Operační systémy 2

Implementace souborových systémů

Petr Krajča

Katedra informatiky
Univerzita Palackého v Olomouci

8. listopad, 2010

Implementace souborových systémů: očekávané vlasnosti

- (budeme předpokládat souborové systémy pro práci s disky)
- schopnost pracovat se soubory a disky adekvátní velikosti
- efektivní práce s místem (evidence volného, nízká fragmentace)
- rychlý přístup k datům
- eliminace roztroušení dat na disku
- odolnost proti poškození při pádu systému (výpadku napájení) $\Longrightarrow$ rychlé zotavení

Další vlastnosti

- snapshoty
- komprese dat
- možnost zvětšovat/zmenšovat FS za běhu
- kontrolní součty
- defragmentace za běhu
- správa oprávnění
- atd.

Struktura disku

- pro jednoduchost předpokládáme, že struktura disku je lineární
- MBR – master boot record: informace o rozdělení disku na svazky + zavaděč
- Tan. 400
- *sektor disku* – obvykle velikost 512 B
- $\Longrightarrow$ pracuje se s většími bloky 1-32 kB (často 4 kB)
- $\Longrightarrow$ optimální velikost bloku? (rychlost vs. úspora místa)
- jednotlivé svazky obsahují souborový systém (vlastní organizace dat)
- VFS – virtualní souborový systém (Sta. 567)
- je potřeba si udržovat informace o jednotlivých souborech (FCB, inody)
- cache

Alokace diskového prostoru

Alokace souvislých bloků

- soubory jsou ukládany na disk za sebe v souvislých blocích
- rychlé sekvenční čtení, problém s uvolněnými soubory
- CD?

Soubor jako spojový seznam

- FS je rozdělený na bloky
- každý blok ma ukazatel na následující
- rychlé sekvenční čtení; problematický náhodný přístup (nutnost číst vše) + poškození disku
- varianta: uchovávání odkazů na bloky ve spec. tabulce (FAT)

Indexová alokace

- soubor si nese informaci o svém uložení v blocích (struktura na začátku souboru)
- problém s velkými soubory $\Longrightarrow$ víceúrovňové tabulky

Evidence volného místa

- je potřeba udržovat informace o volném místě
- použití spojového seznamu volných bloků (možné použít volné bloky); jako u souborů
- vylepšení $\Longrightarrow$ rozsahy volných bloků (problém při fragmentaci)
- bitmapy – každý bit udává, jestli je daný blok volný (nutné místo – obvykle méně než promile kapacity)

Přidělování volného místa

- je žádoucí zapisovat data do souvislých volných bloků $\Longrightarrow$ eliminace přesunů hlavičky
- heuristické algoritmy (složité na testování); problematické zaplnění disku na 95 % + současné zapisování více velkých souborů
- algoritmus v ext2
  - zvolí se cíl zápisu (první volný blok, přip. první volný blok za souborem)
  - hledá se v bitmapě první volných 8 bitů (8 bloků), ležících za cílem; pokud takové nejsou, hledá se po bitech
  - (prealokace) alokátor se snaží zabrat 8 bloků za sebou
- clusterování – souvislý zápis více bloků (možnost je přeskládat); FreeBSD

Adresáře

- organizace adresářů $\Longrightarrow$ vliv na výkon
- různé struktury
  - spojový seznam – jednoduchá implementace; větší složitost
  - hash tabulka (komplikace s implementací)
  - varianty B-stromů (časté u moderních FS)
- umístění informací o souborech
  - součást adresáře
  - součást souboru (UNIX) $\Longrightarrow$ problém: listování adresáře; možnost mít soubor ve více nebo žádném adresáři

Cache a selhání systému

- kvůli rychlejšímu přístupu nejsou často data zapisována přímo na disk $\Longrightarrow$ nejdřív do cache
- při výpadku (pád systému, výpadek napájení) nemusí být data ve write-back cache zapsána $\Longrightarrow$ poškození FS
- potřeba opravit FS (`fsck`, `chkdsk`) $\Longrightarrow$ časově náročné
- případné narušení systému
  - jeden uživatel zapíše data na disk a smaže je
  - druhý uživatel vytvoří velký soubor a po zapsaní metadat vyvolá výpadek
  - po restartu čte data prvního uživatele

Řešení

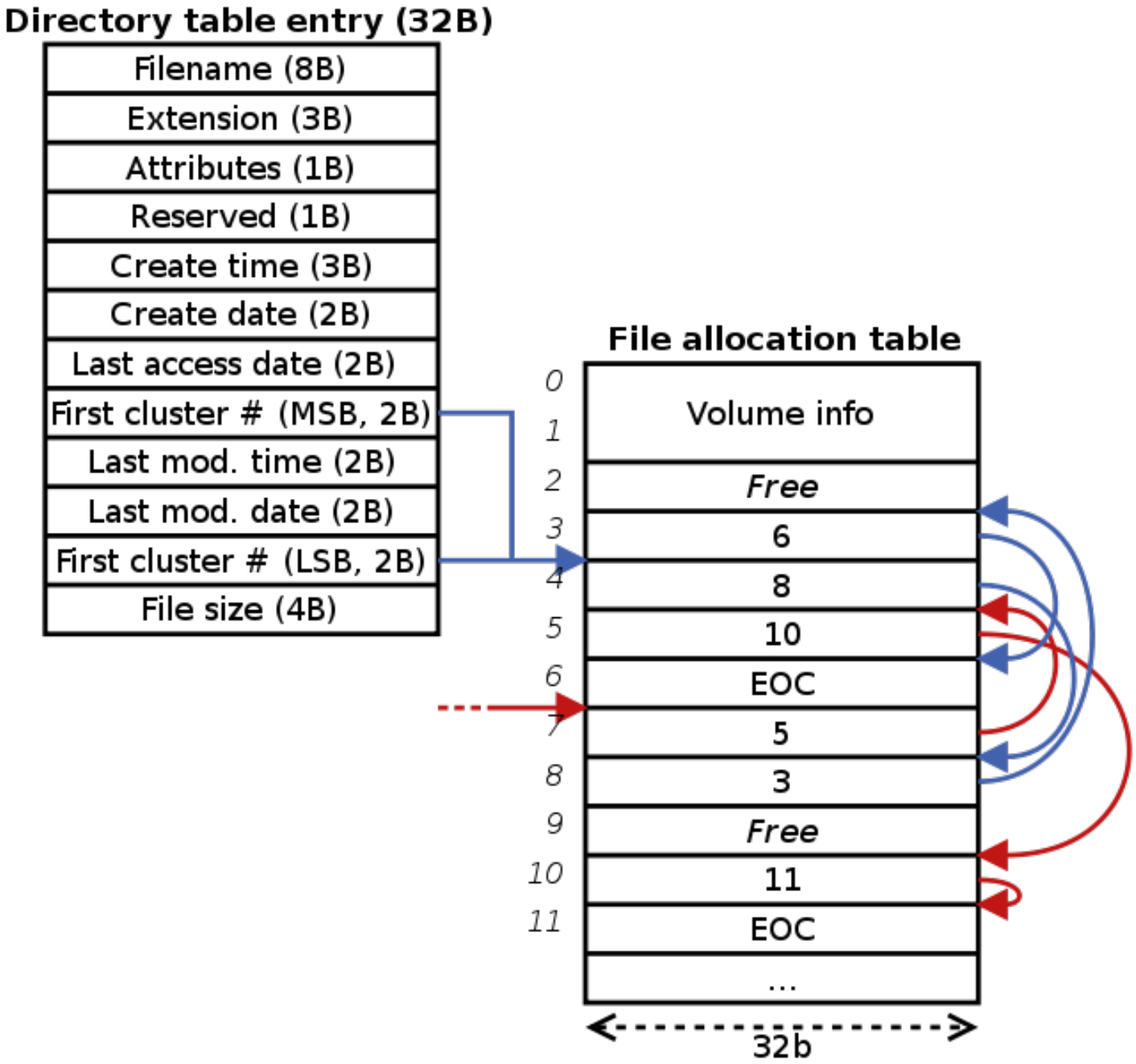
- synchronní zápis $\Longrightarrow$ zpomalení, konzistence nemusí být zaručena
- soft updates – uspořádání zápisů podle určitých pravidel (*BSD)
- žurnálování

Žurnálování

- data se zapisují v transakcích (přesun FS z jednoho konzistentního stavu do druhého)
- nejdřív se transakce zapíše do žurnálu (logu)
- po zapsání do žurnálu je záznam označen spec. značkou a data se můžou zapsat na disk
- po zapsání na disk je zápis z žurnálu odstraněn
- při připojení FS se kontroluje stav žurnálu
  - zápis záznamu do žurnálu nebyl dokončen (transakce se neprovede)
  - případně, transakse se provede podle informací ze žurnálu
- často se žurnálují jen metadata
- žurnál je cyklický; při zaplnění se zapíší/uvolní ty na začátku
- je potřeba atomických zápisů na disk
- cache & buffery komplikují implementaci

FAT

- souborový systém pro MSDOS (přežil se až do Windows ME)
- 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 svazek je vytvořen hned na začátku
- 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
0
1
2
3
4
5
6
7
8
9
10
11
Volume info
Free
6
8
10
EOC
5
3
Free
11
EOC
...
32b

FAT: varianty

- FAT12, 16, 32: podle velikosti clusteru; (max. kapacity – 32 MB, 2 GB, 8 TB)
- 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 (povedně Windows CE 6)
- zatížen patenty

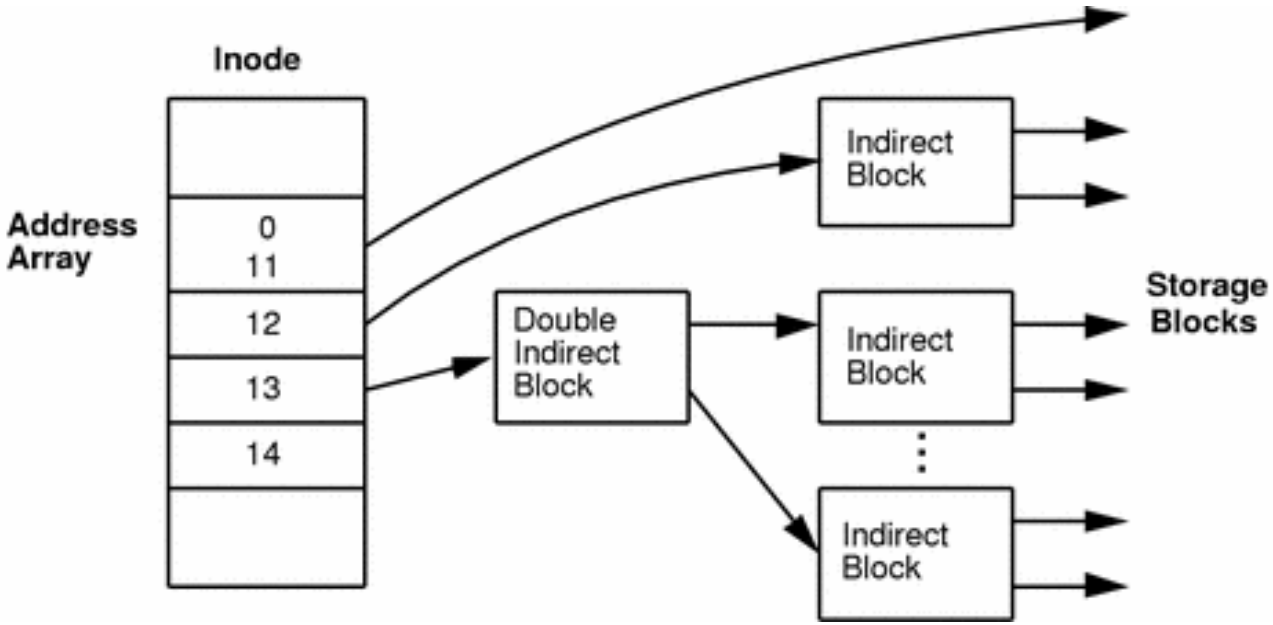
UFS: Unix File System (1/2)

- 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

Inoda

- 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 0-11 ukazují na bloky dat
  - blok 12 – nepřímý blok 1. úrovně
  - blok 13 – nepřímý blok 2. úrovně
  - blok 14 – nepřímý blok 3. úrovně

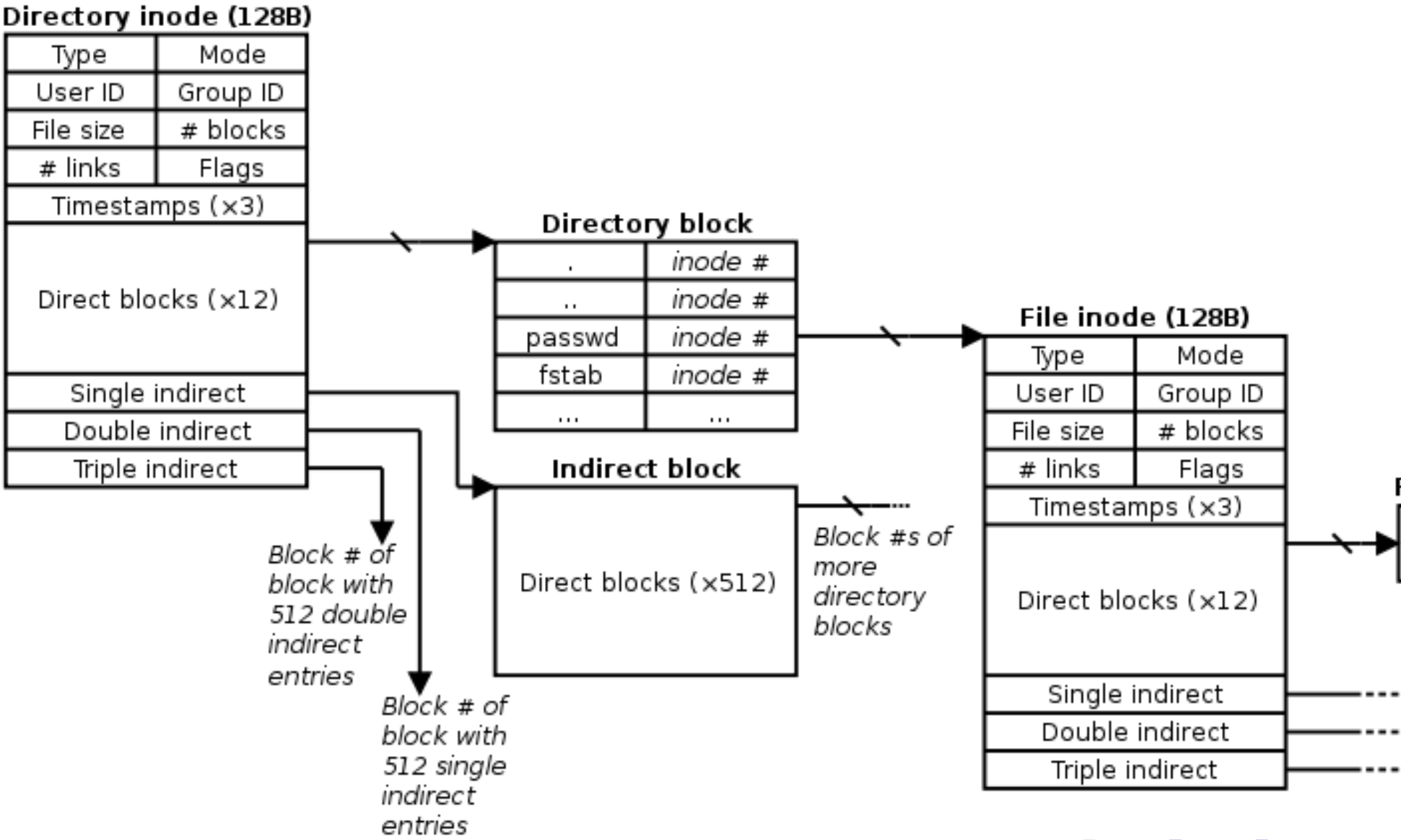
Inode


Inode
Address Array
0
11
12
13
14
Double Indirect Block
Indirect Block
Indirect Block
Indirect Block
Storage Blocks

UFS: Unix File System (2/2)

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

Inode


Directory inode (128B)
Type
Mode
User ID
Group ID
File size
blocks
links
Flags
Timestamps (×3)
Direct blocks (×12)
Single indirect
Double indirect
Triple indirect
Directory block
.
inode #
..
inode #
passwd
inode #
fstab
inode #
...
...
Indirect block
Direct blocks (×512)
Block #s of more directory blocks
Block # of block with 512 double indirect entries
Block # of block with 512 single indirect entries
File inode (128B)
Type
Mode
User ID
Group ID
File size
blocks
links
Flags
Timestamps (×3)
Direct blocks (×12)
Single indirect
Double indirect
Triple indirect

FS v Linuxu

- Linux nemá jeden hlavní FS
- nejčastěji se používá: Ext2/3/4
- název souboru může mít až 256 znaků (s vyjímkou 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–2 TB, disku: 1 EB
  - podpora extentů (místo mapování bloků je možné alokovat i blok až do velikosti 128 MB)
  - optimalizace alokací
  - lepší práce s časem
- další FS: ReiserFS, BtrFS, JFS, XFS, ...