[x6]	0				Save View
\$RECYCLE.BIN	x6[x6]	0				Save View
\$Secure	x7[x7]	0				Save View
\$UpCase	x8[x8]	0				Save View
\$Volume	x9[x9]	0				Save View
\$uall	x10[x10]	16384	19.02.2022 22:10:37	_d_th__		Save View
key_pass_is_test.plx	x11[x11]	0				Save View
KFIATKY.D	x12[x12]	0				Save View
programy	x13[x13]	0				Save View
oltrovacka.bat	x14[x14]	0				Save View
oltrova_pro_analyzu	x15[x15]	0				Save View
soubor_2.bat	x16[x16]	0				Save View
soubor_ktery_ma_vel...	x17[x17]	0				Save View
soubor_premerovan...	x18[x18]	0				Save View
soubor_pro_analyzu.bat	x19[x19]	0				Save View
soubor_zacerny.bat	x20[x20]	0				Save View
System Volume Inf...	x21[x21]	0				Save View
tajny_soubor.bat	x22[x22]	0				Save View

Sector=x000540DA (347610)

Drive: HD 129: (2nd hard drive), 1 048 576 (x00100000) sectors

Path: NTFS\soubor\_pro\_analyzu.bat

Volume: Cluster0: 128, Cluster size: 8 sectors, Type: NTFS

Region: Cluster x00004948 (43435)

Memory in use: 833696 | View | R/O | Unlicensed Evaluation Copy

Obrázek 42: Kořenový adresář oddílu

## Zadání

Zobrazte a analyzujte obsah souborů na disku:

1. Připojte disk „haw\_3\_task\_analyse\_ntfs.vhd“ do systému.
2. Spusťte program „Active@ Disk Editor“ a vyberte disk HAW3
3. Najděte jednotlivé soubory z disku v MFT tabulce.

# Úkol: zobrazení záznamu MFT (řešení)

Active@ Disk Editor (Freeware)

File Edit Navigate View Window Help

Templates

NTFS MFT File Record

RA094.000

0:000

My Computer PhysicalDrive1 - Fixed Disk, SSD

Save Back Forward Edit Find Navigate Go to Offset Go to Sector

View ASCII Unicode

Offset	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ASCII	Unicode
1F4127E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
1F4127F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
1F412800	46	49	4C	45	30	00	03	00	FF	2E	21	00	00	00	00	00	FILE	FILE
1F412810	01	00	01	00	38	00	01	00	98	01	00	00	04	00	00	00	.	.
1F412820	00	00	00	00	00	00	00	00	04	00	00	36	00	00	00	00	.	.
1F412830	07	00	00	00	00	00	00	00	10	00	00	00	60	00	00	00	.	.
1F412840	00	00	00	00	00	00	00	00	48	00	00	00	18	00	00	00	.	.
1F412850	00	DB	85	DA	E3	25	D8	01	06	B8	40	C7	E3	25	D8	01	00	00
1F412860	30	90	BB	DA	E3	25	D8	01	00	DB	85	DA	E3	25	D8	01	0.	00
1F412870	20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.	.
1F412880	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00	.	.
1F412890	F0	05	00	00	00	00	00	00	30	00	00	00	98	00	00	00	0.	00
1F4128A0	00	00	00	00	00	02	00	00	7E	00	00	00	18	00	01	00	.	.
1F4128B0	05	00	00	00	00	05	00	00	00	DB	85	DA	E3	25	D8	01	00	00
1F4128C0	00	DB	85	DA	E3	25	D8	01	00	DB	85	DA	E3	25	D8	01	00	00
1F4128D0	00	DB	85	DA	E3	25	D8	01	00	00	00	00	00	00	00	00	.	.
1F4128E0	00	00	00	00	00	00	00	00	20	00	00	00	00	00	00	00	.	.
1F4128F0	1E	00	73	00	6F	00	75	00	62	00	6F	00	72	00	5F	00	.	.
1F412900	70	00	72	00	65	00	6A	00	60	00	65	00	6E	00	6F	00	p.	00
1F412910	76	00	61	00	6E	00	20	00	20	00	20	00	68	00	6F	00	v.	00
1F412920	70	00	69	00	65	00	2E	00	74	00	78	00	74	00	00	00	p.	00
1F412930	40	00	00	00	28	00	00	00	00	00	00	00	00	03	00	00	@.	00
1F412940	10	00	00	00	18	00	00	00	A3	19	11	32	C7	91	EC	11	@.	00
1F412950	BF	61	5C	BA	EF	21	6E	00	80	00	00	00	38	00	00	00	{	00
1F412960	00	00	18	00	00	00	01	00	1E	00	00	00	18	00	00	00	.	.
1F412970	73	6F	75	62	6F	72	20	6B	74	65	72	79	20	6A	65	20	s	00
1F412980	70	72	65	6A	6D	65	6E	6F	76	61	6E	79	00	0A	00	00	s	00
1F412990	FF	FF	FF	FF	82	79	47	11	00	00	00	00	00	00	00	00	y	00
1F4129A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.	.
1F4129B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.	.
1F4129C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.	.

Sector: 1 024 148 (0x0A094) Offset: 524 363 776 (0x1F412800) Read Only

Obrázek 43: Řešení úlohy „zobrazení záznamu MFT“

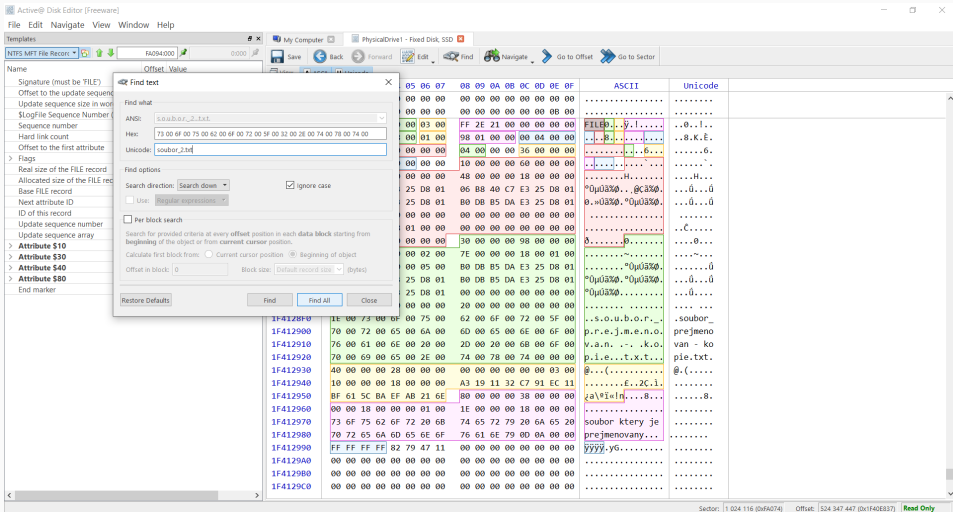
# Úkol: zobrazení záznamu MFT (řešení II)

The screenshot shows the Active@ Disk Editor interface. On the left, the 'NTFS MFT File Record' is displayed with fields like Name, Offset, and Value. The main area shows a hex dump of the MFT record data. A context menu is open over the hex dump, showing options like Undo, Redo, Copy, Paste, and Find. The 'Find' option is highlighted, and the search text 'soubor' is entered in the search bar.

Offset	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ASCII	Unicode
1F4127E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
1F4127F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
1F412800	46	49	4C	45	30	00	03	00	FF	2E	21	00	00	00	00	00	FILE	..y..
1F412810	01	00	01	00	38	00	01	00	98	01	00	00	04	00	00	00	..B...	..8.K.E.
1F412820	00	00	00	00	00	00	00	00	04	00	00	36	00	00	00	00	.....6..	.....6..
1F412830	07	00	00	00	00	00	00	00	10	00	00	00	60	00	00	00	.....	.....
1F412840	00	00	00	00	00	00	00	00	48	00	00	18	00	00	00	00	.....H...	.....H...
1F412850	00	0B	85	DA	E3	25	D8	01	06	B8	40	C7	E3	25	D8	01	*0u0a%0.0ca%	..ú...ú
1F412860	30	90	BB	DA	E3	25	D8	01	00	0B	85	DA	E3	25	D8	01	0.u0a%0.0u0a%	..ú...ú
1F412870	20	00	00	00	65	00	00	00	00	00	00	74	00	00	00	00	.....	.....
1F412880	00	00	00	00	08	01	00	00	00	00	00	00	00	00	00	00	.....	.....
1F412890	F0	05	00	00	00	00	00	00	30	00	00	00	98	00	00	00	.....p	.....p
1F4128A0	00	00	00	00	00	00	02	00	7E	00	00	18	00	01	00	00	.....	.....
1F4128B0	05	00	00	00	00	00	05	00	00	0B	85	DA	E3	25	D8	01	.....0u0a%0.	.....ú
1F4128C0	00	0B	85	DA	E3	25	D8	01	00	0B	85	DA	E3	25	D8	01	*0u0a%0.0u0a%	..ú...ú
1F4128D0	00	0B	85	DA	E3	25	D8	01	00	00	00	00	00	00	00	00	*0u0a%0.	..ú...ú
1F4128E0	00	00	00	00	00	00	00	00	20	00	00	00	00	00	00	00	.....	.....
1F4128F0	1E	00	73	00	6F	00	75	00	62	00	6F	00	72	00	5F	00	..s.o.u.b.o.r_	..soubor_
1F412900	70	00	72	00	65	00	6A	00	60	00	65	00	6E	00	6F	00	p.r.e.j.m.e.n.o.	prejmeno
1F412910	76	00	61	00	6E	00	20	00	20	00	60	00	6F	00	00	00	v.a.n. . . .k.o.	van - ko
1F412920	70	00	69	00	65	00	2E	00	74	00	78	00	00	00	00	00	p.i.e.t.x.t..	pie.txt.
1F412930	40	00	00	00	28	00	00	00	00	00	00	00	00	03	00	00	@...{(...	@.(...
1F412940	10	00	00	00	18	00	00	00	A3	19	11	32	C7	91	EC	11	.....E..2C.i	.....
1F412950	BF	61	5C	BA	EF	AB	21	6E	80	00	00	00	38	00	00	00	za*I=I...8..	.....8..
1F412960	00	00	18	00	00	00	01	00	1E	00	00	00	18	00	00	00	.....	.....
1F412970	73	6F	75	62	6F	72	20	68	74	65	72	79	20	6A	65	20	soubor který je	soubor který je
1F412980	70	72	65	6A	60	65	6E	6F	76	61	6E	79	00	0A	00	00	prejmenovany	prejmenovany
1F412990	FF	FF	FF	FF	82	79	47	11	00	00	00	00	00	00	00	00	yyyy.yG.....	.....
1F4129A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....	.....
1F4129B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....	.....
1F4129C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....	.....

Obrázek 44: Pravým klikem začneme hledání

### Úkol: zobrazení záznamu MFT (řešení III)



**Obrázek 45:** Názvy jsou uloženy jako Unicode (nastavíme parametry hledání)



# Úkol: zobrazení záznamu MFT (řešení IV)

Active@ Disk Editor [Freeware]

File Edit Navigate View Window Help

Templates

NTFS MFT File Record

RA094.000 01000

Name Offset Value

Signature (must be FILE) 000 FILE

Offset to the update sequence 004 0x30

Update sequence size in words 006 3

\$LogFile Sequence Number (LSN) 008 2 174 719

Sequence number 010 1

Hard link count 012 1

Offset to the first attribute 014 0x38

Flags 016 01 00

Real size of the FILE record 01C 1 024

Base FILE record 020 0

Next attribute ID 028 4

ID of this record 02C 54

Update sequence number 030 07 00

Update sequence array 032 00 00 00 00

Attribute \$10 038

Attribute \$30 098

Attribute \$40 130

Attribute \$80 158

End marker 190 0xFFFFFFFF

Find Results

# Offset Data

1 173865194 .....soubor\_2.txt....

2 173865394 .....soubor\_2.txt....

3 178014450 .....soubor\_2.txt....

4 520222954 .....soubor\_2.txt....

5 520223154 .....soubor\_2.txt....

6 524351730 .....soubor\_2.txt....

My Computer PhysicalDrive1 - Fixed Disk SSD

Save Back Forward Edit Find Navigate Go to Offset Go to Sector

View ASCII Unicode

Offset	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ASCII	Unicode
0A9C4810	01	00	01	00	38	00	01	00	A8	01	00	00	00	04	00	00	.....8.....	..8.z.Ě.
0A9C4820	00	00	00	00	00	00	00	00	05	00	00	00	2A	00	00	00	.....".	.....".
0A9C4830	1A	00	00	00	00	00	00	00	10	00	00	00	60	00	00	00	.....".	.....".
0A9C4840	00	00	00	00	00	00	00	00	48	00	00	00	18	00	00	00	.....H.....	.....H.....
0A9C4850	32	99	90	93	D5	25	D8	01	46	DE	6C	AC	D5	25	D8	01	2...0%0.Fp1-0%0.	...ů...ů
0A9C4860	46	DE	6C	AC	D5	25	D8	01	9B	EF	47	AF	D5	25	D8	01	Fp1-0%0..iG-0%0.	...ů...ů
0A9C4870	20	08	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....	.....
0A9C4880	00	00	00	00	08	01	00	00	00	00	00	00	00	00	00	00	.....Ě.....	.....Ě.....
0A9C4890	00	00	00	00	00	00	00	00	30	00	00	00	78	00	00	00	.....0..X..	.....0.X..
0A9C48A0	00	00	00	00	00	00	03	00	5A	00	00	00	18	00	01	00	.....Z.....	.....Z.....
0A9C48B0	05	00	00	00	00	00	05	00	32	99	90	93	D5	25	D8	01	.....2...0%0.	...ů...ů
0A9C48C0	32	99	90	93	D5	25	D8	01	32	99	90	93	D5	25	D8	01	2...0%0.2...0%0.	...ů...ů
0A9C48D0	32	99	90	93	D5	25	D8	01	00	00	00	00	00	00	00	00	2...0%0.	...ů...ů
0A9C48E0	00	00	00	00	00	00	00	00	20	00	00	00	00	00	00	00	.....	.....
0A9C48F0	0C	00	73	00	6F	00	75	00	62	00	6F	00	72	00	5F	00	..S.o.u.b.o.r..	..soubor..
0A9C4900	32	00	2E	00	74	00	78	00	74	00	00	00	00	00	00	00	2...t.x.t....	2.txt...
0A9C4910	40	00	00	00	28	00	00	00	00	00	00	00	00	00	04	00	@...((.....	@.(.....
0A9C4920	10	00	00	00	18	00	00	00	A3	15	11	32	C7	91	EC	11	.....E..2..i.	.....
0A9C4930	BF	61	5C	BA	EF	AF	21	6E	00	00	00	00	68	00	00	00	z\ E n...h...	.....h...
0A9C4940	00	00	18	00	01	00	01	00	4F	00	00	00	18	00	00	00	.....0.....	.....0.....
0A9C4950	54	65	6E	74	6F	20	73	6F	75	62	6F	72	20	6A	65	20	Tento soubor je	.....
0A9C4960	68	6F	6D	70	72	69	6D	6F	76	61	6E	79	2C	20	70	6F	komprimovany, po	.....
0A9C4970	7A	6E	61	63	74	65	20	6B	64	65	20	6A	65	20	74	61	znate kde je ta	.....
0A9C4980	74	6F	20	69	6E	66	6F	72	6D	61	63	65	20	70	6F	7A	to informace poz	.....
0A9C4990	6E	61	6D	65	6E	61	6E	61	20	76	20	4D	46	54	2E	00	namenana v MFT..	.....
0A9C49A0	FF	FF	FF	FF	82	79	47	11	00	00	00	00	00	00	00	00	yyyyy.yG.....	.....
0A9C49B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....	.....
0A9C49C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....	.....
0A9C49D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....	.....
0A9C49E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....	.....
0A9C49F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	1A	00	.....	.....

Sector: 347 684 (0x54E24) Offset: 178 014 430 (0xA9C48F2) Read Only

Obrázek 46: Ve výsledcích prohlédáme výsledky až najdeme záznam, který má hlavičku FILE

# Úkol: zobrazení záznamu MFT (řešení V)

Active@ Disk Editor [Freeware]

File Edit Navigate View Window Help

Templates

NTFS MFT File Record

Name Offset Value

Signature (must be FILE)	000	FILE
Offset to the update sequence	004	0x30
Update sequence size in words	006	3
\$LogFile Sequence Number (LSN)	008	2 174 719
Sequence number	010	1
Hard link count	012	1
Offset to the first attribute	014	0x38
> Flags	016	01 00
Real size of the FILE record	018	408
Allocated size of the FILE record	01C	1 024
Base FILE record	020	0
Next attribute ID	028	4
ID of this record	02C	54
Update sequence number	030	07 00
Update sequence array	032	00 00 00 00
> Attribute \$10	038	
> Attribute \$30	098	
> Attribute \$40	130	
> Attribute \$80	158	
End marker	190	0xFFFFFFFF

Find Results

#	Offset	Data
1	173865194	....soubor_2.txt....
2	173865394	....soubor_2.txtlnl..
3	178014450	....soubor_2.txt...0(
4	520222954	....soubor_2.txt....
5	520223154	....soubor_2.txtlnl..
6	524351730	....soubor_2.txt...0(

Set template offset at the cursor position

My Computer PhysicalDrive1 - Fixed Disk, SSD

Save Back Forward Edit Find Navigate Go to Offset Go to Sector

View ASCII Unicode

Offset	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ASCII	Unicode
0A9C47B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
0A9C47C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
0A9C47D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
0A9C47E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
0A9C47F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	18		
0A9C4800	46	49	4C	45	30	00	03	00	9D	82	20	00	00	00	00	00	FILE0.....	.0. .
0A9C4810																		
0A9C4820																		
0A9C4830																		
0A9C4840																		
0A9C4850																		
0A9C4860																		
0A9C4870																		
0A9C4880																		
0A9C4890																		
0A9C48A0																		
0A9C48B0																		
0A9C48C0																		
0A9C48D0																		
0A9C48E0																		
0A9C48F0																		
0A9C4900																		
0A9C4910																		
0A9C4920																		
0A9C4930																		
0A9C4940																		
0A9C4950																		
0A9C4960	6B	6F	6D	70	72	69	6D	6F	76	61	6E	79	2C	20	70	6F		
0A9C4970	7A	6E	61	63	74	65	20	68	64	65	20	6A	65	20	74	61		
0A9C4980	74	6F	20	69	6E	66	6F	72	6D	61	63	65	20	70	6F	7A		
0A9C4990	6E	61	6D	65	6E	61	6E	61	20	76	20	4D	46	54	2E	00		

Sector: 347 684 [0x5AE24] Offset: 178 014 208 [0xA9C4800] Read Only

Obrázek 47: Označíme začátek a spustíme šablonu

# Úkol: zobrazení záznamu MFT (řešení VI)

Active@ Disk Editor [Freeware]

File Edit Navigate View Window Help

Templates

NTFS MFT File Record

44E24000 0000

Name Offset Value

Signature (must be FILE) 000 FILE

Offset to the update sequence 004 0x30

Update sequence size in words 006 3

\$LogFile Sequence Number (LSN) 008 2 130 589

Sequence number 010 1

Hard link count 012 1

Offset to the first attribute 014 0x38

Offset to the first attribute 016 01 00

> Flags

Real size of the FILE record 018 424

Allocated size of the FILE record 01C 1 024

Base FILE record 020 0

Next attribute ID 028 5

ID of this record 02C 42

Update sequence number 030 1A 00

Update sequence array 032 00 00 00 00

> Attribute \$10 038

> Attribute \$30 098

> Attribute \$40 110

> Attribute \$80 138

End marker 1AD 0xFFFFFFFF

Find Results

# Offset Data

1 173865194 .....soubor\_2.txt....

2 173865394 .....soubor\_2.txtnel..

3 178014450 .....soubor\_2.txt...@.

4 520222954 .....soubor\_2.txtnel..

5 520223154 .....soubor\_2.txtnel..

6 524351730 .....soubor\_2.txt...@.

My Computer PhysicalDrive1 - Fixed Disk, SSD

Save Back Forward Edit Find Navigate Go to Offset Go to Sector

View ASCII Unicode

Offset	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ASCII	Unicode	
0A9C47E0	00	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F		
0A9C47F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
0A9C4800	45	49	4C	45	30	00	03	00	90	82	20	00	00	00	00	00	FILE	..	
0A9C4810	01	00	01	00	38	00	01	00	A8	01	00	00	04	00	00	00	..8.	..E.	
0A9C4820	00	00	00	00	00	00	00	00	05	00	00	2A	00	00	00	00	..	..	
0A9C4830	1A	00	00	00	00	00	00	00	10	00	00	00	60	00	00	00	..	..	
0A9C4840	00	00	00	00	00	00	00	00	48	00	00	00	18	00	00	00	..H.	..H.	
0A9C4850	32	99	90	93	D5	25	D8	01	46	DE	6C	AC	D5	25	D8	01	2...	..ô.	
0A9C4860	46	DE	6C	AC	D5	25	D8	01	9B	EF	47	AF	D5	25	D8	01	Fp1-	..ô.	
0A9C4870	20	08	00	00	00	00	00	00	00	00	00	00	00	00	00	00	..	..	
0A9C4880	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00	..	..	
0A9C4890	00	00	00	00	00	00	00	00	30	00	00	00	78	00	00	00	..0.	..X.	
0A9C48A0	00	00	00	00	00	03	00	00	5A	00	00	00	18	00	01	00	..Z.	..Z.	
0A9C48B0	05	00	00	00	00	05	00	00	3	Length of the	D5	25	D8	01	00	00	..2.	..ô.	
0A9C48C0	32	99	90	93	D5	25	D8	01	3	attribute: 90	D5	25	D8	01	00	00	2...	..ô.	
0A9C48D0	32	99	90	93	D5	25	D8	01	00	00	00	00	00	00	00	00	2...	..ô.	
0A9C48E0	00	00	00	00	00	00	00	00	20	00	00	00	00	00	00	00	..	..	
0A9C48F0	0C	00	73	00	6F	00	75	00	62	00	6F	00	72	00	5F	00	..s.o.u.b.o.r..	..soubor	
0A9C4900	32	00	2E	00	74	00	78	00	74	00	00	00	00	00	00	00	2...t.x.t.	2.txt	
0A9C4910	40	00	00	00	28	00	00	00	00	00	00	00	00	04	00	00	@...((	@.(	
0A9C4920	10	00	00	00	18	00	00	00	A3	15	11	32	C7	91	EC	11	.....E...2C.i.	.....	
0A9C4930	BF	61	5C	BA	EF	AB	21	6E	80	00	00	00	68	00	00	00	ja@i=ln...	...h.	
0A9C4940	00	00	18	00	01	00	00	00	4F	00	00	00	18	00	00	00	.....0.	.....0.	
0A9C4950	54	65	6E	74	6F	20	73	6F	75	62	6F	72	20	6A	65	20	Tento soubor je	.....	
0A9C4960	68	6F	6D	70	72	69	6D	6F	76	61	6E	79	2C	20	70	6F	komprimovany, po	.....	
0A9C4970	7A	6E	61	63	74	65	20	68	64	65	20	6A	65	20	74	61	znate kde je ta	.....	
0A9C4980	74	6F	20	69	6E	66	6F	72	6D	61	63	65	20	70	6F	7A	to informace poz	.....	
0A9C4990	6E	61	6D	65	6E	61	6E	61	20	76	20	4D	46	54	2E	00	namenana v MFT..	.....	
0A9C49A0	FF	FF	FF	FF	82	79	47	11	00	00	00	00	00	00	00	00	yyyy..yg	.....	
0A9C49B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....	.....	
0A9C49C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....	.....	

Sector: 347 684 (0x54E24) Offset: 178 014 208 (0xA9C4800) Read Only

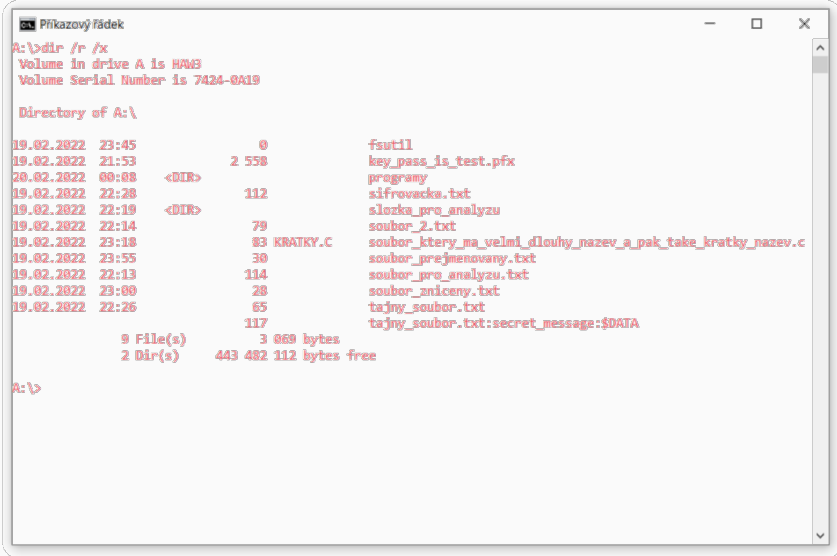
Obrázek 48: Prohlédneme si obsah souboru

## Zadání

Prozkoumejte soubor s alternativním tokem:

1. Podívejte se v průzkumníku na „**tajny\_soubor.txt**“.
2. Spusťte program „**Active@ Disk Editor**“ a najděte soubor, jaké má atributy?
3. Pomocí příkazové řádky zobrazte alternativní toky.
4. Pomocí příkazové řádky zobrazte obsah alternativního toku.
5. Pomocí příkazové řádky vytvořte nový soubor s alternativním tokem.

# Úkol: Prozkoumání ADS (řešení)



```
Příkazový řádek
A:\>dir /r /x
Volume in drive A is HAM3
Volume Serial Number is 7424-8A19

Directory of A:\


19.02.2022  23:45                0      fsutil
19.02.2022  21:53            2 558      key_pass_is_test.pfx
20.02.2022  00:08      <DIR>          programy
19.02.2022  22:28            112      sifrovacka.txt
19.02.2022  22:19      <DIR>          slozka_pro_analyzu
19.02.2022  22:14                79      soubor_2.txt
19.02.2022  23:18            83 KRATKY.C  soubor_ktery_ma_velmi_dlouhy_nazev_a_pak_take_kratky_nazev.c
19.02.2022  23:55                30      soubor_přejmenovaný.txt
19.02.2022  22:13            114      soubor_pro_analyzu.txt
19.02.2022  23:00                28      soubor_zniceny.txt
19.02.2022  22:26                65      tajny_soubor.txt
                                     tajny_soubor.txt:secret_message:$DATA

                9 File(s)            3 069 bytes
                2 Dir(s)          443 482 112 bytes free

A:\>
```

Obrázek 49: Vylistování dalších atributů v programu cmd

## Úkol: Prozkoumání ADS (řešení)



```
Příkazový řádek

A:\>more < tajny_soubor.txt:secret_message
"Toto je skryty obsah souboru, je to alternativni stream. Jaky muze byt dopad nebo aplikace teto vlastnosti N
TFS?"

A:\>
```

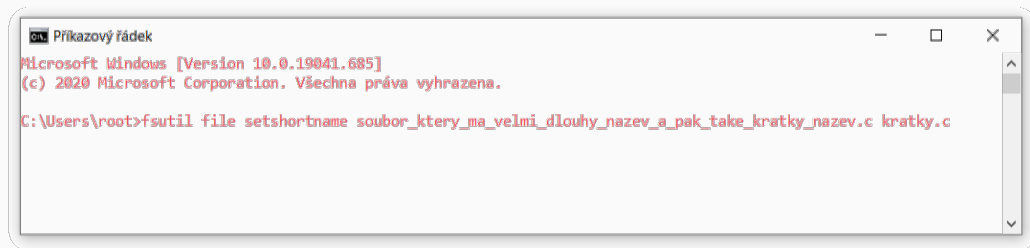
**Obrázek 50:** Výpis obsahu ADS v programu cmd

## Úkol: Prozkoumání ADS (řešení)



**Obrázek 51:** Vytvoření ADS v programu cmd

## Úkol: Prozkoumání ADS (řešení, bonus)



```
Príkazový řádek
Microsoft Windows [Version 10.0.19041.685]
(c) 2020 Microsoft Corporation. Všechna práva vyhrazena.

C:\Users\root>fsutil file setshortname soubor_ktery_ma_velmi_dlouhy_nazev_a_pak_take_kratky_nazev.c kratky.c
```

**Obrázek 52:** Vytvoření alternativního krátkého názvu v programu cmd



## Úkol: Prozkoumání ADS (řešení, bonus)



```
Příkazový řádek

A:\>certutil -hashfile key_pass_is_test.pfx
SHA1 hash of key_pass_is_test.pfx:
134e9bade1678ede69c31c9decaf105080d112f7
CertUtil: -hashfile command completed successfully.

A:\>certutil -hashfile key_pass_is_test.pfx MD5
MD5 hash of key_pass_is_test.pfx:
523f9ef3a7d78dfd128832ab4aff4087
CertUtil: -hashfile command completed successfully.

A:\>
```

**Obrázek 53:** Vygenerování hash souboru pro ověření obsahu

### Zadání

1. Vytvořte obrázek, který bude mít jako alternativní tok program nebo virtuální disk.
2. Spusťte program z alternativního toku (příkaz start).
3. Vytvořte komprimovaný soubor.
4. Vytvořte šifrovaný soubor.
5. Může být soubor komprimovaný i šifrovaný zároveň?
6. prozkoumejte MFT záznamy dalších souborů na disku „HAW3“.

## Kontrolní otázky






---

- Co je to transakce?
- Jaké jsou výhody souborového systému NTFS vzhledem k systému FAT?
- Co to jsou metasoubory?
- Co je to žurnálování a k čemu se prakticky používá?
- Co je to MFT a co obsahuje?
- Nakreslete schéma rozložení disku s NTFS.
- Co je to Cluster?
- Co obsahuje soubor \$Bitmap?
- Co obsahuje soubor \$Badcllus?
- Co všechno obsahuje záznam o jednom souboru, kde se nachází, jakou má (nebo může mít) velikost tento záznam?
- Kde se nachází a co znamená vlajka „dirty“?

**Děkuji za pozornost!**






## Zdroje

---

-  ÅRNES, A., 2017. *Digital Forensics*. Wiley. ISBN 9781119262404. Dostupné také z: <https://books.google.cz/books?id=Fk0nDwAAQBAJ>.
-  CARRIER, Brian, 2005. *File system forensic analysis*. 1st edition. Upper Saddle River: Addison-Wesley. ISBN 978-0321268174. Dostupné také z: <https://repo.zenk-security.com/Forensic/File%20System%20Forensic%20Analysis.pdf>.
-  HENRY, Timothy, 2011. *NTFS Concepts and Analysis*. Kingston (Rhode Island): Department of Computer Science a Statistics, University of Rhode Island. Dostupné také z: [https://homepage.cs.uri.edu/~thenry/CSC487\\_Video\\_Library.html](https://homepage.cs.uri.edu/~thenry/CSC487_Video_Library.html).
-  HORÁK, Jaroslav, 2007. *Hardware: učebnice pro pokročilé*. 4., aktualiz. vyd. Brno: Computer Press. ISBN 978-80-251-1741-5.
-  KCALL.CO.UK, 2020. *Everything I know about NTFS* [online]. [kcall.co.uk](http://kcall.co.uk) [cit. 2022-02-18]. Dostupné z: <http://kcall.co.uk/ntfs/index.html>.

-  KHOLODOV, Igor, 2007. *NTFS File System Overview: Computer Organization and Design* [online]. Fall River: Computer Information Systems Department / STEM, Bristol Community College [cit. 2022-02-20]. Dostupné z:  
<http://c-jump.com/bcc/t256t/Week04NtfsReview/index.html>.
-  MEDEIROS, Jason, 2008. *NTFS Forensics: A Programmers View of Raw Filesystem Data Extraction*. Grayscale Research. Dostupné také z:  
<http://grayscale-research.org/new/pdfs/NTFS%20forensics.pdf>.
-  METZ, Joachim, 2009. *New Technologies File System (NTFS): 0.0.25* [online]. USA: @libyal [cit. 2022-02-20]. Dostupné z:  
[https://github.com/libyal/libfsntfs/blob/main/documentation/New%5C%20Technologies%5C%20File%5C%20System%5C%20\(NTFS\).asciidoc](https://github.com/libyal/libfsntfs/blob/main/documentation/New%5C%20Technologies%5C%20File%5C%20System%5C%20(NTFS).asciidoc).
-  MICROSOFT, 1993. *Windows NT Server Logo*. Dostupné také z:  
<https://logodix.com/logos/340506>.



-  NTFS Basics, 2021. Mississauga: LSoft Technologies. Dostupné také z:  
<http://www.ntfs.com/ntfs-multiple.htm>.
-  RALBOVSKÝ, Petr, 2013. *NTFS struktura*. Havířov: Střední průmyslová škola elektrotechnická.
-  RUSSINOVICH, Mark E.; SOLOMON, David A., 2005. *Microsoft Windows internals: Microsoft Windows Server 2003, Windows XP, and Windows 2000*. 4th ed. Redmond: Microsoft Press. ISBN 07-356-1917-4. Dostupné také z:  
<https://flylib.com/books/en/4.491.1.109/1/>.
-  RUSSON, Richard; FLEDEL, Yuval, 2005. *NTFS Documentation* [online]. Dubeyko [cit. 2022-02-18]. Dostupné z:  
<http://dubeyko.com/development/FileSystems/NTFS/ntfsdoc.pdf>.
-  SEDORY, Daniel B., 2018. *An Introduction to NTFS: New Technology File System* [online]. The Starman's Realm [cit. 2022-02-18]. Dostupné z:  
<https://thestarman.pcministry.com/asm/mbr/IntNTFSfs.htm>.

Tato prezentace slouží jako podklad pro výuku předmětu Operační systémy na *Střední průmyslové škole elektrotechnické v Havířově*. Pro případné připomínky, vylepšení nebo poznatky prosím využijte *issues* v Git-repositáři projektu nebo mého školního e-mailu:

<https://github.com/michto01/spse-materials>

[tomas.michalek@spsehavirov.cz](mailto:tomas.michalek@spsehavirov.cz)

Podporujeme svobodné licencování! Tato prezentace je volně šiřitelná v souladu s licencí Creative Commons 4.0 (CC BY-SA 4.0) :

