

## Operační systémy 2

# Implementace souborových systémů

Petr Krajča



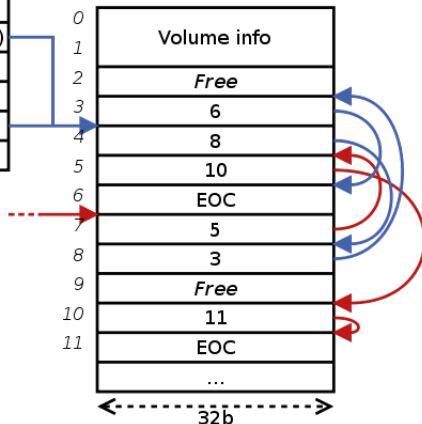
Katedra informatiky  
Univerzita Palackého v Olomouci

- souborový systém pro MSDOS (přežil se až do Windows ME, dodnes ve spotřební elektronice)
- jednoduchý design
- soubory se jmény ve tvaru 8.3, nepodporuje oprávnění
- nemá metody proti poškození dat
- disk rozdělený na bloky (clustery)
- soubory popsány pomocí File Allocation Table (FAT) – spojový seznam
- disk rozdělen na úseky:
  - bootsector (rezervovaná oblast) + informace o svazku
  - 2× FAT
  - kořenový adresář
  - data
- adresáře jako soubory; kořenový adresář je vytvořen společně s FS
- původní FAT nepodporoval adresáře

## Directory table entry (32B)

Filename (8B)
Extension (3B)
Attributes (1B)
Reserved (1B)
Create time (3B)
Create date (2B)
Last access date (2B)
First cluster # (MSB, 2B)
Last mod. time (2B)
Last mod. date (2B)
First cluster # (LSB, 2B)
File size (4B)

## File allocation table



- FAT12, 16, 32; (max. kapacity – 32 MB, 2 GB, 8 TB); záleží na velikosti clusteru
- další omezení na velikost souboru

## Virtual FAT

- podpora dlouhých jmen (LFN)
- až 256 znaků
- soubor má dvě jména – dlouhé a ve tvaru 8.3
- dlouhá jména uložena jako další záznamy v adresáři

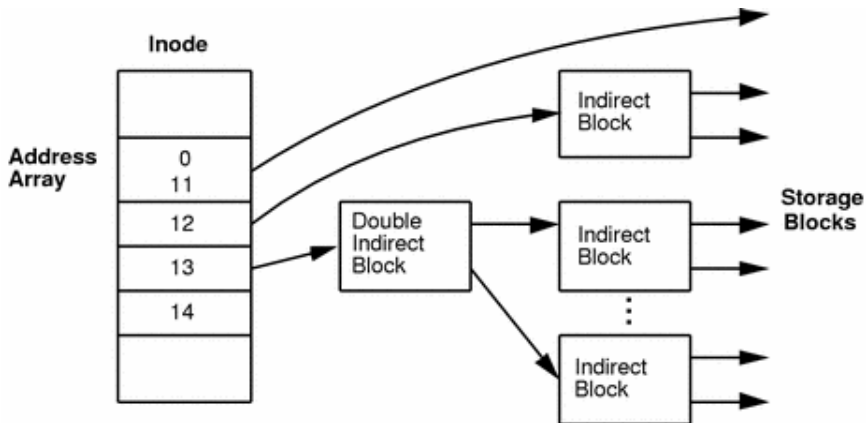
## exFAT

- určen pro flash paměti
- podpora větších disků (512 TB/64 ZB)
- podpora v novějších Windows (původně Windows CE 6)
- zatížen patenty

- v různých variantách přítomný v unixových OS – \*BSD, Solaris, System V, Linux (ext[234])
- disk se skládá:
  - **bootblock** – místo pro zavaděč OS
  - **superblock** – informace o souborovém systému
  - místo pro **inody**
  - místo pro **data**

## I-node

- **struktura popisující soubor**
- **informace o souboru**
  - **typ** souboru, **vlastníka** (UID, GID), **oprávnění** (rwx)
  - **časy** (vytvoření, přístup)
  - **počet ukazatelů**, **počet otevřených popisovačů**
- **informace o uložení dat**
  - **patnáct ukazatelů na bloky na disku**
  - bloky **0 až 11 ukazují na bloky dat**
  - blok 12 – **nepřímý** blok **1. úrovně**
  - blok 13 – **nepřímý** blok **2. úrovně**
  - blok 14 – **nepřímý** blok **3. úrovně**



- adresář je soubor obsahující sekvenci dvojic (jméno souboru, číslo inode)
- struktura inode umožňuje mít *řídke soubory*
- velikost bloku  $\Rightarrow$  rychlejší přístup k větším souborům vs. nevyužité místo
- možnost rozdělit blok na několik fragmentů
- k evidenci volného místa a inode se používají bitmapy
- svazek může být rozdělený na několik tzv. skupin – každá mající vlastní inody, bitmapy, atd. + kopie superbloku  $\Rightarrow$  sloučení souvisejících dat  $\Rightarrow$  eliminace přesunů hlavičky
- konkrétní detaily se mohou lišit
- např. FreeBSD přidává možnost dělat snapshoty

**Directory inode (128B)**

Type	Mode
User ID	Group ID
File size	# blocks
# links	Flags
Timestamps (x3)	
Direct blocks (x12)	
Single indirect	
Double indirect	
Triple indirect	

**Directory block**

.	inode #
..	inode #
passwd	inode #
fstab	inode #
...	...

**Indirect block**

Direct blocks (x512)

**File inode (128B)**

Type	Mode
User ID	Group ID
File size	# blocks
# links	Flags
Timestamps (x3)	
Direct blocks (x12)	
Single indirect	
Double indirect	
Triple indirect	

**File data block**

Data

Block # of  
block with  
512 double  
indirect  
entries

Block # of  
block with  
512 single  
indirect  
entries

Block #s of  
more  
directory  
blocks



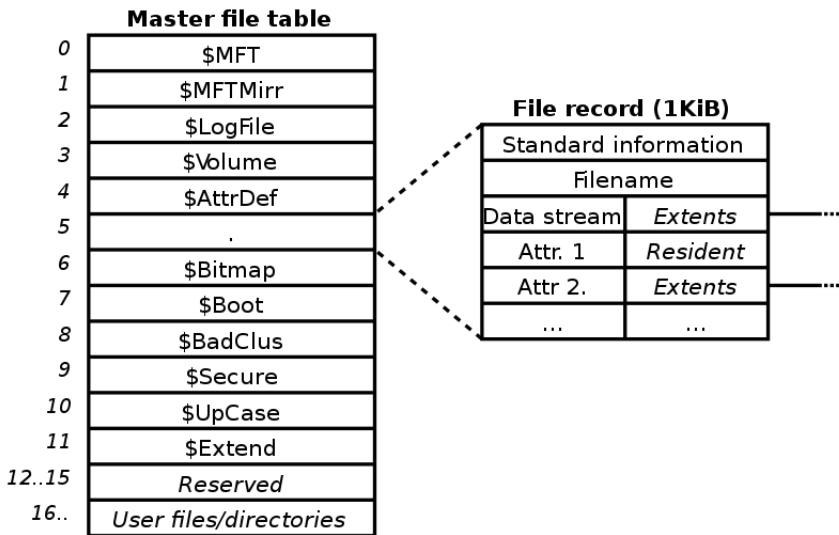
- Linux nemá jeden hlavní FS
- nejčastěji se používá: ext4 (nejkonzervativnější volba)
- název souboru může mít až 256 znaků (s výjimkou znaků / a \0)
- vychází z UFS
- ext2: maximální velikost souboru 16 GB–2 TB, disku: 2 TB – 16 TB
- ext3: přidává žurnál (3 úrovně – journal, ordered, unordered), binárně kompatibilní s ext2
- ext4: přidává vylepšení
  - maximální velikost souboru 16 TB, disku: 1 EB
  - podpora extentů (místo mapování jednotlivých bloků je možné alokovat i souvislou oblast až do velikosti 128 MB)
  - optimalizace alokací
  - lepší práce s časem
- další FS: BtrFS, JFS, XFS, ...

- hlavní souborový systém Windows NT
- kořeny v OS/2 a jeho HPFS (vyvíjen od roku 1993)
- velikost clusteru podle velikosti svazku (512 B–4 KB)  $\Rightarrow$  max. velikost disku 256 TB  
max. velikost souboru 16 TB
- oproti FAT (souborovému systému W9x) ochrana před poškozením + práva
- žurnálování a transakce
- podpora více streamů v jednom souboru
- dlouhé názvy (255 znaků) + unicode
- podpora standardu POSIX; hardlinky, symlinky
- komprese a řídké soubory

## Adresáře

- opět technicky soubory; jména v B+ stromech
- některá metadata souborů jsou součástí adresáře

- na začátku disku: boot sector
- 12 % MFT (Master File Table); 88 % data souborů
- MFT je soubor popisující všechny soubory na FS (MFT je taky soubor)
- MFT se skládá ze záznamů o velikosti 1 KB
- každý soubor je popsán tímto záznamem
- 32 prvních souborů má speciální určení (\$MFT, \$MFTMirr, \$LogFile, \$Volume, \$Bitmap, \$Boot, \$BadClus, ...)
- informace o souborech včetně jména, časů, atd. uloženy jako záznam v MFT jako dvojice *atribut-hodnota*
- tělo souboru je taky atribut  $\implies$  uniformní přístup; možnost uložit malé soubory přímo do MFT
- alternativní proudy  $\implies$  opět atributy
- v případě potřeby může jeden soubor zabrat víc záznamů v MFT
- případně lze použít místo mimo MFT (rezidentní a nerezidentní atributy)



- data v souboru jsou popsána pomocí (atributu) tabulky mapující VCN (virtual cluster number) na LCN (logical cluster number)
- VCN – číslo clusteru v souboru (indexováno od nuly)
- LCN – číslo clusteru ve svazku
- každý záznam v tabulce je ve tvaru: VCN, LCN, počet clusterů, např.

VCN	LCN	počet
0	123	4
4	42	8
32	456	15

## Komprese

- řídké soubory
- možnost transparentně komprimovat obsah (vždy po 16 clusterech)  $\implies$  bloky dat zarovnány na 16 clusterů; pokud zabírá méně místa, je komprimován
- čtení i zápis provádí (de)kompresi (LZ77)  $\implies$  dopad na výkon

## ■ vyvážené stromy

- všechny listy jsou ve stejné hloubce
- každý uzel (mimo kořenového) obsahuje nejméně  $t - 1$  klíčů, tj. má  $t$  potomků; v neprázdném stromě kořen obsahuje alespoň jeden klíč
- každý uzel má nanejvýš  $2t - 1$  klíčů  $\implies 2t$  potomků  $\implies$  plný uzel
- **Věta:** Pokud je počet klíčů  $n \geq 1$ , pak pro B-strom stupně  $t \geq 2$  a výšky  $h$  platí

$$h \leq \log_t \frac{n + 1}{2}.$$

- ne všechna data v paměti (vs. běžné binární stromy)
- rozdílné přístupové doby primární paměti a sekundární (desítky až stovky nanosekund vs. milisekundy–vystavení hlavičky)
- preferované sekvenční čtení  $\implies$  načtení celých stránek
- zobecnění 2,3-stromů, 2,3,4-stromů
- rozlišujeme složitost operací (porovnání, atd.) a I/O operací (zápisy, čtení)
- složitost vyhledávání, vložení:  $O(th) = O(t \log_t n)$
- počet přístupů na disk:  $O(h) = O(\log_t n)$

## XFS

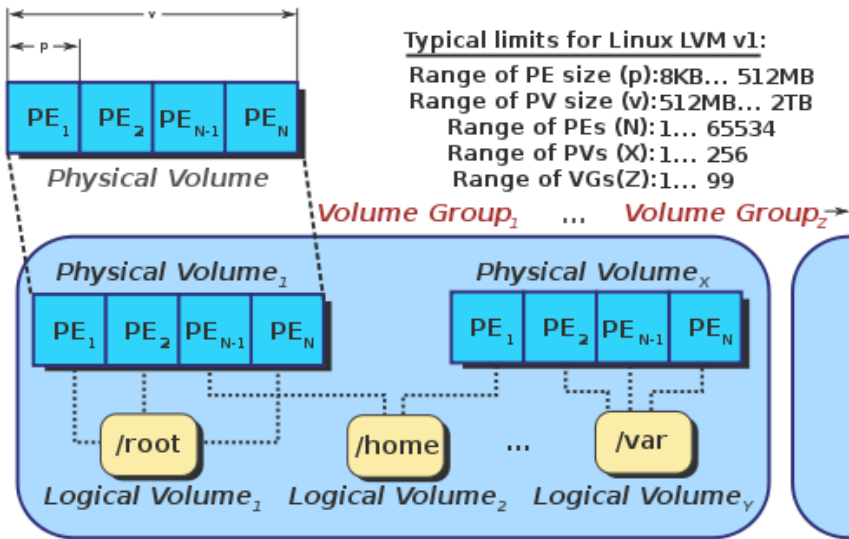
- navržen SGI pro operační systém IRIX (k dispozici i v Linuxu)
- spoléhá na B+stromy (nutné zaplnění ze 2/3)
- rozdělení disku na agregační jednotky
- evidence volného místa v B-stromech (dva stromy  $\implies$  vyhledávání podle pozice, velikosti)
- uložení souborů  $\implies$  extenty jako v případě NTFS (uloženo v inodách)
- u větších souborů použití B-stromů  $\implies$  zřetězení listů
- malé adresáře v inodách; větší  $\implies$  B-stromy

## JFS

- navržen IBM pro AIX
- koncepce žurnálování blízká databázovým systémům
- v některých ohledech podobný přístup jako u XFS

- problém: **disky a oddíly mají pevnou velikost** (rozdělení disku je pevně dané)
- řešení: **logical volume management**—**vrstva mezi blokovým zařízením a FS**
- fyzické disky (PV: **physical volumes**) **rozdělen na rozsahy** (PE: **physical extents**)
- **jednotlivé PE poskytnuty do společné Volume Group**
- odtud jsou pak **přidělovány jednotlivým logickým svazkům**  $\implies$  možnost **dynamicky měnit velikost svazku**  $\implies$  **nutná podpora FS**
- možnost **emulovat RAID**
- možnost **vložit vrstvu**, která se bude starat o **snapshoty/klony** (CoW)
- možnost **transparentně provádět šifrování**
- ve **Windows** implementace podobná: **Logical Disk Manager** & **Volume Snapshot Service** (umožňují SW RAID); spolupráce s FS
- někdy **dodáván jako software třetích stran**

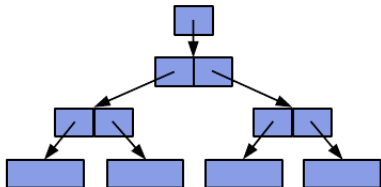




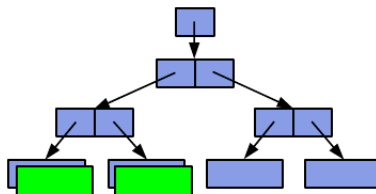
- moderní souborový systém (r. 2005); SUN (Oracle)
- podpora (open) Solaris, FreeBSD, NetBSD, OS X?, Linux (licenční problémy)
- kombinuje prvky LVM, RAID
- interně 128 bitová adresace (max. kapacita 256 ZB, ostatní limity kolem 16 EB)
- disky jsou spojeny do *poolu*, FS dělá automatický stripping  $\Rightarrow$  rozprostře se přes všechny disky
- bloky dat různých velikostí
- little- a big-endian (podle aktuální situace)
- ditto blocks (zdvojené zápisy)
- deduplikace
- podpora komprese

- RAID-Z: podobný RAID-5, ale má různě velké bloky (odpovídají logickým blokům)  $\Rightarrow$  např. 3 bloky dat + 1 paritní, atd.
- u dat jsou evidovány kontrolní součty  $\Rightarrow$  ochrana proti tichému poškození (chyba HW i SW)
- konzistence založena na metodě Copy-on-Write
- používaná data nikdy nejsou přepsána  $\Rightarrow$  nejdřív jsou zapsána data a pak jsou (atomicky) změněna metadata
- $\Rightarrow$  výhodné slučovat operace do transakcí
- $\Rightarrow$  FS je vždy v konzistentním stavu
- $\Rightarrow$  infrastruktura pro vytváření snapshotů/klonů souborového systému

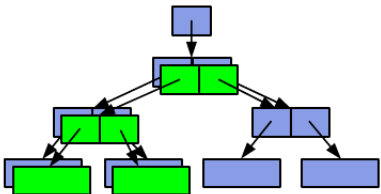
## 1. Initial block tree



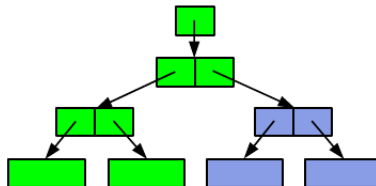
## 2. COW some blocks

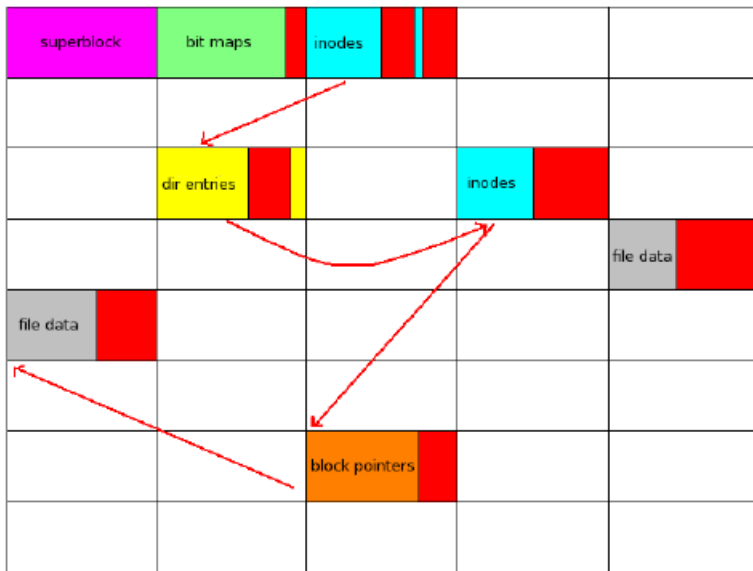


## 3. COW indirect blocks



## 4. Rewrite uberblock (atomic)

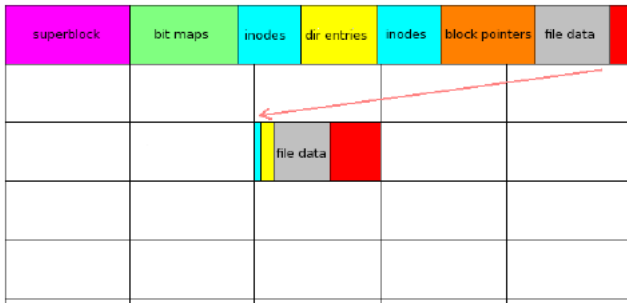
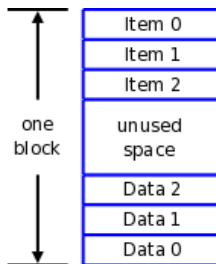




- všechna data uložena v B-stromech
- varianta podporující CoW (řeší integritu)
- jedna implementace (jednoduchá implementace, kontrolní součty, atd.)
- všechny klíče ve tvaru:

```
struct btrfs_disk_key {  
    __le64 objectid;  
    u8 type;  
    __le64 offset;  
}
```

- slučování souvisejících dat vedle sebe
- malé soubory ve stromu
- velké soubory (vlastní extenty), popsané v klíči (využití offsetu)
- automatická defragmentace
- několik speciálních stromů (volné místo)



- souborový systém pro CD-ROM; podpora všech OS
- zápis jen jednou; sekvenční čtení  $\Rightarrow$  není potřeba dělat kompromisy
- logický sektor 2048 B (může být i větší)
- na disku může být víc logických svazků; svazek může být na více discích
- na začátku 16 rezervovaných bloků + 1 blok (Primary Volume Descriptor)  $\Rightarrow$  informace o disku; odkaz na kořenový adresář
- adresář popsán pomocí záznamů proměnlivé délky (viz Tan. 432)
  - textová data v ASCII
  - binární 2 $\times$  (little- i big-endian)
- možnosti formátu určeny úrovněmi a rozšířeními
- **Level 1** – soubory 8.3; všechny soubory spojitě; 8 úrovní adresářů
- **Level 2** – jména až 31 znaků
- **Level 3** – nespojitě soubory (jednotlivé souvislé bloky se mohou opakovat)
- *Rock Ridge* – kompatibilita s unixy (jména, oprávnění, odkazy, speciální soubory)
- *Joliet* – kompatibilita s Windows



- náhrada za ISO-9660
- používán převážně pro DVD a Blue-ray disky
- dlouhé názvy, soubory až 1EB
- různé varianty formátu:
  - **Plain build** – základní formát (data lze přepisovat, pokud to médium podporuje; přepis konkrétních sektorů – DVD-RAM, DVD+RW)
  - **Virtual Allocation Table** – inkrementální zápisy (CD-R)
  - **Spare build** – pokud to médium podporuje, lze data přepisovat; zahrnuta obrana proti opotřebení sektorů