



EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání



Operační systémy

Technologie ukládání dat a souborové systémy

Strategický projekt UTB ve Zlíně, reg. č. CZ.02.2.69/0.0/0.0/16_015/0002204



Martin Sysel
Fakulta aplikované informatiky
Univerzita Tomáše Bati ve Zlíně

Úvod

□ Technologie ukládání dat

■ Fyzické

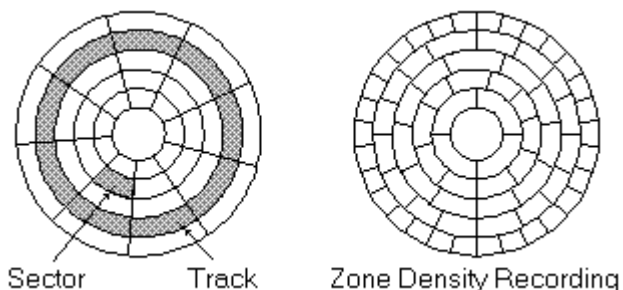
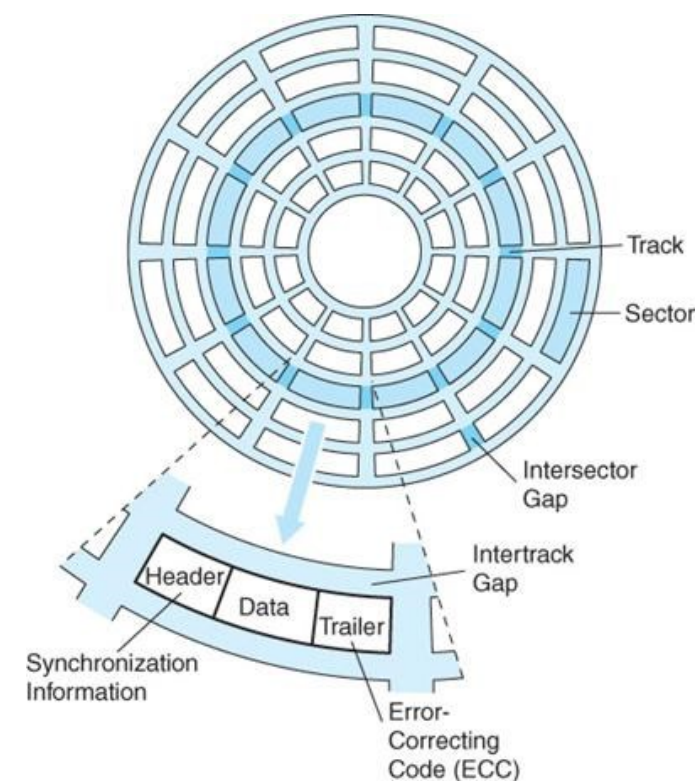
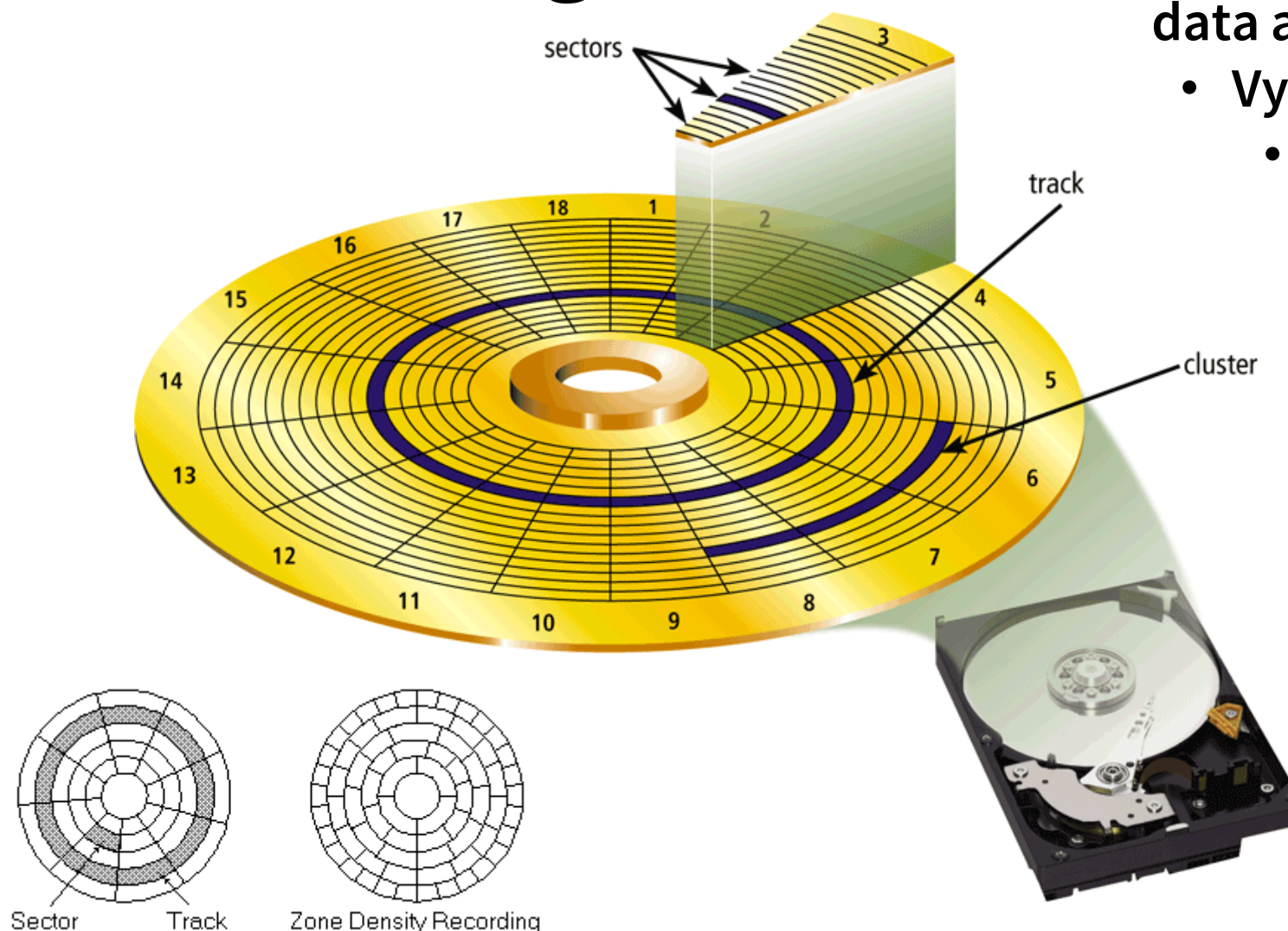
- Pevný disk, Solid State Disk (SSD), ...
- Sekundární úložiště je obecně úzkým místem
 - Vylepšení výkonu sekundárního úložiště významně zvyšuje celkový výkon systému

■ Logické

- Souborový systém
 - vytváří strukturu svazků a složek, data jsou ukládána do jednotlivých souborů.
 - Souborový systém je integrovanou součástí operačního systému
 - » alespoň jeden souborový systém
 - » Operační systémy podporují několik různých systémů souborů.

Terminologie

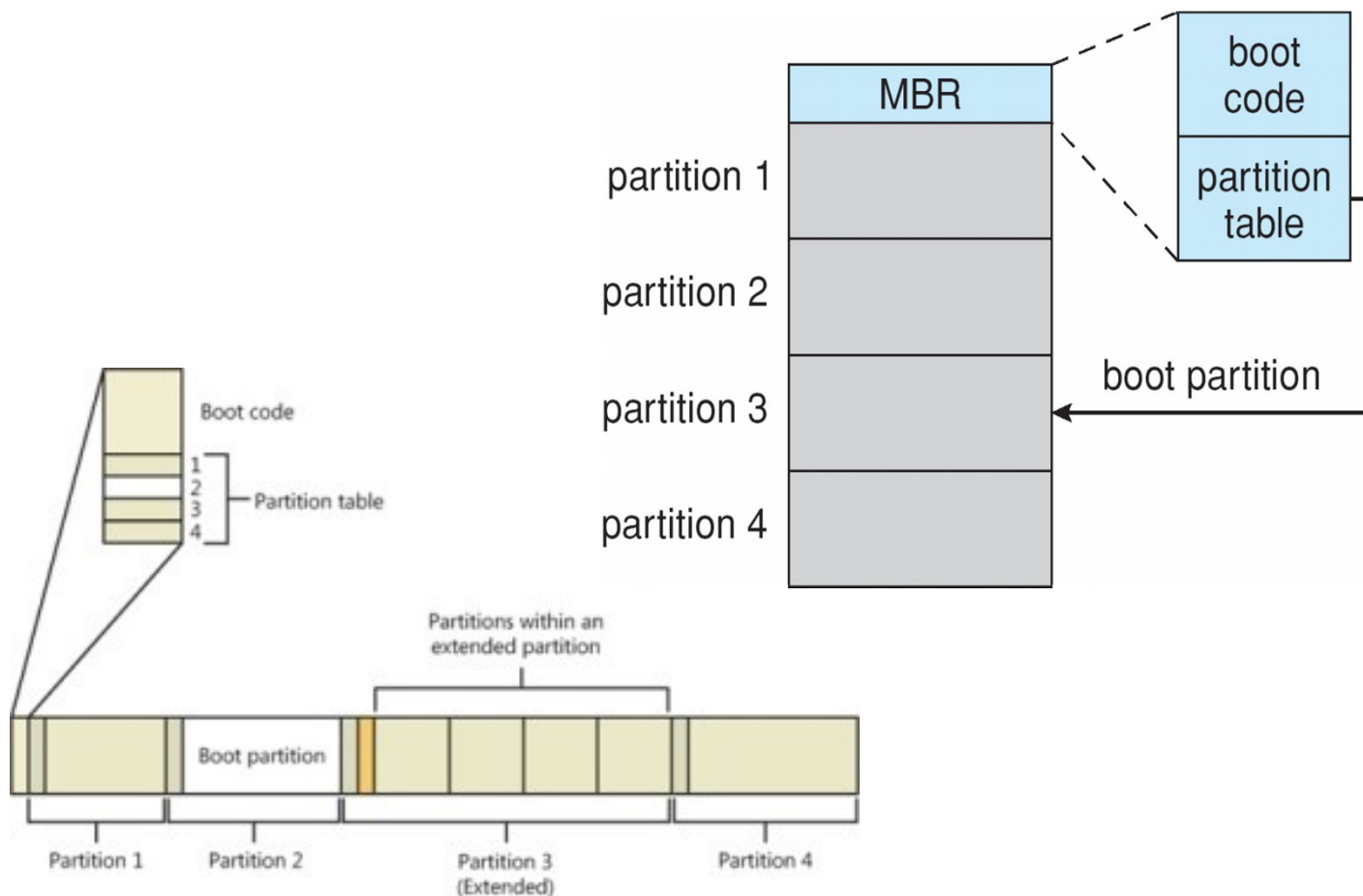
- Každý sektor obsahuje hlavičku, data a kontrolní součet (ECC)
 - Využitelná velikost sektoru pro data
 - 512 B nebo 4 kB



Správa disku

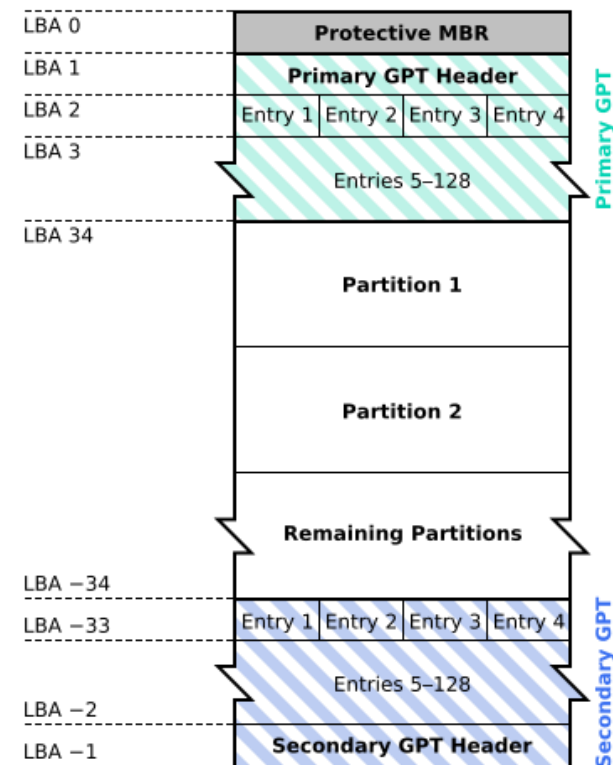
- ❑ Nízkoúrovňové (fyzické) formátování
 - Rozdělení disku na stopy a sektory. Adresování pomocí LBA.
 - Každý sektor obsahuje informace záhlaví, data a ECC
 - Obvykle 512 B nebo 4 kB dat
- ❑ Rozdělení disku na jednu nebo více skupin cylindrů.
 - Rozdělení disku na několik logických částí (MBR, GPT).
 - Primární, rozšířený a logický oddíl.
- ❑ Logické formátování (vysokoúrovňové)
 - Vytvoření systému souborů
 - Většina souborových systémů seskupuje sektory do klastrů (datové bloky)
 - Hlavním důvodem bylo omezení v adresování klastrů.

Oddíly



(Zdroj: https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/10_MassStorage.html)

GUID Partition Table Scheme



GUID Partition Table Scheme
(Zdroj: https://en.wikipedia.org/wiki/GUID_Partition_Table)

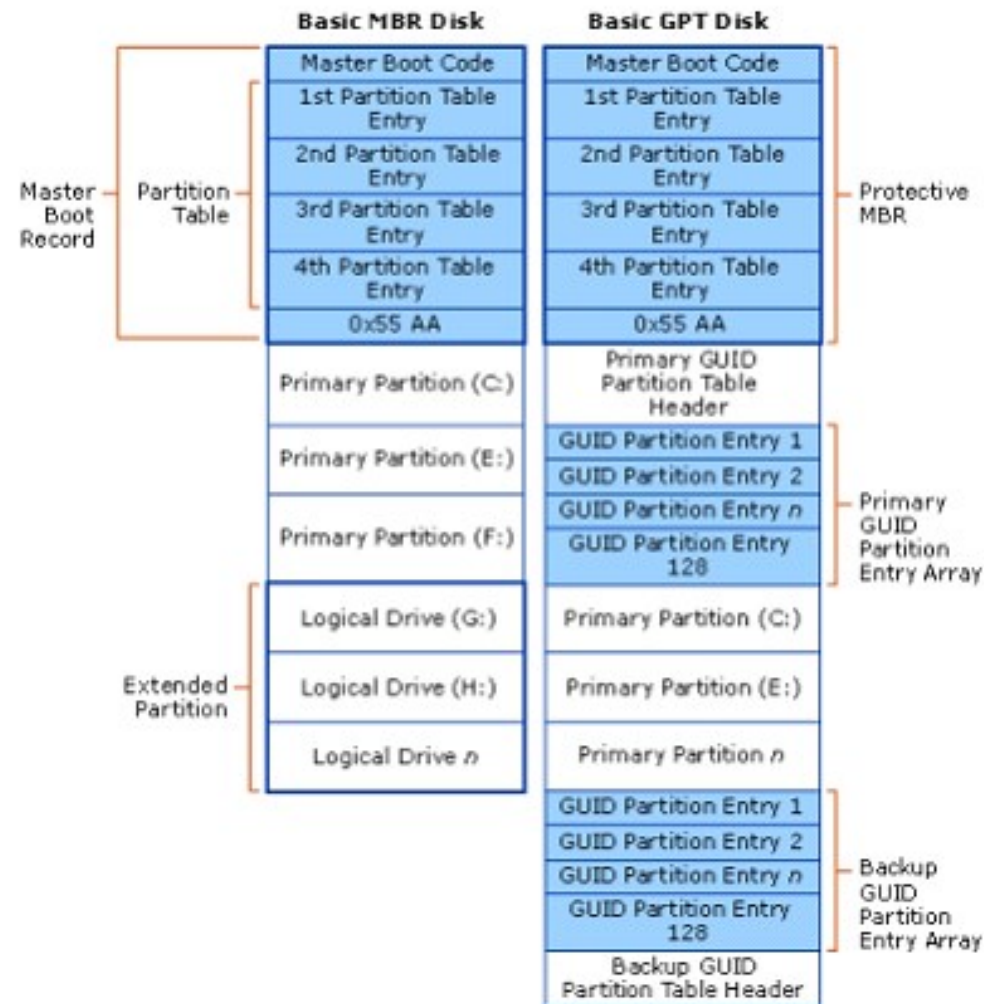
Master Boot Record (MBR)

- ❑ Nachází se na začátku rozdělení úložiště (první sektor)
 - LBA 0 (dříve Cylindr 0, hlava 0, sektor 1)
- ❑ Obsahuje informace o umístění oddílů na médiu.
- ❑ Obsahuje také spustitelný kód, který spustí zavaděč OS.
 - Obvykle najde druhou fázi zavaděče.

GUID Partition Table

- ❑ Standard pro rozložení oddílů na HD
 - Podpora více než 2 TB disků
 - Podpora 128 oddílů
 - Používá globální jedinečné identifikátory (GUID)
 - Na konci disku se ukládá záložní kopie tabulky

Comparison of MBR and GPT Disks



GUID Partition Table Scheme

(Zdroj: https://en.wikipedia.org/wiki/GUID_Partition_Table)⁷

Souborové systémy (File System)

- ❑ Organizují soubory a spravují přístup k datům.
- ❑ Poskytují jmenný prostor pro soubory prostřednictvím adresářů.
- ❑ Uživatelé odkazují na soubory podle symbolických názvů.
- ❑ Správa souborů a adresářů, správa pomocných úložišť, mechanismy integrity souborů a metody přístupu.
 - Především se jedná o správu sekundárního úložného prostoru, zejména diskového úložiště.

(Deitel, Deitel & Choffness, 2004; Silberschatz, Galvin & Gagne, 2013; Tanenbaum, 2015)

Soubor

□ Soubory

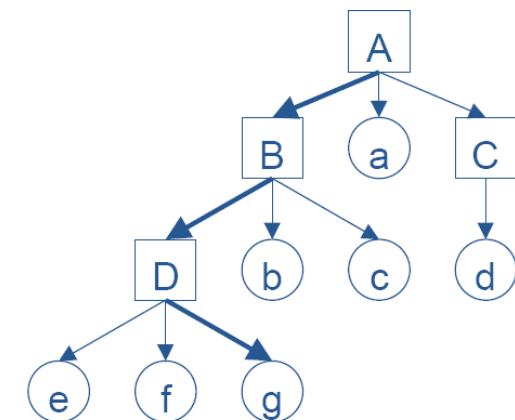
- Pojmenovaná kolekce dat, které jsou zpracovávány jako jednotka.
- Uloženo na sekundárních úložných zařízeních.

□ Vlastnosti souboru

- Má metadata (atributy) pro popis souboru.
 - Metadata znamenají údaje o uživatelských datech.
- Různé operační systémy udržují různé atributy souborů.
 - Jméno, identifikátor, typ, velikost, oprávnění, čas a datum, vlastník,...
- Operace se soubory: vytváření, mazání, otevírání, zavírání, čtení, zápis, vyhledávání

Adresáře

- ❑ Katalogizace struktury systému souborů, která obsahuje odkazy na jiné počítačové soubory a případně další adresáře (hierarchický souborový systém).
 - Adresáře jsou známy jako složky, katalogy
 - Názvy souborů musí být jedinečné pouze v adresáři
- ❑ Adresář je zvláštní případ souboru.
 - Soubory obsahující názvy a umístění dalších souborů v systému souborů k uspořádání a rychlému vyhledání souborů
- ❑ Hierarchický souborový systém
 - Podadresář je adresář obsažený v jiném adresáři
 - Často se používají rodiče a potomci
 - Kořenový adresář - nemá nadřazeného rodiče

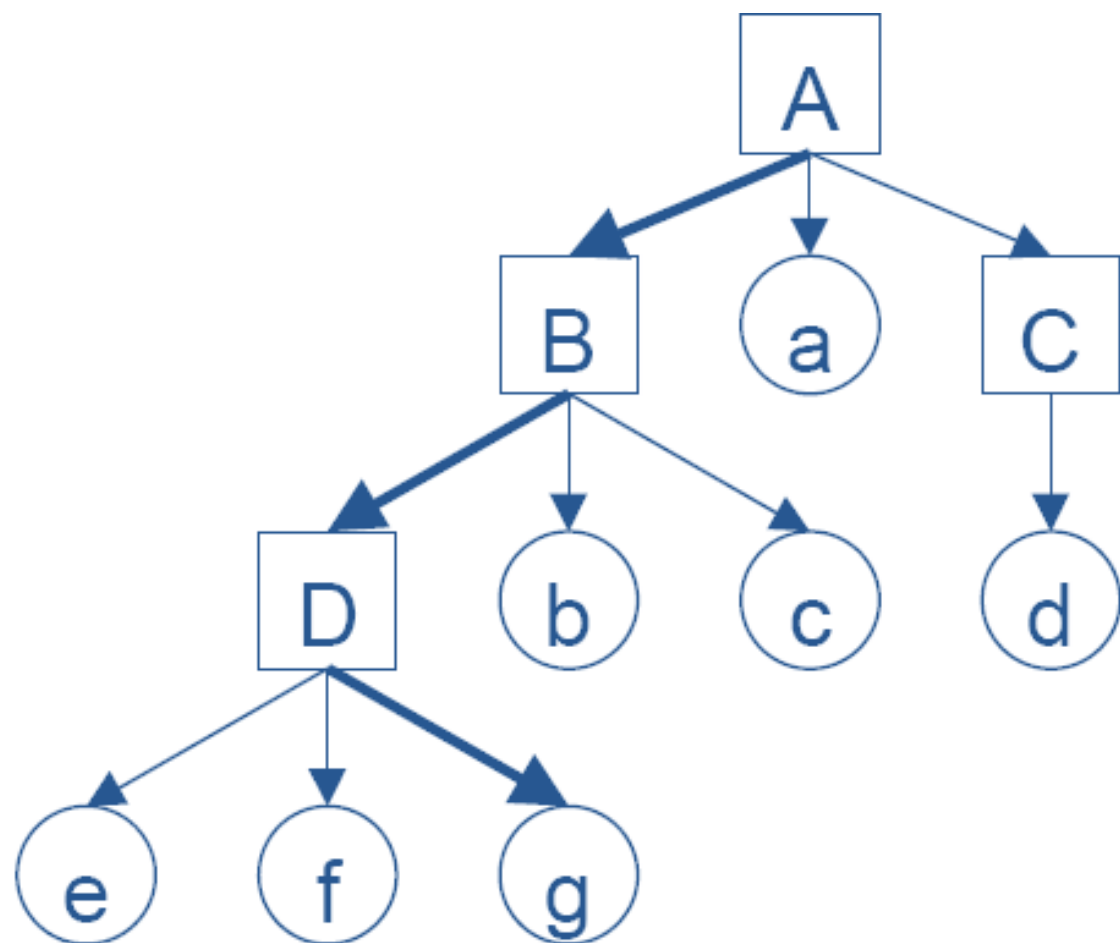


Adresářový strom

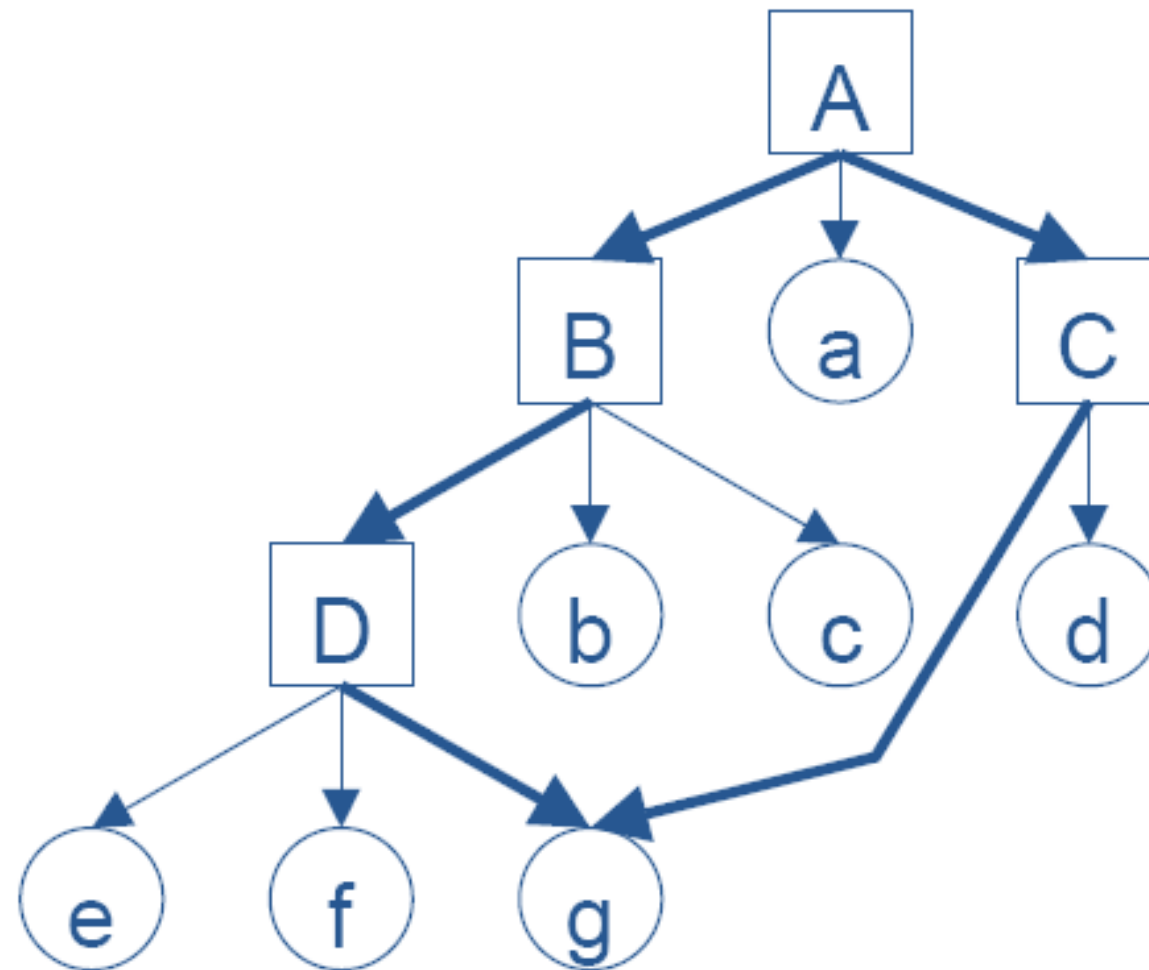
- ❑ Vytváří se adresářový strom obsahující soubory
 - Každý adresář může obsahovat soubory nebo položky adresáře
 - Zjednodušuje navigaci pomocí cest
- ❑ Název souboru je obvykle tvořen jako cesta z kořenového adresáře do souboru - **Absolutní cesta**
 - Začíná lomítkem / (zpětné lomítko \)
- ❑ Každý proces má **aktuální (pracovní) adresář**
 - Můžete změnit aktuální adresář podle potřeby
 - Můžete pojmenovávat soubory relativně k tomuto adresáři
- ❑ **Relativní cesta** nezačíná v kořenovém adresáři, ale v aktuálním adresáři

(Deitel, Deitel & Choffness, 2004; Silberschatz, Galvin & Gagne, 2013; Tanenbaum, 2015)

Adresářový strom



Acyklický adresářový strom



Acyklická adresářová struktura

- ❑ Link: položka adresáře, která odkazuje na datový soubor nebo adresář umístěný v jiném adresáři.
- ❑ Stejný soubor může mít několik jmen.
 - absolutní název cesty se liší, ale obsah souboru je stejný
- ❑ Smazání - jeden uživatel odstraní sdílený soubor.
 - Dojde k odstranění pro všechny uživatele. Linky ukazují do prázdna.
 - NEBO soubor zůstane, dokud není smazán poslední odkaz (link)
 - Udržuje počet odkazů na danou položku
- ❑ Soft link: položka adresáře obsahující cestu k jinému souboru.
- ❑ Hard link: položka adresáře, která určuje umístění souboru
 - např. číslo uzlu na paměťovém zařízení.

(Deitel, Deitel & Choffness, 2004; Silberschatz, Galvin & Gagne, 2013; Tanenbaum, 2015)

Organizace souborů - metody přidělování

- ❑ Metoda alokace se týká způsobu přidělování diskových bloků pro soubory
 - Souvislé přidělování
 - Nesouvislé přidělování
 - Vázané (Linked) přidělování
 - Tabulkové přidělování
 - Indexované přidělování
- ❑ Problém s přidělováním a uvolňováním místa na sekundárním úložišti.
 - Souvislé alokace byly nahrazeny dynamičtějším nesouvislým přidělováním místa.
 - Soubory mají v průběhu času tendenci růst nebo se zmenšovat.
 - Uživatelé zřídka předem vědí, jak velké budou jejich soubory.

Problém s **Fragmentací**

(Deitel, Deitel & Choffness, 2004; Silberschatz, Galvin & Gagne, 2013; Tanenbaum, 2015)

Fragmentace souborů

- ❑ Fragmentace systému souborů zvyšuje čas přístupu k datům
 - platí pouze pro HDD, nikoli pro SSD.
- ❑ Fragmentace souboru
 - Externí fragmentace souboru
 - Jeden soubor je rozdělen do několika částí.
 - Vnitřní fragmentace souborů
 - Poslední klastr není plně využit. Větší klastry znamenají větší ztrátu místa.
- ❑ Fragmentace volného místa
 - Existuje několik malých nevyužitých oblastí FS.
 - Je to obvykle způsobeno vymazáním nebo zkrácením souborů.

(Deitel, Deitel & Choffness, 2004; Silberschatz, Galvin & Gagne, 2013; Tanenbaum, 2015)

Souvislé přidělování místa na disku

□ Každý soubor je uložen v sousedících blocích na disku

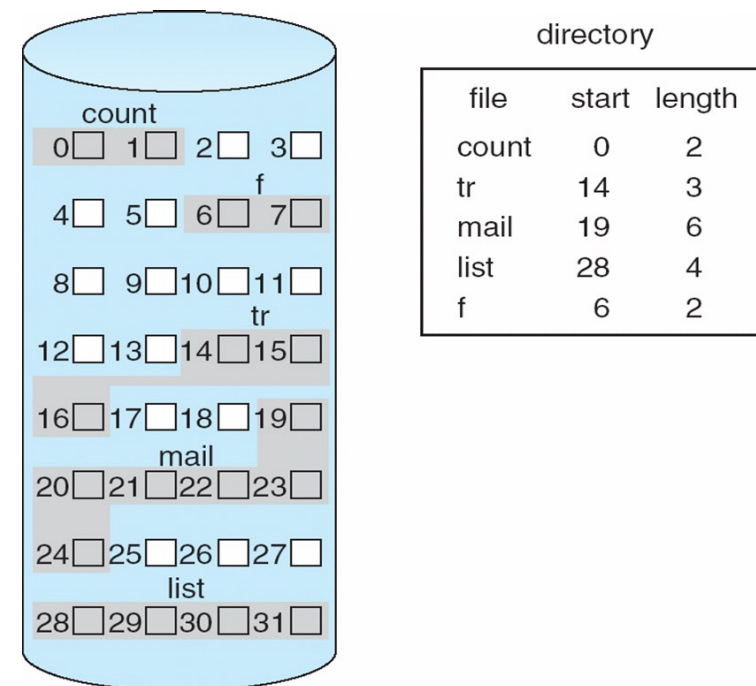
▪ Jednoduché - vyžaduje pouze:

- Počáteční pozici (číslo bloku)
- Délka (počet bloků)

□ Nevýhody

- Plýtvání místem
- Fragmentace
- Zvětšování souborů je problematické

(Silberschatz, Galvin & Gagne, 2013)

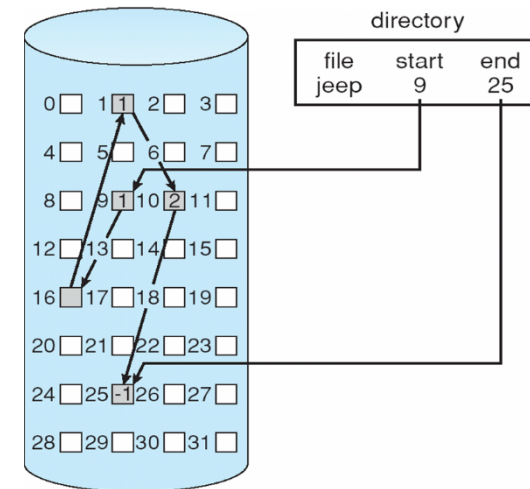


(Silberschatz, Galvin & Gagne, 2013)

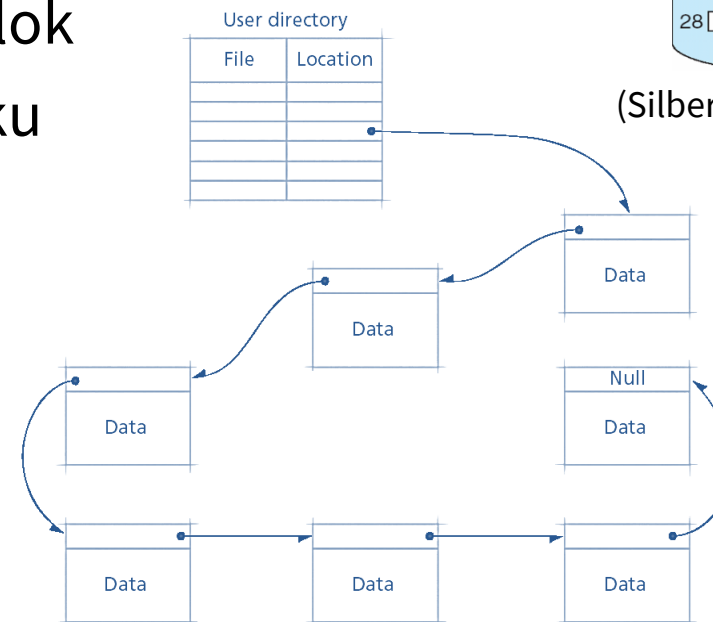
Vázané (Linked) přidělování místa na disku

- ❑ Každý soubor je provázaným seznamem diskových bloků
- ❑ Bloky mohou být umístěny kdekoli
 - Odkazujeme na první (případně i poslední) blok souboru
 - Každý blok obsahuje ukazatel na další blok
 - Řetězec musí být prohledáván od začátku

(Deitel, Deitel & Choffness, 2004; Silberschatz, Galvin & Gagne, 2013)



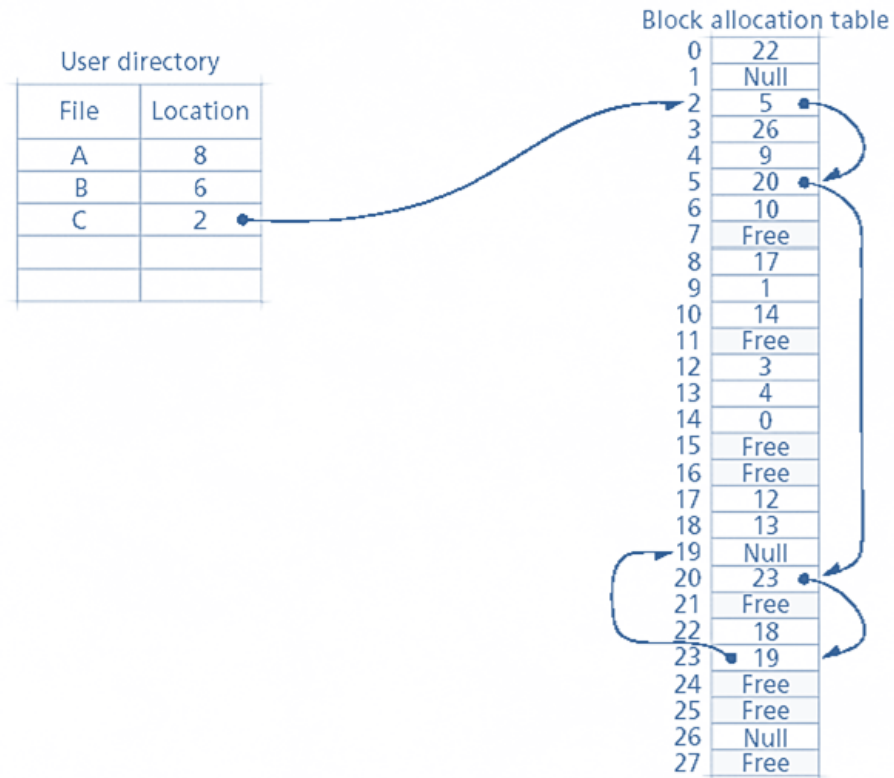
(Silberschatz, Galvin & Gagne, 2013)



(Deitel, Deitel & Choffness, 2004)

Tabulkové přidělování místa

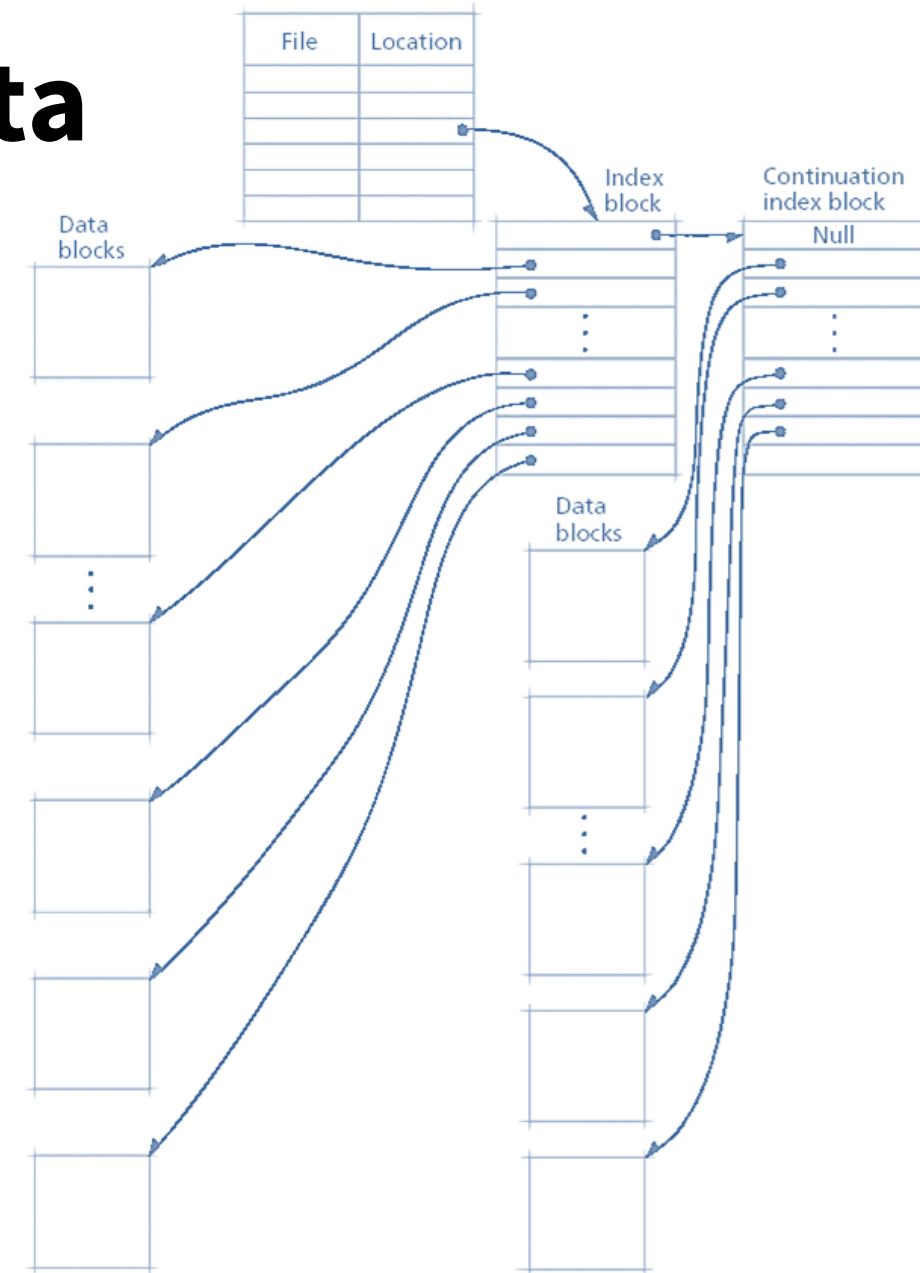
- ❑ Udržujte provázaný seznam odkazů v tabulce
 - Snižuje počet vyhledávání požadovaných pro přístup k fyzickým blokům. Využití vyrovnávací paměti.
- ❑ Položky adresáře označují první blok souboru.
- ❑ Aktuální číslo bloku se používá jako index do tabulky přidělování bloků k určení umístění dalšího bloku.
- ❑ např. Souborový systém FAT



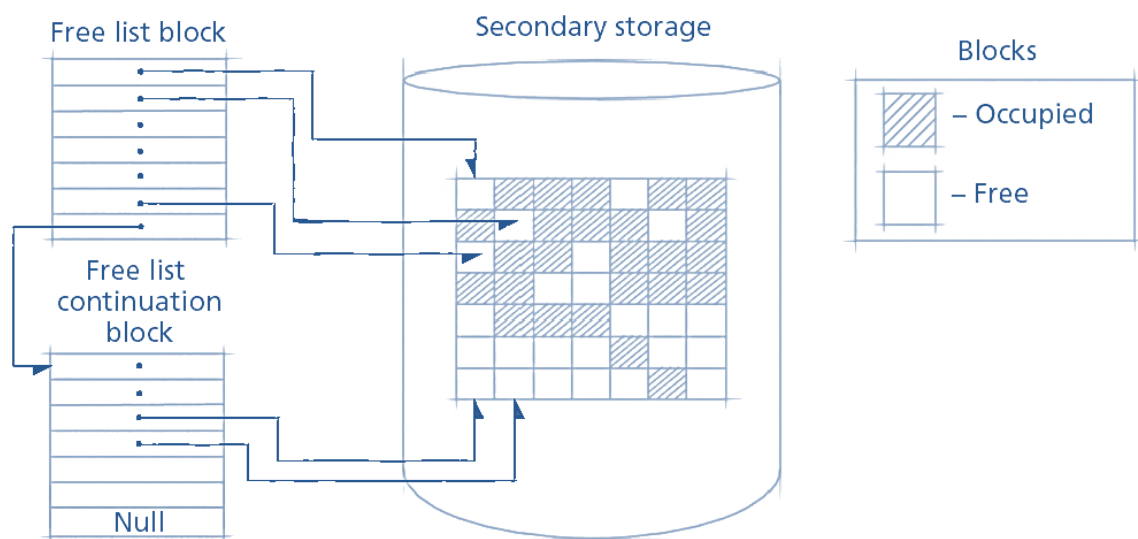
Block 0 B(4)	Block 1 B(10)	Block 2 C(1)	Block 3 A(4)	Block 4 B(8)	Block 5 C(2)	Block 6 B(1)
Block 7 Free	Block 8 A(1)	Block 9 B(9)	Block 10 B(2)	Block 11 Free	Block 12 A(3)	Block 13 B(7)
Block 14 B(3)	Block 15 Free	Block 16 Free	Block 17 A(2)	Block 18 B(6)	Block 19 C(5)	Block 20 C(3)
Block 21 Free	Block 22 B(5)	Block 23 C(4)	Block 24 Free	Block 25 Free	Block 26 A(5)	Block 27 Free

Indexované přidělování místa

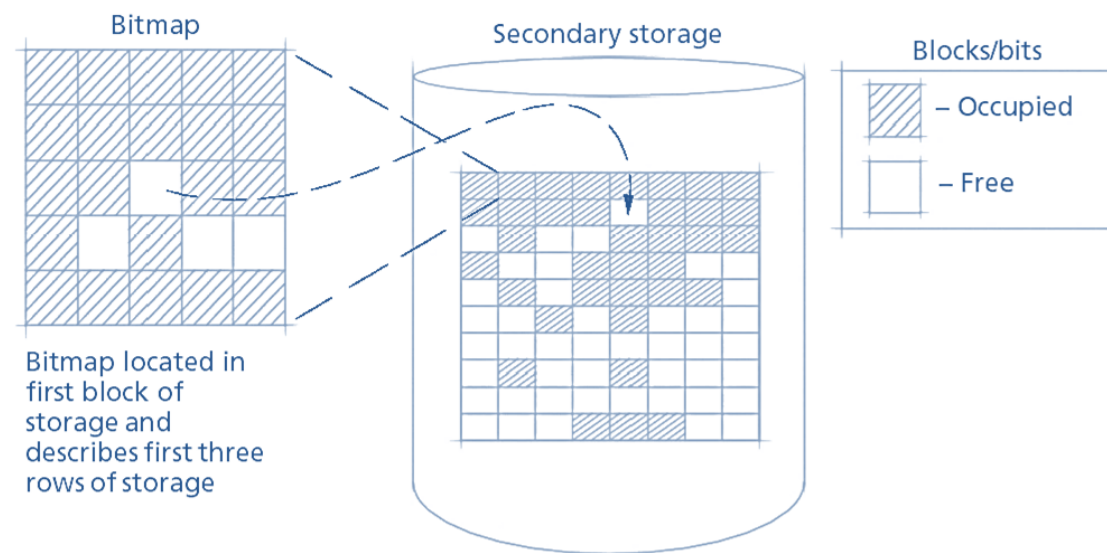
- ❑ Adresářový záznam souboru odkazuje na jeho indexový blok (tabulka indexů).
 - Indexový blok si může vyhradit posledních několik položek pro uložení ukazatelů do dalších indexových bloků
- ❑ Každý soubor má svůj indexový blok
 - případně několik indexových bloků
- ❑ Indexové bloky obsahují seznam ukazatelů, které ukazují přímo na datové bloky souborů
- ❑ např. Souborové systémy NTFS nebo EXT



Management volného místa



Management volného místa používající seznam volných bloků.



Management volného místa používající bitmapu.

(Deitel, Deitel & Choffness, 2004)

Integrita dat

- ❑ Pokud během operace zápisu dojde k selhání systému, mohou být data souboru v nekonzistentním stavu.
 - Pokud nelze tolerovat ztrátu dat => protokolování transakcí
- ❑ Souborové systémy založené na transakcích
 - Snižuje možnost ztráty dat pomocí atomických transakcí:
 - Proveďte skupinu operací jako celek nebo vůbec
 - Pokud dojde k chybě, která brání dokončení transakce, je stav vrácen zpět do stavu před zahájením transakce.
 - Provádění všech operací FS jako protokolované transakce se nazývá **žurnálování**.
 - Některé souborové systémy se snaží snížit režie pomocí protokolování pouze metadat

(Deitel, Deitel & Choffness, 2004; Silberschatz, Galvin & Gagne, 2013)

Souborové systémy

- ❑ Existuje mnoho různých druhů souborových systémů.
 - Každý z nich má jinou strukturu a logiku.
- ❑ FAT (FAT12, FAT16, FAT32), ExFAT
- ❑ NTFS
- ❑ Ext (ext2, ext3, ext4)
- ❑ BTRFS
- ❑ ZFS

FAT – File Allocation Table

- ❑ Jednoduchý souborový systém původně navržený pro malé disky a jednoduché struktury složek.
- ❑ Pojmenován podle způsobu organizace
 - Alokační tabulka souborů (FAT)
 - Udržuje provázaný seznam v samostatné tabulce.
 - Nachází se na začátku svazku.
 - Uchovávají se dvě kopie tabulky pro případ poškození.
- ❑ Alokační tabulka a kořenová složka je uložena fixně
- ❑ Svazek naformátovaný pomocí FS FAT je alokován po klastrech (Cluster).
 - Výchozí velikost klastru závisí na velikosti svazku a souborovém systému.
 - Počet sektorů v klastru.

FAT – File Allocation Table

- ❑ Původně navržen v roce 1977 pro diskety.
- ❑ Tři hlavní varianty systému souborů
 - FAT12
 - Menší kapacita než 16 MiB. (256 MiB pro klastry 64 kiB)
 - FAT16
 - 16 MiB - 2 GiB (Windows NT a novější podpora 4 GiB pro klastry 64 kiB)
 - FAT32
 - 512 MB - 2 TB (16 TiB pro 4 KiB sektory, 64KiB cluster)
 - ExFAT - nástupce rodiny FAT (podobný design a mnoho vylepšení)
 - Teoretická velikost svazku 128 PiB, doporučeno 512 TiB
 - Velikost souboru 128 PiB (teoreticky 16 EiB – 1 velikost souboru uložena jako 64bitové číslo)

Klastr (Cluster)

- ❑ Klastr představuje nejmenší možné místo na disku, které lze použít k uložení souboru. Používá se také pojem alokační jednotka.
- ❑ Větší soubor používá více klastrů. Poslední klastr souboru není plně využit (interní fragmentace).
 - Průměrné množství prostoru ztraceného tímto způsobem na disku lze vypočítat pomocí rovnice
 - $(\text{velikost clusteru}) / 2 * (\text{počet souborů})$
- ❑ Organizace (adresace) na pevném disku je založena na velikosti klastru.
- ❑ Velikost klastru je určena při formátování oddílu.
 - Výchozí hodnoty jsou vybírány na základě velikosti oddílu.

Defaultní velikost klastru (Windows)

FAT 16

Volume size	Windows 7, Windows Vista, Windows Server
7 MB–8 MB	Not supported
8 MB–32 MB	512 bytes
32 MB–64 MB	1 KB
64 MB–128 MB	2 KB
128 MB–256 MB	4 KB
256 MB–512 MB	8 KB
512 MB–1 GB	16 KB
1 GB–2 GB	32 KB
2 GB–4 GB	64 KB

FAT 32

Volume size	Windows 7, Windows Vista, Windows Server
7 MB–16 MB	Not supported
16 MB–32 MB	Not supported
32 MB–64 MB	512 bytes
64 MB–128 MB	1 KB
128 MB–256 MB	2 KB
256 MB–8 GB	4 KB
8 GB–16 GB	8 KB
16 GB–32 GB	16 KB
32 GB–2 TB	Not supported

ExFAT

Volume size	Windows 7, Windows XP
7 MB–256 MB	4 KB
256 MB–32 GB	32 KB
32 GB–256 TB	128 KB
> 256 TB	Not supported

NTFS

Volume size	Windows 7, Windows Vista, Windows Server
7 MB–512 MB	4 KB
512 MB–1 GB	4 KB
1 GB–2 GB	4 KB
2 GB–2 TB	4 KB
2 TB–16 TB	4 KB
16 TB–32 TB	8 KB
32 TB–64 TB	16 KB
64 TB–128 TB	32 KB
128 TB–256 TB	64 KB

Struktura souborového systému FAT

❑ Vyhrazená oblast

- Boot sektor oddílu (blok parametrů)

❑ Oblast FAT

- Dvě kopie alokačních tabulek pro soubory (adresáře).

❑ Kořenový adresář

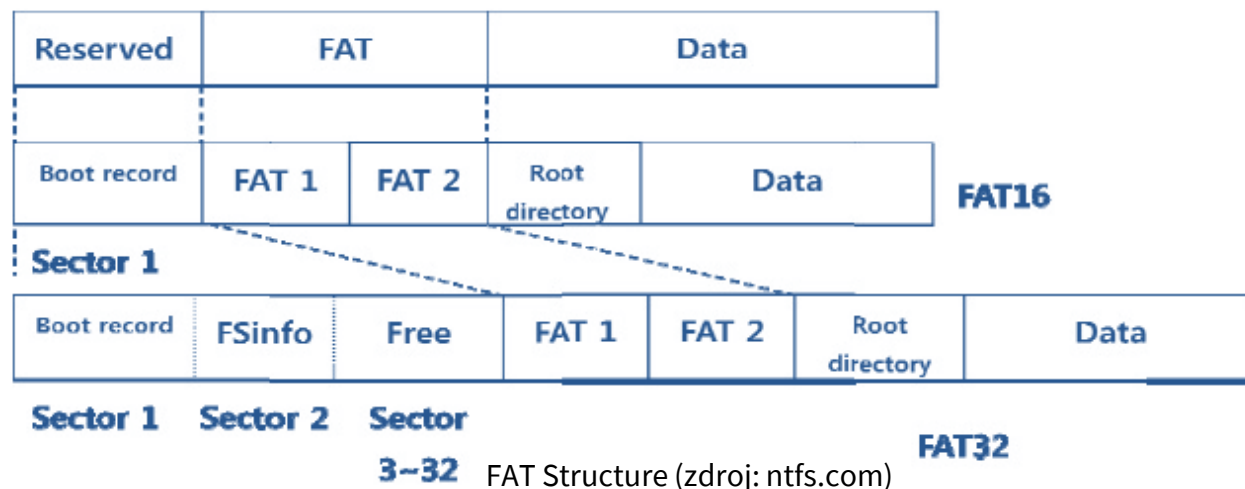
- Popisuje soubory a složky v kořenovém adresáři oddílu.

❑ Klastry

- Soubory a adresáře

Boot Sector	Reserved Sectors	FAT 1	FAT 2 (Duplicate)	Root Folder	Other Folders and All Files
-------------	------------------	-------	-------------------	-------------	-----------------------------

FAT Structure (source: microsoft.com)



Boot sektor oddílu (FAT Partition Boot Sector)

- ❑ Obsahuje informace, které FS používá pro přístup ke svazku.
- ❑ Tabulka popisuje pole v oddílu spouštěcího oddílu pro svazek naformátovaný pomocí systému souborů FAT.

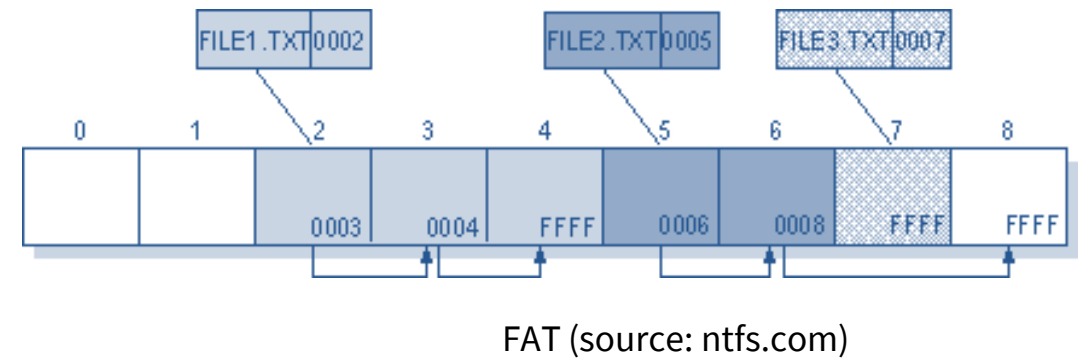
Byte Offset (in hex)	Field Length	Sample Value	Meaning
00	3 bytes	EB 3C 90	Jump instruction
03	8 bytes	MSDOS5.0	OEM Name in text
0B	25 bytes		BIOS Parameter Block
24	26 bytes		Extended BIOS Parameter Block
3E	448 bytes		Bootstrap code
1FE	2 bytes	0x55AA	End of sector marker

(ntfs.com, 2019)

Byte Offset	Délka	Hodnota (ukázka)	Význam
0x0B	WORD	0x0002	Bytes per Sector.
0x0D	BYTE	0x08	Sectors Per Cluster.
0x0E	WORD	0x0100	Reserved Sectors. The number of sectors from the Partition Boot Sector to the start of the first file allocation table, including the Partition Boot Sector.
0x10	BYTE	0x02	Number of file allocation tables (FATs). The number of copies of the file allocation table on the volume.
0x11	WORD	0x0002	Root Entries. The total number of file name entries that can be stored in the root folder of the volume.
0x13	WORD	0x0000	Small Sectors. The number of sectors on the volume if the number fits in 16 bits (65535). For volumes larger than 65536 sectors, this field has a value of 0 and the Large Sectors field is used instead.
0x15	BYTE	0xF8	Media Type. Provides information about the media being used. A value of 0xF8 indicates a hard disk.
0x16	WORD	0xC900	Sectors per file allocation table (FAT). Number of sectors occupied by each of the file allocation tables on the volume. By using this information, together with the Number of FATs and Reserved Sectors, you can compute where the root folder begins. By using the number of entries in the root folder, you can also compute where the user data area of the volume begins.
0x18	WORD	0x3F00	Sectors per Track. The apparent disk geometry in use when the disk was low-level formatted.
0x1A	WORD	0x1000	Number of Heads. The apparent disk geometry in use when the disk was low-level formatted.

Alokační tabulka

- ❑ U FAT16 a FAT12 je první FAT vždy primární a všechny změny jsou automaticky zapsány do kopie.
- ❑ Alokační tabulka obsahuje informace o každém klastru.
 - popisuje pouze datové klastry
- ❑ Typy hodnot:
 - Nepoužitý (0x0000)
 - Klastr používaný souborem
 - Chybný cluster (0xFFFF7)
 - Poslední klastr souboru (0xFFFF8-0xFFFFF)
- ❑ Každý záznam obsahuje ukazatel na další klastr souboru nebo indikaci (0xFFFF), že tento klastr je poslední.
 - Počáteční adresa klastru je uložena v adresářovém záznamu.



(ntfs.com, 2019)

Kořenový adresář

- ❑ Kořenový adresář obsahuje záznamy pro soubory a složky v tomto adresáři.
- ❑ Jediný rozdíl mezi kořenovou složkou a ostatními složkami je v tom, že kořenová složka je na určeném místě na disku
 - U FAT12 a FAT16 má pevnou velikost (omezený počet záznamů) a pevné umístění.
 - Kořenový adresář FAT32 nemá pevnou velikost, může růst. Toto je obyčejný klastrový řetězec.
- ❑ Adresáře obsahují 32B adresářové záznamy pro každý soubor a podsložku obsaženou v adresáři.
 - Všechny záznamy v adresáři mají stejnou velikost

Struktura adresářového záznamu

- ❑ Jméno (8 + 3 znaky)
- ❑ Atributy (8 bitů)
- ❑ Čas vytvoření (24 bitů)
- ❑ Datum vytvoření (16 bitů)
- ❑ Datum posledního přístupu (16 bitů)
- ❑ Čas poslední úpravy (16 bitů)
- ❑ Datum poslední úpravy (16 bitů)
- ❑ Počáteční číslo clusteru (16 bitů)
- ❑ Velikost souboru (32 bitů)

T	H	E	Q	U	I	~	1	F	O	X	0X20	NT	Create time
Create date		Last access date		0x0000		Last modified time		Last modified date		First cluster		File size	

Root Directory SFN Entry Data Structure	
Bytes	Purpose
0	First character of file name (ASCII) or allocation status (0x00=unallocated, 0xe5=deleted)
1-10	Characters 2-11 of the file name (ASCII); the "." is implied between bytes 7 and 8
11	File attributes (see File Attributes table)
12	Reserved
13	File creation time (in tenths of seconds)*
14-15	Creation time (hours, minutes, seconds)*
16-17	Creation date*
18-19	Access date*
20-21	High-order 2 bytes of address of first cluster (0 for FAT12/16)*
22-23	Modified time (hours, minutes, seconds)
24-25	Modified date
26-27	Low-order 2 bytes of address of first cluster
28-31	File size (0 for directories)

File Attributes	
Flag Value	Description
0000 0001 (0x01)	Read-only
0000 0010 (0x02)	Hidden file
0000 0100 (0x04)	System file
0000 1000 (0x08)	Volume label
0000 1111 (0x0f)	Long file name
0001 0000 (0x10)	Directory
0010 0000 (0x20)	Archive

* Bytes 13-22 are unused by DOS

Souborový systém FAT

Adresářový záznam

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00h	N	E	C	O					D	A	T	20h				
10h							time		date		02h 00h	27h 10h	00h 00h			

Alokační tabulka

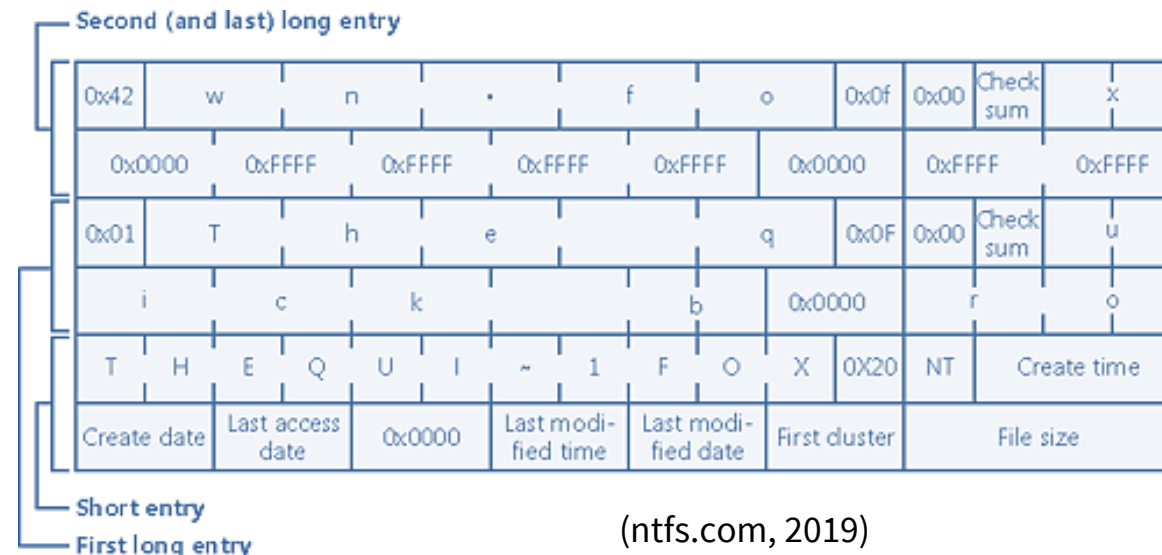
FAT

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00h	FFh	FFh	F8h	FFh	05h	00h	04h	00h	06h	00h	08h	00h	07h	00h	09h	00h
10h	FFh	FFh	0Ah	00h	FFh	FFh	00h	00h	00h	00h	00h	00h	00h	00h	00h	00h
20h	00h	00h	FFh	FFh	00h	00h	00h	00h	00h	00h	00h	00h	00h	00h	00h	00h
30h																

Datové klastry

VFAT a dlouhé názvy

- ❑ Windows 95, Windows 95 OSR2
- ❑ Virtuální alokační tabulka.
 - Dlouhá jména - 255 znaků
 - Zpětná kompatibilita



- Jméno 8+3 znaky
- Vytváří pro dlouhý název jeden nebo více sekundárních záznamů
 - jeden pro každých 13 znaků v dlouhém názvu souboru
 - Unicode
- ❑ Windows nastavuje atributy *svazek, jen pro čtení, systémový a skrytý soubor* u sekundárního záznamu - interpretováno jako součást dlouhého názvu
 - MS-DOS obecně ignoroval položky složky se všemi čtyřmi těmito atributy

(ntfs.com, 2019)

Vlastnosti FAT32

- ❑ Specifikace souborového systému
 - Souborový systém FAT32 obsahuje čtyři bajty na klastr v alokační tabulce souborů.
 - Používá pouze 28 bitů. Horní 4 bity jsou rezervovány.
- ❑ Boot Sector and Bootstrap
 - Obsahuje více vyhrazených sektorů. Počet je obvykle 32 sektorů (může se lišit).
 - Spouštěcí záznam na jednotkách FAT32 je větší než 1 sektor.
- ❑ Zrcadlení alokační tabulky
 - Zrcadlení FAT lze deaktivovat a primární kopii může být jiná než první FAT
- ❑ Kořenový adresář
 - Kořenový adresář se může podle potřeby rozšiřovat
 - je běžný řetězec clusterů.

(ntfs.com, 2019)

Souborový systém exFAT

- ❑ Nástupce rodiny systémů souborů FAT (FAT12 / 16 / 32).
- ❑ Má podobný design a mnoho významných vylepšení:
 - Větší limity objemu a velikosti souboru
 - Velikost svazku 128 PiB, doporučeno 512 TiB
 - Velikost souboru 128 PiB (teoreticky 16 EiB – 1, protože velikost souboru uložena jako 64bit číslo)
 - Maximální velikost clusteru 32 MiB, maximální počet clusterů 2^{32}
 - Nativní názvy souborů v Unicode
 - Větší spouštěcí oblast umožňující větší spouštěcí kód
 - Alokační bitmapa, UpCase tabulka
 - Lepší výkon
 - Podpora offsetu časového pásma
 - Podpora OEM parametrů

(Microsoft, 2019; ntfs.com, 2019)

Boot Sector of ExFAT

WinHex - [Drive F:]

File Edit Search Position View Tools Specialist Options Window Help

Drive F: [unregistriert] 100% free exFAT

Default Edit Mode State: original

Undo level: 0

Undo reverses: n/a

Physical sector No.: 63

Logical sector No.: 0

Used space: 192 KB 196,608 bytes

Free space: 3.7 GB 4,013,096,960 bytes

Total capacity: 3.7 GB 4,013,904,384 bytes

Bytes per sector: 512

Sector count: 7,839,657

Physical disk: 2

Mode: hexadecimal

Character set: ANSI ASCII

Offsets: hexadecimal

Bytes per page: 46x16=736

Window #: 1

No. of windows: 1

Offset: 0 1 2 3 4 5 6 7 8 9 A B C D E F

00000000 EB 76 90 45 58 46 41 54 20 20 20 00 00 00 00 00

00000001 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000002 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000003 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000004 3F 00 00 00 00 00 00 00 A9 9F 77 00 00 00 00 00

00000005 80 00 00 00 C0 03 00 00 80 04 00 00 6C DE 01 00

00000006 04 00 00 00 FF 02 56 74 00 01 00 00 09 06 01 80

00000007 00 00 00 00 00 00 00 00 33 C9 8E D1 BC F0 7B 8E

00000008 00 00 00 00 00 00 00 00 AC 98 40 74 0C 48 74 0E

00000009 00 00 00 00 00 00 00 00 EF A0 FD 7D EB E6 CD 16

0000000A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0000000B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0000000C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0000000D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0000000E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0000000F 0D 0A 52 65 6D 6F 76 65 20 64 69 61 2E FF 0D

00000010 72 28 6F 74 68 65 72 20 6D 65 64 69 61 2E FF 0D

00000011 0A 44 69 73 6B 20 65 72 72 6F 72 FF 0D 0A 50 72

00000012 65 73 73 20 65 73 74 61 65 73 74 61 65 73 74 61

00000013 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000014 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000015 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000016 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000017 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000018 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000019 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0000001A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0000001B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0000001C FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

0000001D FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

0000001E FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

0000001F FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

00000200 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00000210 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Boot sector snap shot of ExFAT file system in 4GB USB stick

File system name "ExFAT"

Total number of sectors

Number of clusters

Number of FATs

Number of sectors per cluster

$2^6 = 64$ sectors

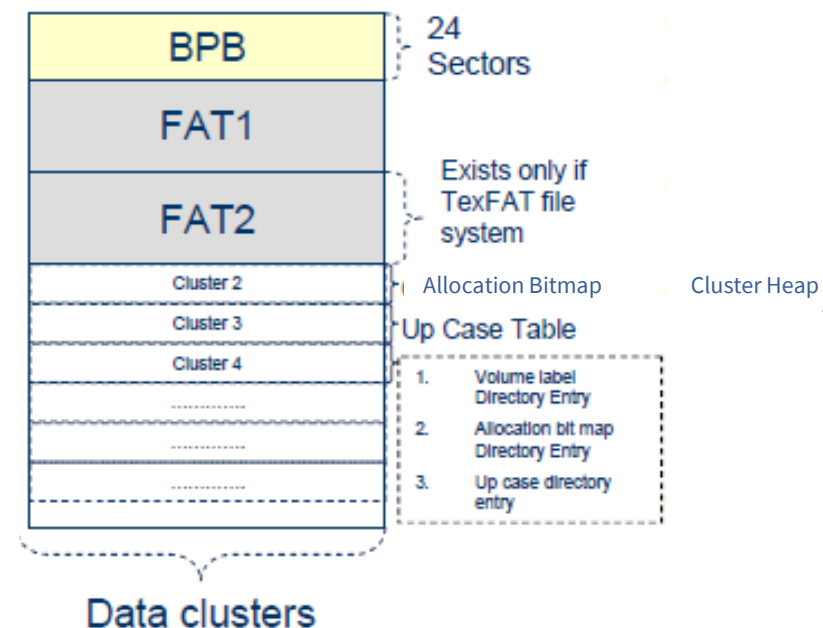
Number of Bytes per sector

$2^9 = 512$ bytes

Cluster size = $2^6 * 2^9 = 2^{15} = 32\text{KB}$

Struktura exFAT

- ❑ 12 sektorů Boot Area + dalších 12 sektorů pro zálohu
- ❑ FAT2 existuje, pouze pokud je použit TexFAT (Transactional FAT)
- ❑ První datové klastry počínaje klastrem s indexem 2 obsahují
 - Bitmapovou tabulku přidělení klastrů
 - Up-case tabulka
- ❑ Další cluster obsahují adresářové záznamy
 - Záznam Volume label
 - Záznam Bitmapové tabulky
 - Záznam Up-case tabulky
 - Záznamy souborů a adresářů
 - Každý 3 až 19 záznamů o velikosti 32B
 - **File / Directory Entry (1)** - atributy
 - **Stream Extension Entry (1)** – informace o umístění
 - **File Name Entry (1-17)** – jméno, max. 15 znaků na záznam



Alokační tabulka

- ❑ Alokační tabulka (FAT) existuje v 1 nebo 2 kopiích
 - definované v poli *NumberOfFats*.
 - Pole *ActiveFat* ve *VolumeFlags* v hlavním spouštěcím sektoru určuje, která FAT je aktivní.
- ❑ První klastr je s indexem 2 (podobně jako ve FAT32).
 - Každá položka FAT představuje jeden cluster.
- ❑ V exFAT se FAT nepoužívá pro sledování alokace
 - pro tento účel se používá alokační bitmapa.
 - FAT se používá pouze k udržování informací o souboru (atributy, jméno, umístění).
- ❑ Druhá tabulka FAT (používá se pouze v TexFAT) je umístěna bezprostředně za první a má stejnou velikost.

(Microsoft, 2019; ntfs.com, 2019)

Alokační tabulka

- Platné hodnoty záznamů FAT:
 - 0x00000002 - *ClusterCount* + 1 (max. 0xFFFFFFFF6) - další klastr v řetězci
 - 0xFFFFFFFF7 - chybný cluster
 - 0xFFFFFFFFF - konec souboru (značka EOF)
 - 0x00000000 neznamená, že je klastr volný
 - je to nedefinovaná hodnota.

(Microsoft, 2019; ntfs.com, 2019)

Alokační bitmapa

- ❑ Alokační bitmapa sleduje stav přidělování klastrů.
- ❑ Tabulka alokace bitmap je umístěna v haldě
- ❑ Alokační bitmapa se skládá z počtu 8 bitových bajtů, které lze považovat za posloupnost bitů.
 - Každý bit v bitmapě odpovídá datovému klastru.
 - Hodnota 1 znamená, že je klaster obsazen.
 - Hodnota 0 znamená, že klaster je volný.
- ❑ Nejméně významný bit bitmapové tabulky se týká prvního klastru.
- ❑ V TexFAT mohou být 2 Alokační bitmapy, jinak jen jedna.

(Microsoft, 2019; ntfs.com, 2019)

Up-case tabulka

- ❑ Záznam názvu souboru (File Name Entry) používá Unicode
 - zachovává velikost písmen při ukládání názvu souboru.
- ❑ Tabulka up-case obsahuje data pro převod z malých na velká písmena.
- ❑ ExFAT nerozlišuje velká a malá písmena
- ❑ Při vyhledávání porovnává názvy souborů převedené na velká písmena.
- ❑ Tabulka up-case je pole znaků Unicode, jejichž index představuje znaky Unicode, které mají být verzálky a hodnota je cílový znak.
- ❑ Tabulka Up-case se obvykle nachází hned za tabulkou Alokační bitmapou.

Index	hodnota	Popis
0x061	0x041	'a' is mapped into 'A'

Struktura adresáře

- ❑ Soubor je reprezentován 32bajtovými adresářovými záznamy.
 - Sada záznamů - directory file set (entry set)
 - 3 až 19 záznamů
 - **File / Directory Entry (1)** - atributy
 - **Stream Extension Entry (1)** – informace o umístění
 - **File Name Entry (1-17)** – jméno, max. 15 znaků na záznam
- ❑ Podadresář pojme až 2 796 202 sad souborů. (max. 256 MiB)
 - Pokud mají soubory delší názvy, toto číslo se adekvátně sníží
 - exFAT používá stromovou strukturu k popisu vztahu mezi soubory a adresáři.
 - Podadresáře jsou propojeny s rodiči.
 - Neexistují speciální (.) a (..) adresáře směřující na sebe a na rodiče jako ve FAT16 / FAT32.

Field name	Ofs	Size	Function
EntryType	0	1	0x85 (File and directory entry).
SecondaryCount	1	1	Number of following entries. The size of this entry set is <code>SecondaryCount + 1</code> entries.
SetChecksum	2	2	A 16-bit checksum to validate this entry set. The checksum of an entry set is calculated in byte-by-byte as a simple byte array excluding only this field.
FileAttribute	4	2	File attribute in combination of following flags. The meaning of these flags are same as FAT filesystem. bit0: Read-Only. bit1: Hidden. bit2: System. bit3: Reserved (0). bit4: Directory. bit5: Archive. bit6-15: Reserved (0).
Reserved1	6	2	Reserved (0).
CreateTimestamp	8	4	Timestamp of the file created. Upper 16 bits contains the local date and lower 16 bits contains the local time. Each fields in the date and time are same as FAT filesystem.
LastModifiedTimestamp	12	4	Timestamp of the file modified.
LastAccessedTimestamp	16	4	Timestamp of the file accessed.
Create10msIncrement	20	1	Sub-second information of <code>CreateTimestamp</code> in unit of 10 ms (from 0 to 199).
LastModified10msIncrement	21	1	Sub second information of <code>LastModifiedTimestamp</code> .
CreateTZOffset	22	1	Time zone offset of <code>CreateTimestamp</code> in unit of quarter hour with ORed by 0x80. For example, $+9 * 4 \mid 0x80 = 0xA4$ for JST(+9:00), $-7 * 4 \mid 0x80 = 0xE0$ for PST(-7:00). The timestamp in UTC can be got by this field. Set 0x00 when time zone is not used.
LastModifiedTZOffset	23	1	Timezone offset of <code>LastModifiedTimestamp</code> .
LastAccessedTZOffset	24	1	Timezone offset of <code>LastAccessedTimestamp</code> .
Reserved2	25	7	Reserved (0).

Field name	Ofs	Size	Function
EntryType	0	1	0xC0 (Stream extension entry).
GeneralSecondaryFlags	1	1	<p>These flags indicate status of the file allocation.</p> <p>bit0(AllocationPossible): 0 = Cluster allocation is not possible and FirstCluster and DataLength field are undefined, 1 = Cluster allocation is possible and FirstCluster and DataLength field are valid as defined.</p> <p>bit1(NoFatChain): 0 = Cluster chain on the FAT is valid, 1 = Cluster chain is contiguous and not recorded on the FAT.</p> <p>bit2-15: Reserved (0).</p> <p>If AllocationPossible flag is 0 (actually always 1), ValidDataLength, FirstCluster, DataLength are invalid. When cluster chain is contiguous, NoFatChain can be set and any recording on the FAT is not needed.</p>
Reserved1	2	1	Reserved (0).
NameLength	3	1	Length of the file name in UTF-16 encoding unit. Valid value for this field is 1 to 255.
NameHash	4	2	16-bit checksum of the up-case converted file name. The value is calculated in byte-by-byte as a UTF-16LE string. This enables to skip comparison at most unmatched file when look for a file in the directory.
Reserved2	6	2	Reserved (0).
ValidDataLength	8	8	Valid data length of the file in unit of byte. It indicates how long the data actually written. Valid value for this field is 0 to DataLength. Any data located beyond this offset is undefined and zeros should be returned if read it. This is the feature to implement fallocation() efficiently. As for the sub-directories, the value must be equal to DataLength.
Reserved3	16	4	Reserved (0).
FirstCluster	20	4	First cluster number of the cluster chain. When DataLength is zero, also this value must be zero and the file does not have cluster chain.
DataLength	24	8	Actual length of the file in unit of byte. Size of sub-directory is always multiple of cluster size.

Directory Entry set

Field name	Ofs	Size	Function
EntryType	0	1	0xC1 (File name entry).
GeneralSecondaryFlags	1	1	Always 0.
FileName	2	30	Contains file name string in UTF-16LE. If the name length is less than 15, rest of space in this field is filled with zero. If the name length exceeds 15 characters, multiple file entries ($(\text{NameLength} + 14) / 15$) are used and set in the entry block in ascending order.

(Source: http://elm-chan.org/docs/exfat_e.html#dir_struct)

- Libovolný znak, vyjma řídících znaků
 - (U+0000 až U+001F, U+007F)
 - " * / : < > ? \ |

NTFS (New Technology File System)

❑ Souborový systém Windows

- Max velikost oddílu 16 TiB (4 kiB klastr), 256 (64 kiB klastr)
- Max počet souborů 4,294,967,295 ($2^{32} - 1$)
- Max velikost souboru 2^{64} (16 ExaBytes) minus 1KB

❑ Podporuje dlouhé názvy souborů

- Vytváří i druhý 8.3 krátký název souboru.
- Délka názvu může mít až 255 znaků a může obsahovat mezery, více teček a speciální znaky, kromě:
 - `<> "/ \: | *? a některá speciální jména jako PRN, CON, NUL, ...`
- Podpora POSIX
 - Podpora case-sensitive (WinAPI nepodporuje)
 - Hard linky – soubor může mít víc jak jedno jméno.

(NTFS.com, 2019; Microsoft, 2009)

Vlastnosti NTFS

- ❑ Oprávnění k objektům - vestavěné funkce zabezpečení, jako jsou oprávnění k souborům a složkám
- ❑ The Encrypting File System (EFS) - šifrování na svazku NTFS.
- ❑ Podpora šifrování jednotky BitLocker
- ❑ Komprese
- ❑ Diskové kvóty
- ❑ Odkazové body (ReparsePoints)
 - Programy mohou sledovat požadavky na otevření jednotlivých souborů a před skutečným otevřením mohou provádět nějaký kód.
 - Toto může být využito k rozšíření vlastností souborového systému.
 - Např. montování svazků k adresářům.
- ❑ „Řídké" soubory (Sparse Files)
 - umožňuje vytvářet obrovské soubory, neobsahující žádná data; prostor je pro daný soubor vyhrazen a je využíván podle potřeby.
- ❑ Násobné datové proudy

(NTFS.com, 2019; Microsoft, 2009)

NTFS

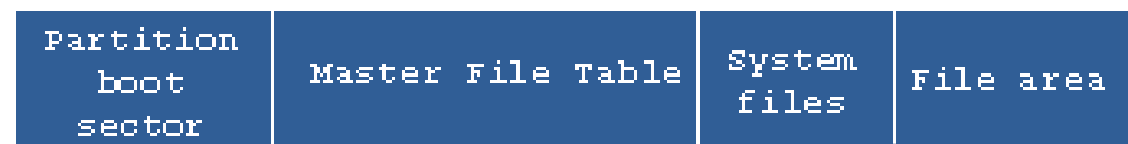
- ❑ Na začátku oddílu jsou uloženy systémové soubory
 - Tzv. metadata
- ❑ Vznikají při formátování svazku
 - Celkem 12 souborů
 - další 4 vyhrazeny pro budoucí použití
 - Plus další volitelné
- ❑ Nejsou normálně viditelné



(NTFS.com, 2019; Microsoft, 2009)

Struktura NTFS

- ❑ Zaváděcí sektor NTFS (Boot Sector)
 - Obsahuje blok parametrů BIOSu, který ukládá informace o rozvržení svazku a strukturách souborového systému (umístění \$MFT), plus spouštěcí kód.
 - Začíná na sektoru 0, délka až 16 sektorů.
- ❑ Master File Table
 - Obsahuje informace o všech souborech (atributy)
 - Kopie Master File Table
 - Zrcadlení prvních 16 MFT záznamů
- ❑ Data souborového systému
 - Ukládá data, která nejsou obsažena v Master File Table



(NTFS.com, 2019; Microsoft, 2009)

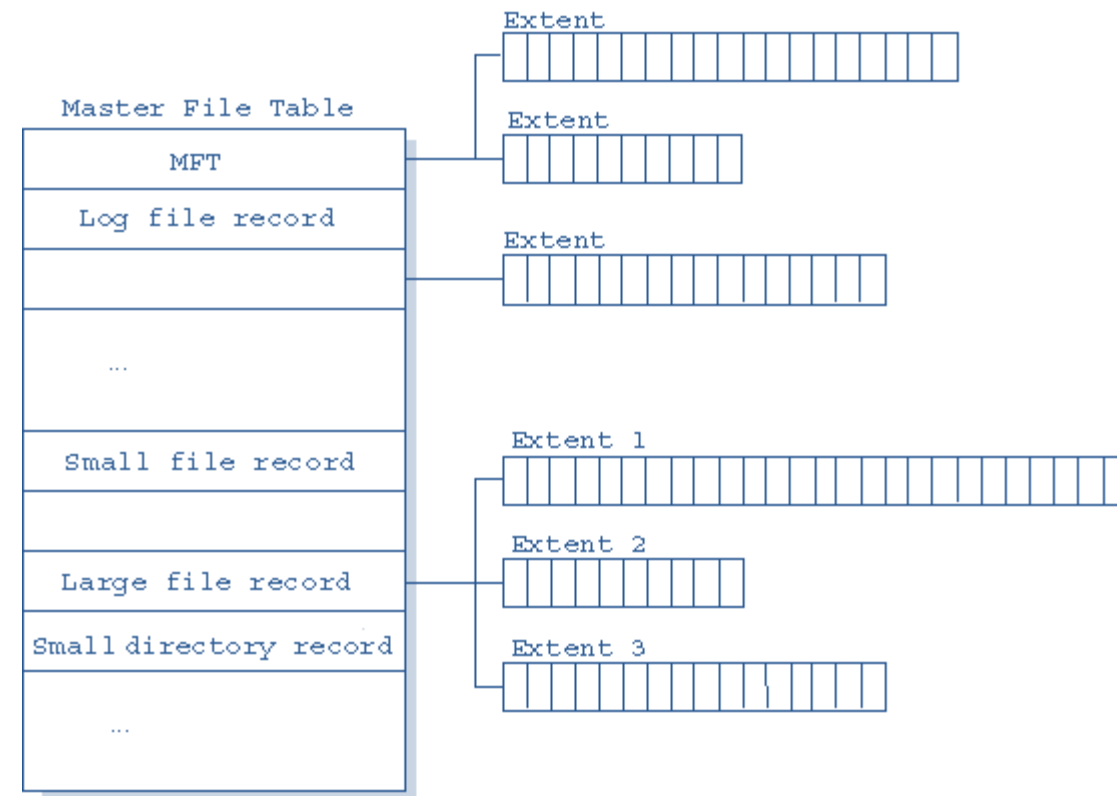
(NTFS.com, 2019; Microsoft, 2009)

Metadata NTFS

Jméno v MFT		Popis
\$MFT		Master File Table
\$MFTMIRR		Kopie prvních záznamů NTFS
\$LOGFILE		Transakční soubor
\$VOLUME		Informace o oddílu
\$ATTRDEF		Tabulka atributů (jméno, čísla, popis)
.		Kořenový adresář
\$BITMAP		Mapa použití klastrů
\$BOOT		Boot record
\$BADCLUS		Informace o vadných klastrech, přemapování klastrů.
\$QUOTA		Obsahuje limity využití souborového systému pro uživatele
\$UPCASE		Konverze malých písmen na velká (verzálky), UNICODE
\$Extend		Adresář pro různá rozšíření NTFS
\$Secure		Jedinečné deskriptory zabezpečení pro všechny soubory

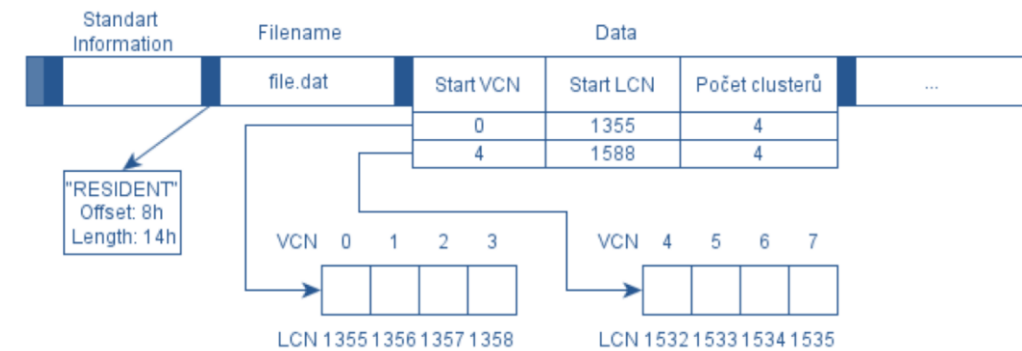
NTFS – Master File Table

- ❑ MFT je relační databáze
 - Skládá se ze záznamů o souborech (řádky) a atributů souboru (sloupce).
 - Obsahuje alespoň jednu položku pro každý soubor na svazku, včetně samotného MFT.
 - Pokud jsou informace o alokaci pro soubor nebo adresář příliš velké, aby se vešly do jednoho záznamu, alokují se další záznamy
 - NTFS si ve výchozím nastavení vyhrazuje 12,5 procenta objemu pro výhradní použití MFT.



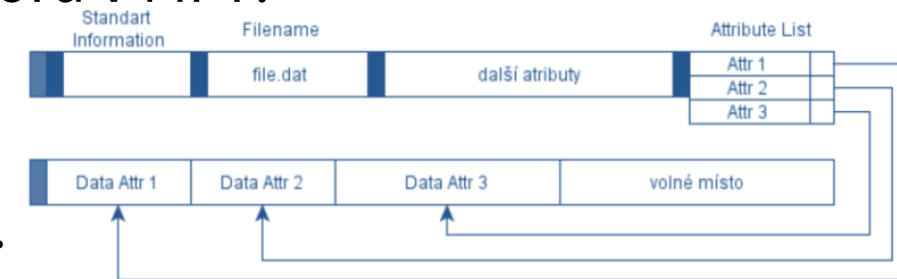
MFT Structure (NTFS.com, 2019)

NTFS – záznam MFT



□ Každý záznam je tvořen množinou atributů souboru. (Šváb, 2012)

- Atributy souboru se zapisují do přiděleného prostoru v MFT.
 - Rezidentní/Nerezidentní
 - reprezentovány číselným kódem s referencí na \$AttrDef
- Záznamy o souborech a složkách mají každý 1 KB.
 - Malé soubory a adresáře (obvykle až 512 B) mohou být zcela obsaženy v MFT.



(Šváb, 2012)

- Záznamy adresářů jsou také umístěny v MFT
- Místo dat obsahují adresáře indexové informace.
 - Velké adresáře jsou organizovány do B-stromů
 - Záznamy s ukazateli na externí klastry obsahující položky adresáře

(NTFS.com, 2019)

Atributy NTFS

Typ atributu	Popis
Standard Information	Zahrnuje informace jako např. časové razítko a počet linků.
Attribute List	Seznam umístění všech atributů záznamu, které se nevlezly do hlavního záznamu v MFT.
Filename	Jméno souboru. Krátký (8.3, case-insensitive) i dlouhý název (255, UNICODE).
Security Descriptor	Vlastník souboru a práva k přístupu.
Data	Data souboru.
Object ID	Jedinečný identifikátor souboru.
Reparse Point	Používá se jako přípojný bod oddílu.
Index Root	
Index Allocation	Používá se na implementaci adresářů a dalších indexů
Bitmap	
Volume Information	Verze oddílu (pouze u metadata souboru \$Volume)
Volume Name	Jméno oddílu (pouze u metadata souboru \$Volume)

NTFS Log File

- ❑ Obsahuje seznam transakčních protokolů změn metadat FS.
 - Transakční kroky používané pro obnovu NTFS.
 - Používá se k obnovení konzistence NTFS po selhání systému.
- ❑ Velikost souboru protokolu závisí na velikosti svazku a může být až 4 MiB.
- ❑ Obnova souborů protokolu
 - REDO - zopakujte neúspěšnou transakci
 - UNDO - navrácení zpět transakce, která selhala.

(NTFS.com, 2019; Microsoft, 2009)

Násobné datové proudy (Multiple Data Streams)

- ❑ Každý soubor má hlavní nepojmenovaný proud
- ❑ Alternativní datový proud
 - myfile.dat: stream2
- ❑ Datový proud je jedinečná sada atributů souboru.
 - Proudý mají samostatné zámky, ale společná oprávnění.
- ❑ Při kopírování souboru NTFS do non-NTFS dojde ke ztrátě datových toků a dalších nepodporovaných atributů.

```
C:\Users\sysel>echo text > program:sourcefile

C:\Users\sysel>dir program
Volume in drive C is System
Volume Serial Number is 0663-4948

Directory of C:\Users\sysel

20. 04. 2019  13:58                0 program
                1 File(s)                0 bytes
                0 Dir(s)  96 663 453 696 bytes free

C:\Users\sysel>more < program:sourcefile
text
```

(NTFS.com, 2019; Microsoft, 2009)

Přístupová práva

- Základní oprávnění jsou
 - **Full Control** - prohlížení, spuštění, úpravy, mazání a nastavení oprávnění
 - **Modify** - prohlížení, spuštění, úpravy a mazání
 - **Read & Execute** - prohlížení a spuštění
 - **List Folder Content** - zobrazení obsahu adresáře, pouze pro adresáře
 - **Read** - prohlížení
 - **Write** - prohlížení, spuštění, úpravy a mazání

Permissions	Basic Full Control	Basic Modify	Basic Read & Execute	Basic List Folder Contents	Basic Read	Basic Write
Travers Folder/Execute File	×	×	×	×		
List Folder/ Read Data	×	×	×	×	×	
Read Attributes	×	×	×	×	×	
Read Extended Attributes	×	×	×	×	×	
Create Files/Write Data	×	×				×
Create Folders/Append Data	×	×				×
Write Attributes	×	×				×
Write Extended Attributes	×	×				×
Delete Subfolders and Files	×					
Delete	×	×				
Read Permissions	×	×	×	×	×	×
Change Permissions	×					
Take Ownership	×					
Synchronize	×	×	×	×	×	×

Nastavení pokročilých oprávnění

Pokročilá oprávnění:

- ☒ Úplné řízení
- ☒ Procházet složkou / Spouštět soubory
- ☒ Zobrazovat obsah složky / Číst data
- ☒ Číst atributy
- ☒ Číst rozšířené atributy
- ☒ Vytvářet soubory / Zapisovat data
- ☒ Vytvářet složky / Připojovat data
- ☒ Zapisovat atributy
- ☒ Zapisovat rozšířené atributy
- ☒ Odstraňovat
- ☒ Číst oprávnění
- ☒ Měnit oprávnění
- ☒ Přebírat vlastnictví

Pokročilá oprávnění

❑ Procházet složkou/Spouštět soubory

- Možnost Procházet složkou povoluje nebo zakazuje průchod složkami k dalším souborům či adresářům i v případě, kdy nemá uživatel k procházeným složkám oprávnění. Oprávnění Procházet složkou je uplatněno pouze tehdy, pokud uživatel nebo skupina nemá uživatelské právo **Vynechat kontrolu přecházení** v Konzole pro správu zásad skupiny. Uživatelské právo **Vynechat kontrolu přecházení** je ve výchozím nastavení uděleno skupině Everyone. (Platí pouze pro složky.)
- Oprávnění Spouštět soubory povoluje či zakazuje spouštění programových souborů. (Platí pouze pro soubory.)
- I když nastavíte pro danou složku oprávnění Procházet složkou, neznamena to, že pro všechny soubory dané složky je nastaveno oprávnění Spouštět soubory automaticky.

❑ Zobrazovat obsah složky/Číst data

- Možnost Zobrazovat obsah složky povoluje či zakazuje zobrazování jmen souborů a podadresářů v rámci dané složky. Možnost Zobrazovat obsah složky má platnost pouze na obsah této složky a nemá vliv na zobrazení názvu složky, u které nastavíte toto oprávnění. (Platí pouze pro složky.)
- Možnost Číst data povoluje či zakazuje zobrazování dat souborů. (Platí pouze pro soubory.)

❑ Číst atributy

- Povoluje či zakazuje zobrazování atributů souboru nebo složky (pouze pro čtení, skrytý atp.). Atributy jsou určeny systémem souborů NTFS.

❑ Číst rozšířené atributy

- Povoluje či zakazuje zobrazování rozšířených atributů souboru nebo složky. Rozšířené atributy jsou odvislé od konkrétního programu. Mohou se program od programu lišit.

Pokročilá oprávnění

- ❑ **Vytvářet soubory/Zapisovat data**
 - Možnost Vytvářet soubory povoluje nebo zakazuje vytváření souborů v rámci složky. (Platí pouze pro složky.)
 - Možnost Zapisovat data povoluje nebo zakazuje změny souborů a přepsání stávajícího obsahu souborů. (Platí pouze pro soubory.)
- ❑ **Vytvářet složky/Připojovat data**
 - Možnost Vytvářet složky povoluje nebo zakazuje vytváření složek v rámci původní složky. (Platí pouze pro složky.)
 - Oprávnění Připojovat data povolí nebo odepře přidávání dat na konec souboru, nikoli však změnu, odstranění či přepsání stávajících dat. (Platí pouze pro soubory.)
- ❑ **Zapisovat atributy**
 - Povoluje nebo zakazuje změnu atributů souboru nebo složky (pouze pro čtení, skrytý atp.). Atributy jsou určeny systémem souborů NTFS.
 - Oprávnění Zapisovat atributy v sobě nezahrnuje oprávnění k vytváření a odstraňování souborů nebo složek. Umožňuje pouze měnit atributy souborů a složek. Další informace o operacích povolení (či odepření) vytváření a odstraňování naleznete u položek **Vytvářet soubory/Zapisovat data, Vytvářet složky/Připojovat data, Odstraňovat podsložky a soubory a Odstraňovat**.
- ❑ **Zapisovat rozšířené atributy**
 - Povoluje či zakazuje změnu rozšířených atributů souboru nebo složky. Rozšířené atributy jsou odvislé od konkrétního programu. Mohou se program od programu lišit.
 - Oprávnění Zapisovat rozšířené atributy v sobě nezahrnuje oprávnění k vytváření a odstraňování souborů nebo složek. Umožňuje pouze měnit atributy souborů a složek.
 - Další informace o operacích povolení (či odepření) vytváření a odstraňování naleznete u položek **Vytvářet soubory/Zapisovat data, Vytvářet složky/Připojovat data, Odstraňovat podsložky a soubory a Odstraňovat**.

Pokročilá oprávnění

- ❑ **Odstraňovat podsložky a soubory**
 - Povoluje či zakazuje odstraňování podsložek a souborů, i když danému souboru či složce nebylo přiřazeno oprávnění Odstraňovat.
- ❑ **Odstraňovat**
 - Povoluje či zakazuje odstraňování souborů či složek. I když nemáte k dané složce nebo souboru přiřazeno oprávnění Odstraňovat, máte tuto možnost v případě, že bylo nadřazené složce přiřazeno oprávnění Odstraňovat podsložky a soubory.
- ❑ **Číst oprávnění**
 - Povoluje nebo zakazuje možnost číst oprávnění daného souboru či složky (Uplné řízení, Číst, Zapisovat atp.).
- ❑ **Měnit oprávnění**
 - Povoluje nebo zakazuje změnu oprávnění daného souboru či složky (Uplné řízení, Číst, Zapisovat atp.).
- ❑ **Přebírat vlastnictví**
 - Povoluje či zakazuje možnost převzít vlastnictví souboru či složky. Vlastník souboru nebo složky může vždy měnit příslušná oprávnění bez ohledu na stávající oprávnění souboru nebo složky.
- ❑ **Synchronizovat**
 - Povoluje či zakazuje podprocesy, které čekají na popisovač souboru či složky a synchronizují se s jiným podprocesem, který tuto událost signalizuje. Toto oprávnění lze aplikovat pouze na multithreadingové a multiprocesové programy.

(NTFS.com, 2019; Microsoft, 2009)

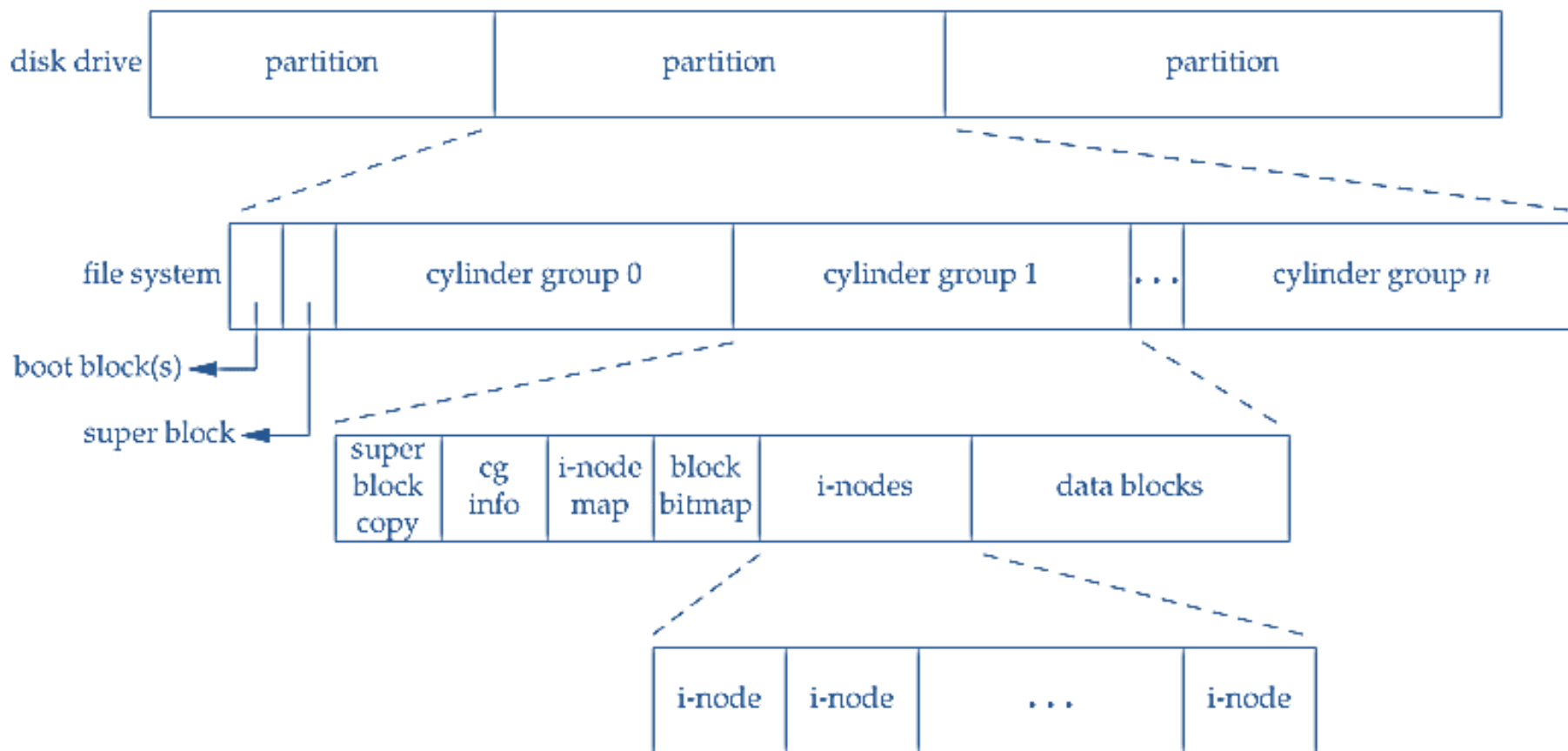
UNIX-like souborové systémy

- ❑ Operační systémy podobné Unixu vytvářejí virtuální souborový systém
- ❑ Všechno existuje v jedné hierarchii
 - Existuje pouze jeden kořenový adresář.
- ❑ Připojení souborového systému k bodu připojení - mountování
 - Filesystem Hierarchy Standard
- ❑ Linux podporuje mnoho souborových systémů
 - Rodina ext * (ext, ext2, ext3, ext4) - Extended file system, vychází z UFS
 - BTRFS
 - ZFS
 - JFS

Souborové systémy ext*

- ❑ Každý systém souborů má (alespoň) jeden superbloek.
 - Nachází se na začátku přiděleného úložiště.
 - obsahuje všechny informace o FS.
 - Info o konfiguraci: například velikost bloku, rozsah adres bloku, stav připojení, ukazatel na tabulku i-uzlů,...
 - Kopie superbloku jsou téměř vždy uloženy na několika dalších místech v rámci oddílu.
- ❑ Souborový systém má alespoň jednu tabulku, která popisuje soubory.
 - Položky v této tabulce se nazývají i-uzly (*i-nodes*).
 - Obsahuje atributy souboru a mapu označující, kde jsou datové bloky souboru umístěny na disku.

Struktura ext*



(source: http://poincare.matf.bg.ac.rs/~ivana/courses/ps/sistemi_knjige/pomocno/apue/APUE/0201433079/ch04lev1sec14.html)

Struktura ext*

- ❑ Redundantní kopie superbloku
- ❑ Redundantní kopie tabulky deskriptorů skupiny (cg informace)
- ❑ Bitmapa bloků
 - Obsahuje bit pro každý blok označující, zda je volný nebo je použit.
- ❑ Bitmapa i-uzlů
 - Obsahuje bit pro každý i-uzel označující, zda je volný nebo je použit.
- ❑ Tabulka i-uzlů pro i-uzly v této skupině
- ❑ Datové bloky

Superblock

- ❑ Superblock obsahuje všechny informace o konfiguraci FS.
 - položky, jako je celkový počet i-uzlů a bloků v souborovém systému a kolik je jich volných, kolik i-uzlů a bloků je v každé skupině, když byl souborový systém připojen, když byl změněn, jaká je verze souborového systému. Název svazku, jedinečné identifikační číslo, velikost i-uzlů a prostor pro volitelné funkce souborového systému pro ukládání konfiguračních informací.
 - Všechna položky jsou uložena na disku v little-endian formátu.
- ❑ Primární kopie je uložena v ofsetu 1024 B od začátku
 - První verze ext2 (revize 0) ukládala kopii na začátek každé skupiny válců
 - Pozdější verze šetří prostor a používají pouze specifické skupiny (0, 1 a mocniny 3, 5, 7).

(Poirier, 2011)

Skupiny (Cylinder Groups - Block Groups)

- ❑ Bloky jsou seskupeny do skupin bloků, aby se snížila fragmentace a minimalizoval počet přesunů hlavičky disku.
- ❑ Informace o každé skupině bloků jsou uchovávány v tabulce deskriptorů.
- ❑ Dva bloky blízko začátku každé skupiny jsou vyhrazeny pro bitmapu využití bloků a bitmapu využití i-uzlů
 - ukazují, které bloky a i-uzly se používají.
 - Protože je každá bitová mapa omezena na jeden blok, znamená to, že maximální velikost skupiny bloků je 8krát větší než velikost datového bloku (klastru).
- ❑ Bloky následující za bitmapami jsou označeny jako tabulka i-uzlů.
- ❑ Zbytek jsou datové bloky.

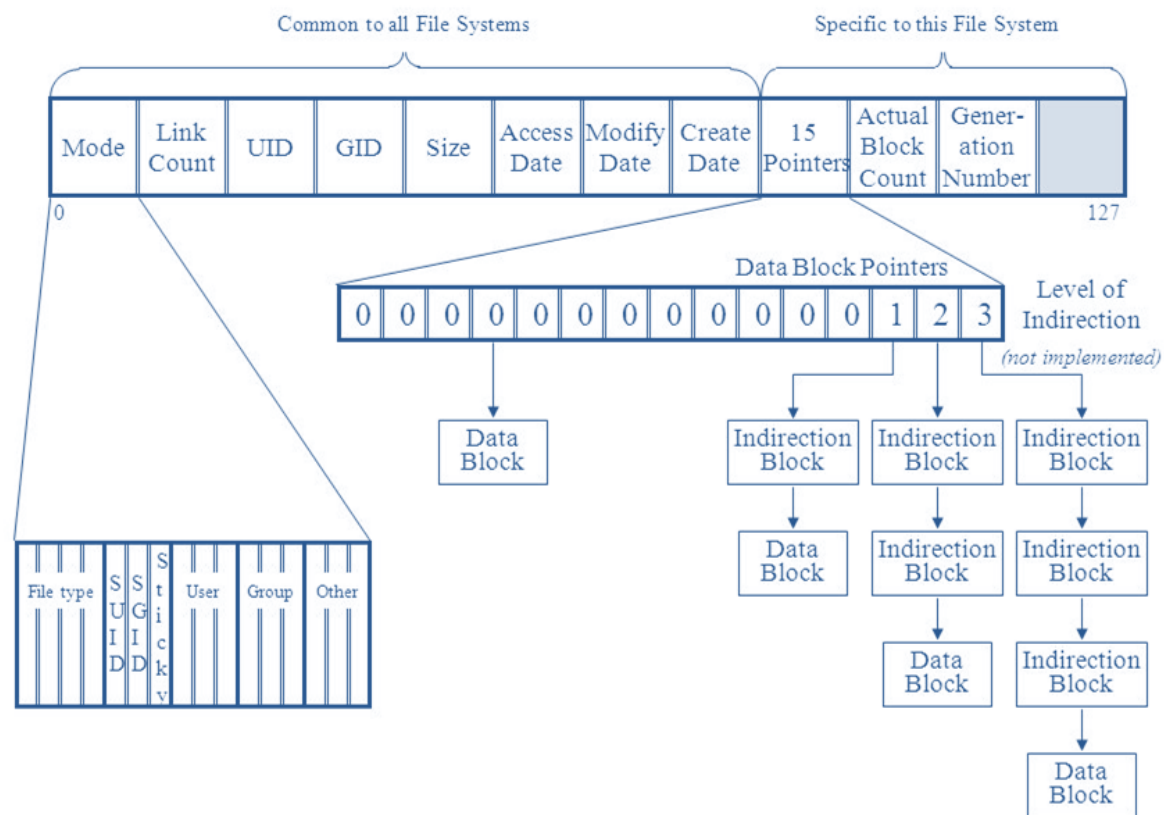
(Poirier, 2011)

i-uzel (i-node)

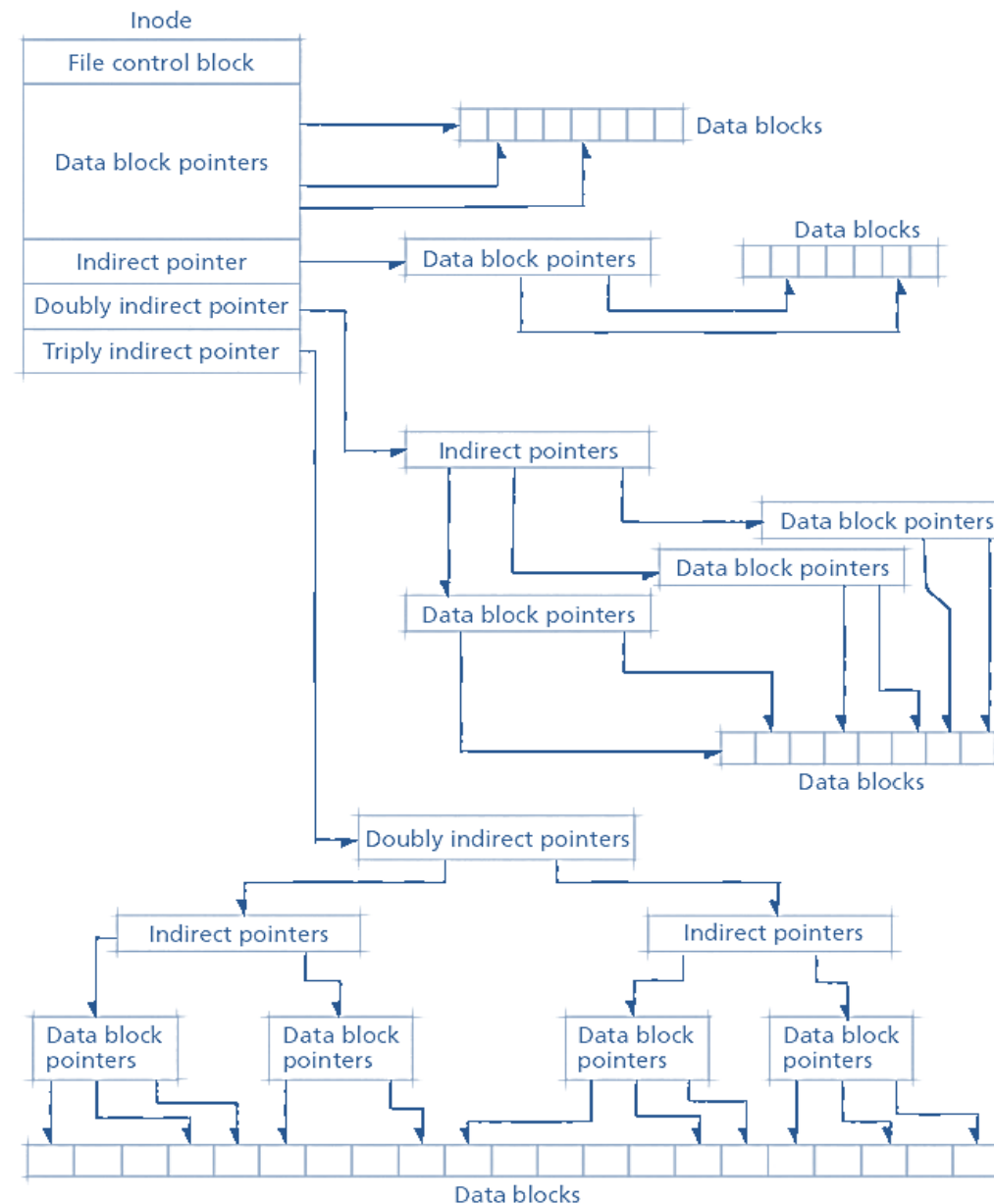
- ❑ Každý objekt v souborovém systému je reprezentován i-uzlem.
- ❑ Struktura i-uzlu obsahuje ukazatele na bloky souborového systému.
 - Obsahuje všechna metadata o objektu.
 - Kromě jeho názvu.
- ❑ Metadata o objektu zahrnují
 - oprávnění, vlastníka, skupinu, příznaky, velikost, počet použitých bloků, čas přístupu, čas změny, čas modifikace, čas odstranění, počet odkazů, fragmenty, verze (pro NFS) a rozšířené atributy (EA) a/nebo Access Control Lists (ACL).
 - Některé položky jsou nevyužité a některé přetížené.
 - ukazatele na prvních 12 bloků, které obsahují data souboru.
 - ukazatele na nepřímý blok, dvojnásobně nepřímý blok a trojnásobně nepřímý blok

Struktura i-uzlu

Disk Inode



UFS i-node (<https://slideplayer.com/slide/9035067/>)



i-node (Deitel, 2004)

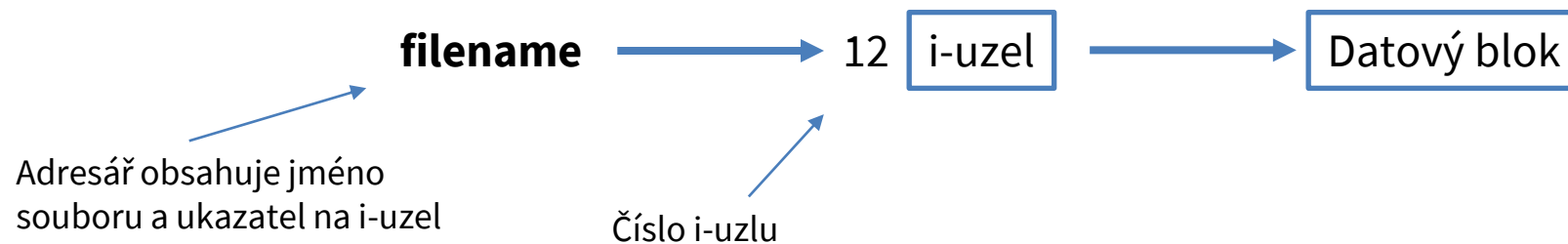
Datové bloky

- ❑ Velikost bloku se obvykle určuje při formátování disku
 - Velikost má vliv na výkon, maximální velikost souboru a maximální velikost systému souborů.
 - Obvykle 1 kiB, 2 kiB, 4 kiB nebo 8 kiB

Limits	1 KiB	2 KiB	4 KiB	8 KiB
file system blocks	2 147 483 647	2 147 483 647	2 147 483 647	2 147 483 647
blocks per block group	8 192	16 384	32 768	65 536
inodes per block group	8 192	16 384	32,768	65 536
bytes per block group	8MiB	32MiB	128MiB	512MiB
file system size	4TiB	8TiB	16TiB	32TiB
file size	16GiB	256GiB	2TiB	2TiB

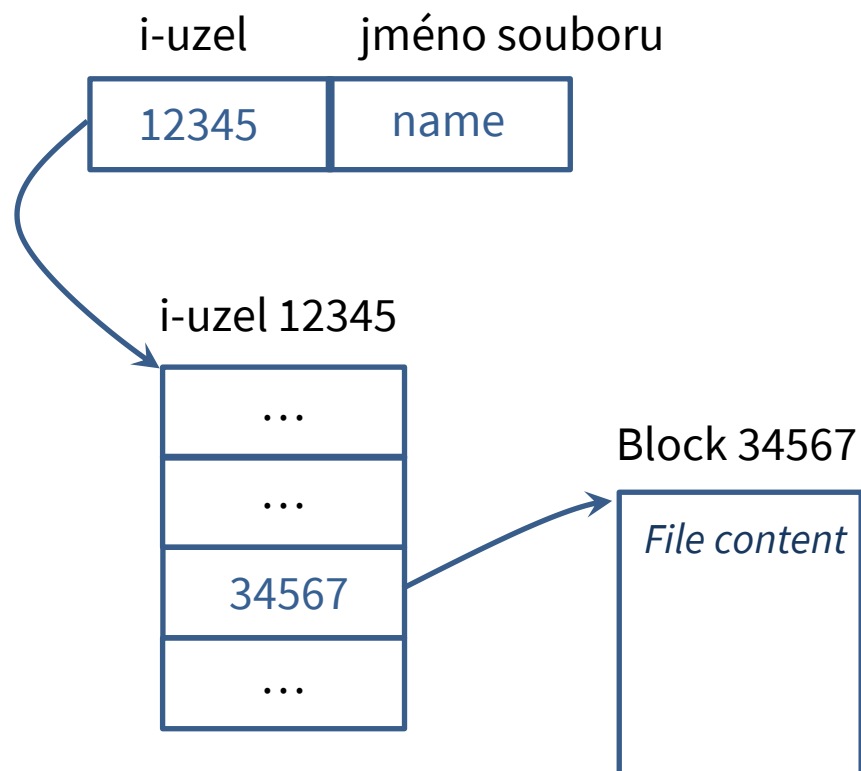
Adresář

- ❑ Adresář je objekt souborového systému a má i-uzel stejně jako soubor.
 - Jedná se o speciálně formátovaný soubor obsahující záznamy (jména s číslem i-uzlu).
 - Pozdější revize také vkládají typ objektu (soubor, adresář, symbolický odkaz, zařízení, fifo, socket), aby se zabránilo nutnosti kontrolovat i-uzel.
- ❑ Původní revize ext2 použila k uložení názvů souborů do adresáře propojený seznam; novější revize jsou schopné používat hash a binární stromy.

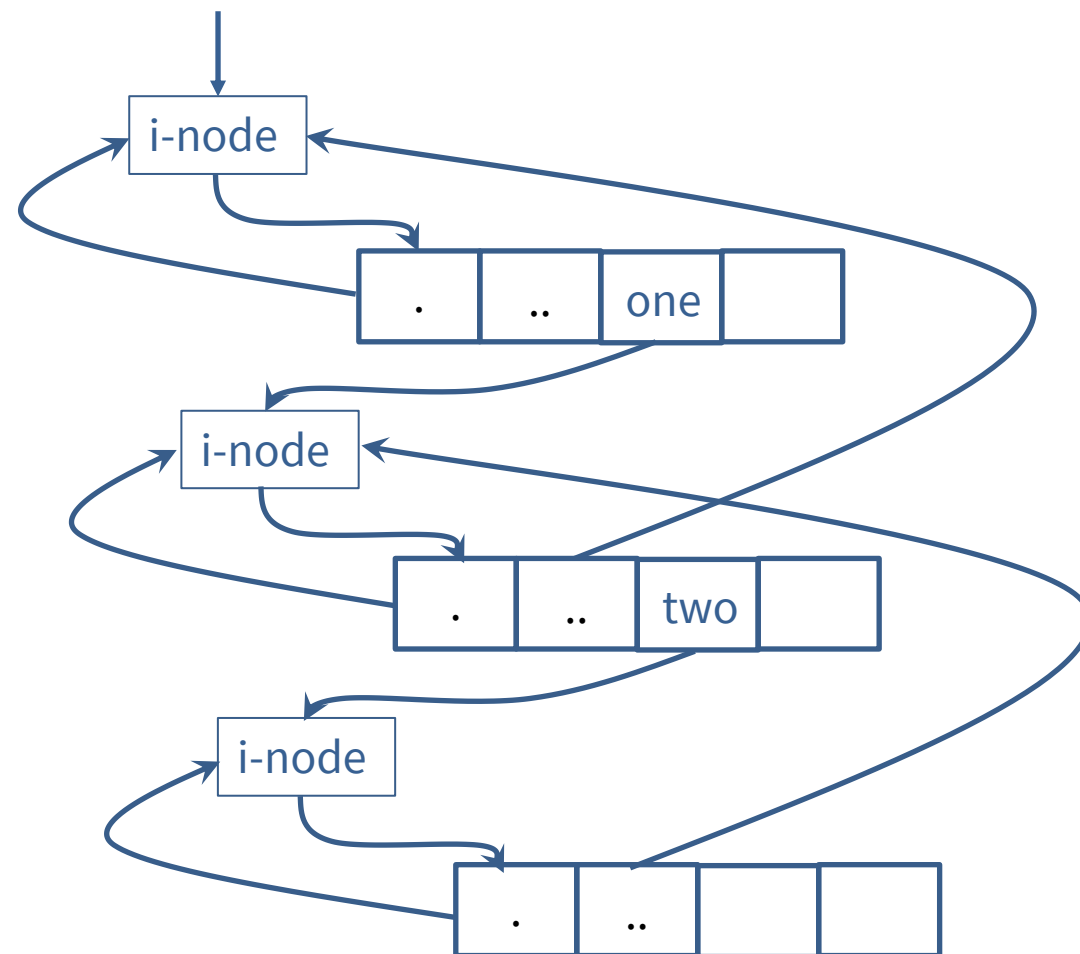


Adresáře

Záznam adresáře v /dir

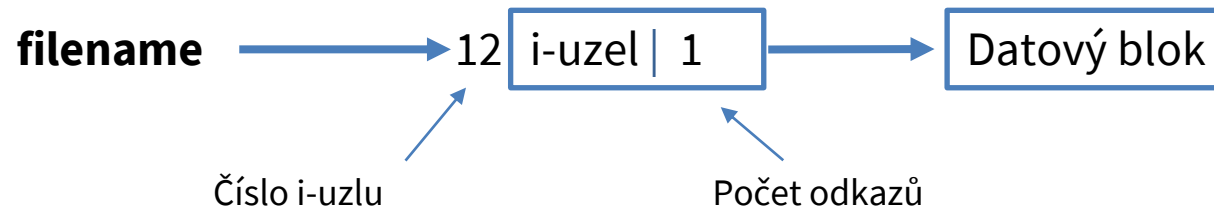


i-uzel, soubor, ukazatele

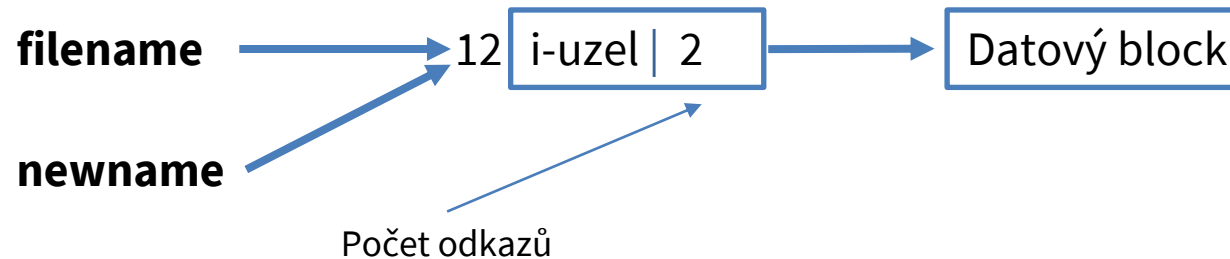


i-uzel, adresáře, ukazatele

HardLink



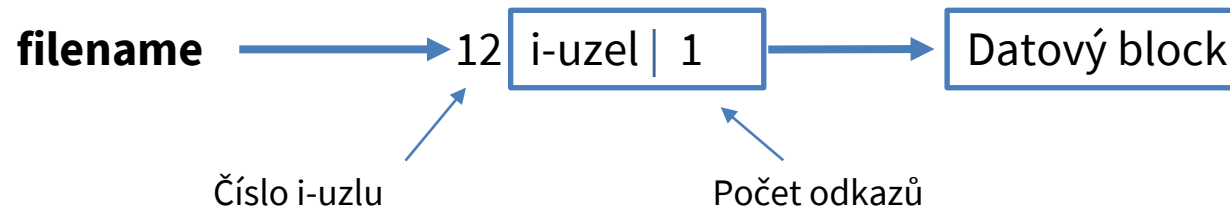
❑ \$ln filename newname



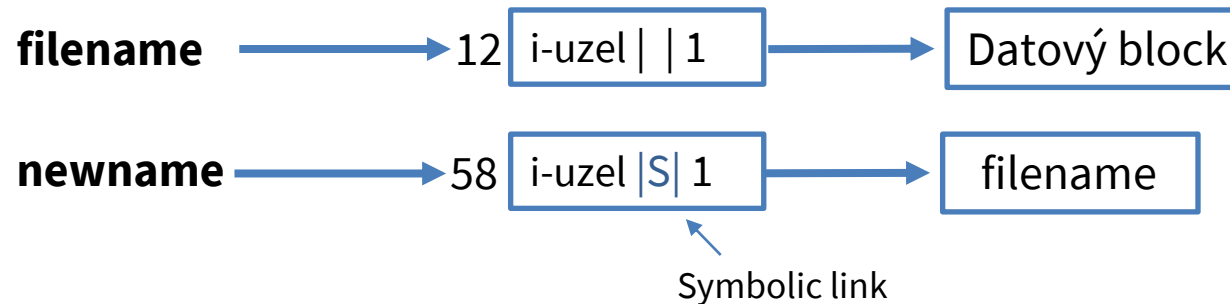
❑ \$ls -l

```
-rw-r--r-- 2 student users 5 Mar 10 21:25 filename
-rw-r--r-- 2 student users 5 Mar 10 21:25 newname
```

Symbolický odkaz (Symlink, SoftLink)



❑ `$ln -s filename newname`



❑ `$ ls -l`

<code>-rw-r--r--</code>	<code>1</code>	<code>student users</code>	<code>5</code>	<code>Mar 10 21:25</code>	<code>filename</code>
<code>lrwxrwxrwx</code>	<code>1</code>	<code>student users</code>	<code>8</code>	<code>Mar 10 21:25</code>	<code>newname -> filename</code>

Souborový systém Ext3

- ext3 = ext2 + žurnálování
 - Zvyšuje spolehlivost a eliminuje nutnost kontroly FS po neplánovaném vypnutí.
- ext3 přidává následující funkce k ext2:
 - žurnálování
 - Tři úrovně
 - Online rozšiřování souborového systému
 - HTree indexování pro větší adresáře

Žurnálování

❑ Writeback

- nejrychlejší varianta - uchovává pouze změny metadat
- zajišťuje jen limitovanou schopnost zachování integrity dat .
 - Obsah může být zapsán před nebo po zápisu do žurnálu. Stejná bezpečnost jako ext2

❑ Ordered

- uchovává pouze změny metadat, ale změny metadat jsou uskutečněny, až po změnách ve filesystému.
- je defaultním módem a garantuje konzistenci dat ve filesystému

❑ Journal

- zapisujeme do žurnálu data i metadata.
- Každá operace se tedy musí uskutečnit dvakrát. Může proto snížit výkonnost systému
- minimalizujeme šance ztráty nějakých dat

Souborový systém Ext4

- ❑ Zpětná kompatibilita
- ❑ Velký souborový systém
 - Jeden soubor až 16 TiB, velikost svazku až 1 EiB (exbibyte).
- ❑ Extenty
 - Nahrazuje tradiční mapování bloků. Extent je řada sousedících fyzických bloků. Jediný extent v ext4 může mapovat až 128 MiB souvislého prostoru s velikostí bloku 4 KiB. Přímě v i-uzlu lze uložit čtyři extenty, ostatní extenty jsou indexovány ve stromu.
- ❑ Perzistentní předběžná alokace
 - ext4 umí předem přidělit místo na disku pro soubor (do souboru jsou zapsány nuly).
- ❑ Zpožděná alokace
- ❑ Neomezený počet podadresářů (dříve 32 000 podadresářů, nyní v Htree)
- ❑ Žurnál a kontrolní součet u metadat.
- ❑ Vylepšení časových razítek (dříve sekundy, nyní nanosekundy, vyřešen problém roku 2038)
- ❑ Transparentní šifrování (od roku 2015, jádro 4.1)

(Ext4, 2016)

Použitá a doporučená literatura

- ❑ DUARTE, Gustavo. Anatomy of a Program in Memory. *Many But Finite: Tech and science for curious people*. [online]. 2009, Jan 27th, 2009 [cit. 2018-02-28]. Dostupné z: <https://manybutfinite.com/post/anatomy-of-a-program-in-memory/>
- ❑ DEITEL H. M., DEITEL P. J. & CHOFFNES D. R.: *Operating systems*. 3rd ed., Pearson/Prentice Hall, 2004. ISBN 0131246968.
- ❑ TANENBAUM A. S.: *Modern operating systems*. 4th ed. Boston: Pearson, 2015. ISBN 0-13-359162-x.
- ❑ SILBERSCHATZ A., GALVIN P. B. & GAGNE G.: *Operating system concepts*. 9th ed. Hoboken, NJ: Wiley, 2013. ISBN 978-1-118-06333-0.
- ❑ STALLINGS W.: *Operating Systems: Internals and Design Principles*. 8th ed., Pearson Education Limited, 2014.

Použitá a doporučená literatura

- ❑ YOSIFOVICH, P., IONESCU, A., RUSSINOVICH, M.E., SOLOMON, D. A.: Windows Internals, Part 1: System architecture, processes, threads, memory management, and more (7th Edition). Microsoft Press, 2017.
- ❑ Yu-Hsin Hung. *Linux Kernel: Process Scheduling* [online]. Mar 25, 2016 [cit. 2019-02-07]. Dostupné z: <https://medium.com/hungys-blog/linux-kernel-process-scheduling-8ce05939fabd>
- ❑ HOFFMAN, Chris. What Is a “Zombie Process” on Linux?. *How-To Geek* [online]. September 28, 2016 [cit. 2019-02-07]. Dostupné z: <https://www.howtogeek.com/119815/htg-explains-what-is-a-zombie-process-on-linux/>

Použitá a doporučená literatura

- ❑ MICROSOFT CORPORATION. FAT32 File System Specification - General Overview of On-Disk Format. In: *Hardware White Paper* [online]. Microsoft Extensible Firmware Initiative, 2000. Dostupné z: <http://msdn.microsoft.com/en-us/windows/hardware/gg463080>
- ❑ JEFF, Hamm. *Extended FAT File System: exFAT* [online]. Paradigm Solutions, 2009, , 68 [cit. 2019-03-01].
- ❑ *NTFS.com: Data Recovery Software, File Systems, Hard Disk Internals, Disk Utilities* [online]. Active@, 2019 [cit. 2019-03-01]. Dostupné z: <https://www.ntfs.com/>
- ❑ Keshava Munegowda , Venkatraman S., Raju G. T. : The Extended FAT file system, Differentiating with FAT32 file system. Linux Conference, Prague , Czech Republic, October 26- 28, 2011.

Použitá a doporučená literatura

- ❑ Local File Systems: NTFS Technical Reference. *Microsoft: Microsoft Docs* [online]. 2009 [cit. 2019-01-20]. Dostupné z: [https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc758691\(v%3dws.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc758691(v%3dws.10))
- ❑ Šváb Jan. Souborový systém NTFS (New Technology File System). [online]. Dostupné z: https://www.download.zcu.cz/public/Prezentace/seminare_CIV_2012/KIV-PD_SvabJan_NTFS.pdf
- ❑ POIRIER, Dave. *The Second Extended File System: Internal Layout* [online]. 2011, 58 pages [cit. 2018-09-10]. Dostupné z: <http://www.freesoftware.fsf.org/ext2-doc/>
- ❑ Ext4 (and Ext2/Ext3) Wiki [online]. 2016. Dostupné z: https://ext4.wiki.kernel.org/index.php/Main_Page

Použitá a doporučená literatura

- ❑ Bell John: Operating Systems. Dostupné z:
<https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/>