OS – Správa souborů

Tomáš Hudec

Tomas.Hudec@upce.cz

http://fei-as.upceucebny.cz/usr/hudec/vyuka/os/

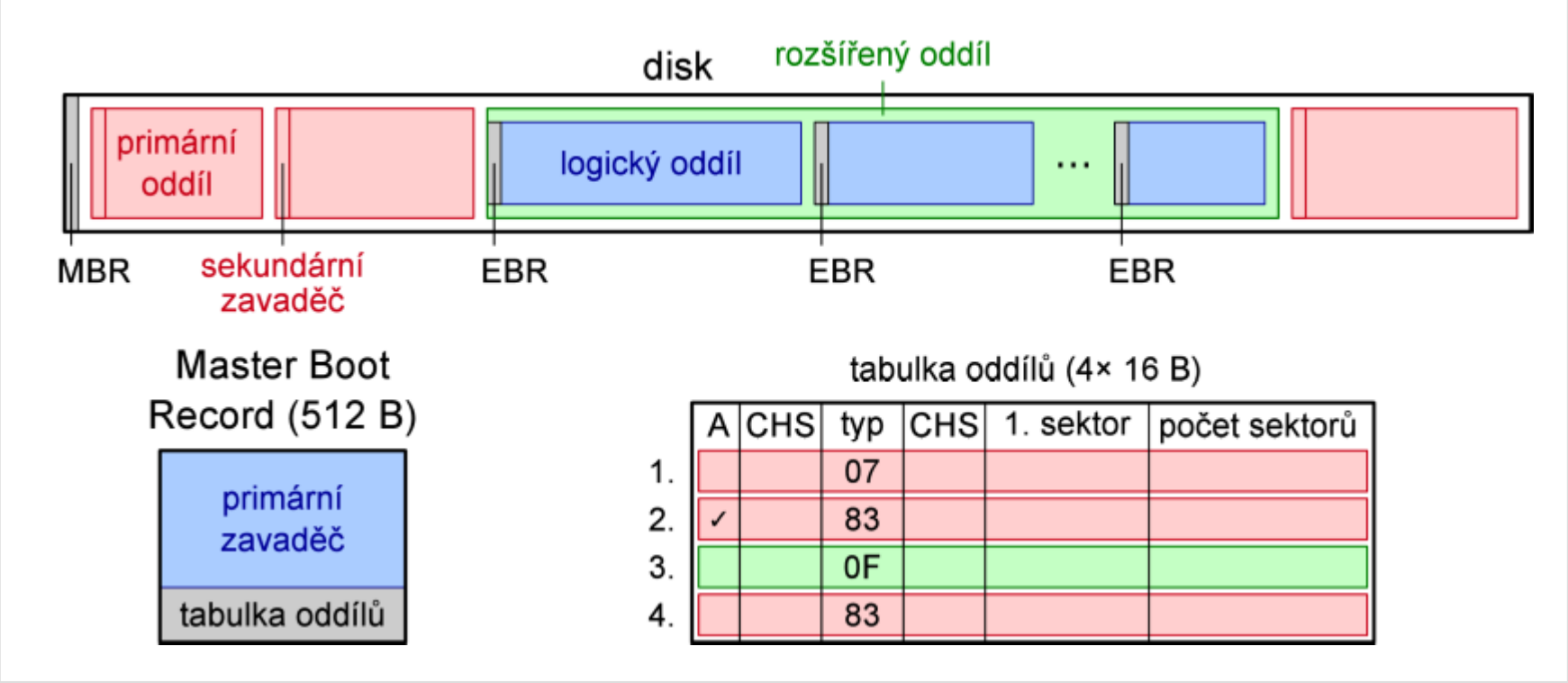
Úložiště, důvody dělení na oddíly

- úložiště (storage)
  - magnetický (mechanický) disk, SSD (Solid State)
  - lze dělit na oddíly (partitions)
    - jiný souborový systém (např. pro jiný OS)
    - snadnější správa – oddělení systému od dat
      - např. adresáře uživatelů nemohou zaplnit systémový disk
    - možnost snadnější obnovy části dat při přepsání či při HW chybě – chyba se může týkat jen některého oddílu
    - oddíl pro swap – na rozdíl od souboru oddíl netrpí datovou fragmentací – rychlejší přístup

Diskový oddíl

- MBR (Master Boot Record) – první sektor disku
  - tabulka rozdělení disku na oddíly (čtyři záznamy)
    - primární – lze z něj zavést OS sekundárním zavaděčem
    - rozšířený – dělí se na logické disky
  - zavaděč systému, může být univerzální – skok na sekundární zavaděč na aktivním primárním oddíle
- GPT (GUID Partition Table) – standard UEFI
  - za „protective MBR“ + záloha na konci disku
  - až 128 (primárních) oddílů (1 záznam má 128 B)
  - odstraňuje limity MBR, nutné pro disky nad 2 TiB

Diskové oddíly MBR (obrázek)



EBR (Extended BR) má v tabulce oddílů dva záznamy: následující logický oddíl + odkaz na další EBR

Soubor

- soubor (file)
  - univerzální forma dlouhodobého uložení dat v sekundární paměti (na disku)
  - vstupní data pro programy
  - uložení výstupních dat
  - uložení programů
  - distribuce, archivace programů a dat
  - sdílení dat

Adresář

- adresář, složka (directory, folder)
  - umožňuje hierarchické uspořádání souborů
  - obvykle je reprezentován speciálním souborem, jehož obsah je interpretován systémem
  - obsahuje informace o souborech
- adresáře tvoří hierarchickou strukturu – strom
  - hlavní adresář se označuje jako kořenový (root)
  - adresář aktuální – pracovní (current, working)
  - cesta (path) určuje logické umístění souboru
    - absolutní (od kořene), relativní (z aktuálního adresáře)

Systém správy souborů

- souborový systém (file system, FS)
  - umožňuje uživatelům a procesům přístup k souborům (podle jména) na paměťovém médiu
  - vytváří logickou (adresářovou) strukturu souborů
  - eviduje metadata souborů (atributy, oprávnění)
  - organizuje uspořádání souborů na médiu
    - programátoři nemusejí vyvíjet vlastní SW prostředky pro manipulaci s daty na médiu, alokaci místa apod.
  - sjednocuje přístup k datům různého typu a původu
    - vstup a výstup na různé periferie se neliší

Příklady souborových systémů

- unixové systémy
  - BSD: UFS, Solaris: UFS, ZFS, IRIX: XFS, AIX: JFS
  - Linux: ext2/3/4, ReiserFS, BtrFS, NILFS, F2FS
    - podporuje mnoho dalších unixových i neunixových
- OS X [ou es ten], iOS, Mac OS
  - HFS, HFS Plus, UFS (od OS X; z NeXTSTEP, BSD)
- Windows
  - FAT, NTFS, exFAT, ReFS (Resilient FS, od Win8)
- síťové – NFS, SMB (CIFS)

Úložiště

- DAS (Directly Attached Storage)
  - lokální paměťové úložiště (blokové zařízení)
  - SCSI, SATA, IDE
- NAS (Network Attached Storage)
  - souborový systém zpřístupněný síťovým protokolem
  - NFS, CIFS (SMB, Samba), AFS
- SAN (Storage Area Network)
  - blokové zařízení zpřístupněné síťovým protokolem
  - iSCSI, Fibre Channel

RAID

- Redundant Array of Independent Disks
  - zapojení více disků do jednoho virtuálního zařízení
    - HW: závislé na řadiči, SW: výpočet parity v CPU
  - výhodou je rychlost a/nebo redundance
  - RAID-0 – stripping: prokládané ukládání
    - rychlejší, ale výpadek disku znamená ztrátu všeho
  - RAID-1 – mirroring: zrcadlené ukládání
    - rychlejší, redundantní, kapacita jen jednoho disku
  - RAID-5 – stripping + parity: prokládané s paritou
    - rychlejší, redundantní, kapacita snížená o jeden disk

Metadata

- metadata (data o datech)
  - unixové systémy – superblok, i-uzel (i-node)
  - NTFS – boot sektor, Master File Table (MFT)
  - FAT – boot sektor, adresář a alokační tabulka
- metadata souborového systému (FS)
  - velikost FS, vlastnosti FS, seznam volných bloků, odkaz na evidenci metadat souborů
- metadata souborů – atributy
  - typ, velikost, čas změny, oprávnění, umístění, …

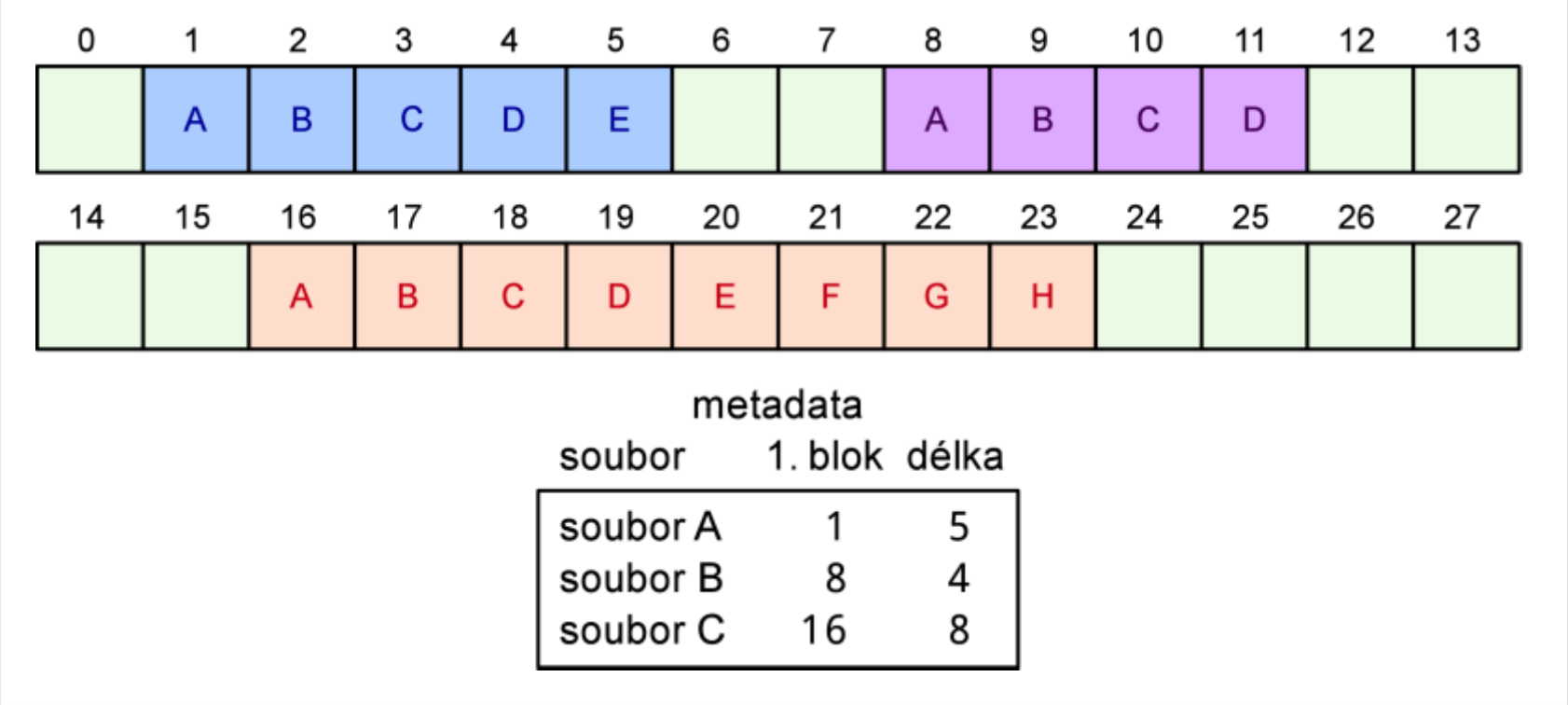
Alokace prostoru na médiu

- soubory alokují prostor po alokačních blocích
  - alokační blok (cluster) má velikost několik sektorů
    - obvykle mocnina dvou, tj. 1, 2, 4, 8, 16, …
  - sektor je nejmenší alokovatelná jednotka na médiu
    - typicky 512 bajtů, 2 KiB (CD) nebo 4 KiB (disky od r. 2010)
  - blok nemusí být využit zcela – vnitřní fragmentace
- metody alokace
  - souvislá (contiguous allocation)
  - řetězená (chained allocation)
  - indexová (indexed allocation)

Souvislá alokace

- souboru jsou přiděleny po sobě jdoucí bloky
- v metadatech souboru se eviduje:
  - adresa prvního (počátečního) bloku
  - velikost souboru (v alokačních blocích)
- při alokaci prostoru na médiu dochází k vnější fragmentaci
  - vznikají díry, které je obtížné využít
  - soubor nemůže (bez přemístění) růst nad limit volného prostoru za posledním blokem

Souvislá alokace (obrázek)



metadata

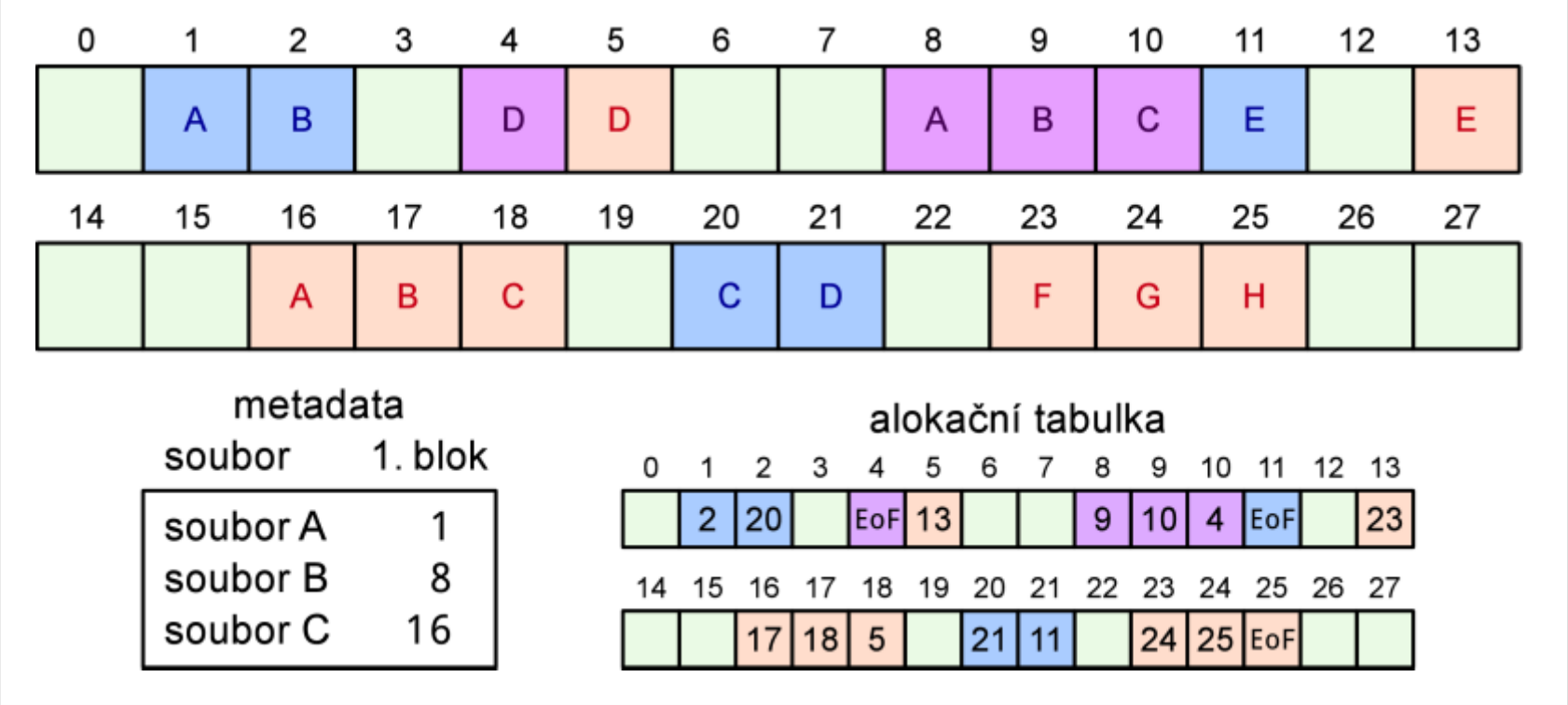
soubor	1. blok	délka
soubor A	1	5
soubor B	8	4
soubor C	16	8

metadata obsahují odkaz na 1. blok a délku

Řetězená alokace

- alokují se jednotlivé bloky
- alokační tabulka obsahuje zřetězený seznam
  - každý blok obsahuje odkaz na následující blok
- odstraněna vnější fragmentace
  - souboru lze přidělit libovolný další blok
- logicky sousedící bloky mohou být na různých místech na médiu – datová fragmentace
- př.: souborový systém FAT (DOS / Windows)

Řetězená alokace (obrázek)



metadata obsahují odkaz na 1. blok,

alokační tabulka obsahuje pro každý blok odkaz na další

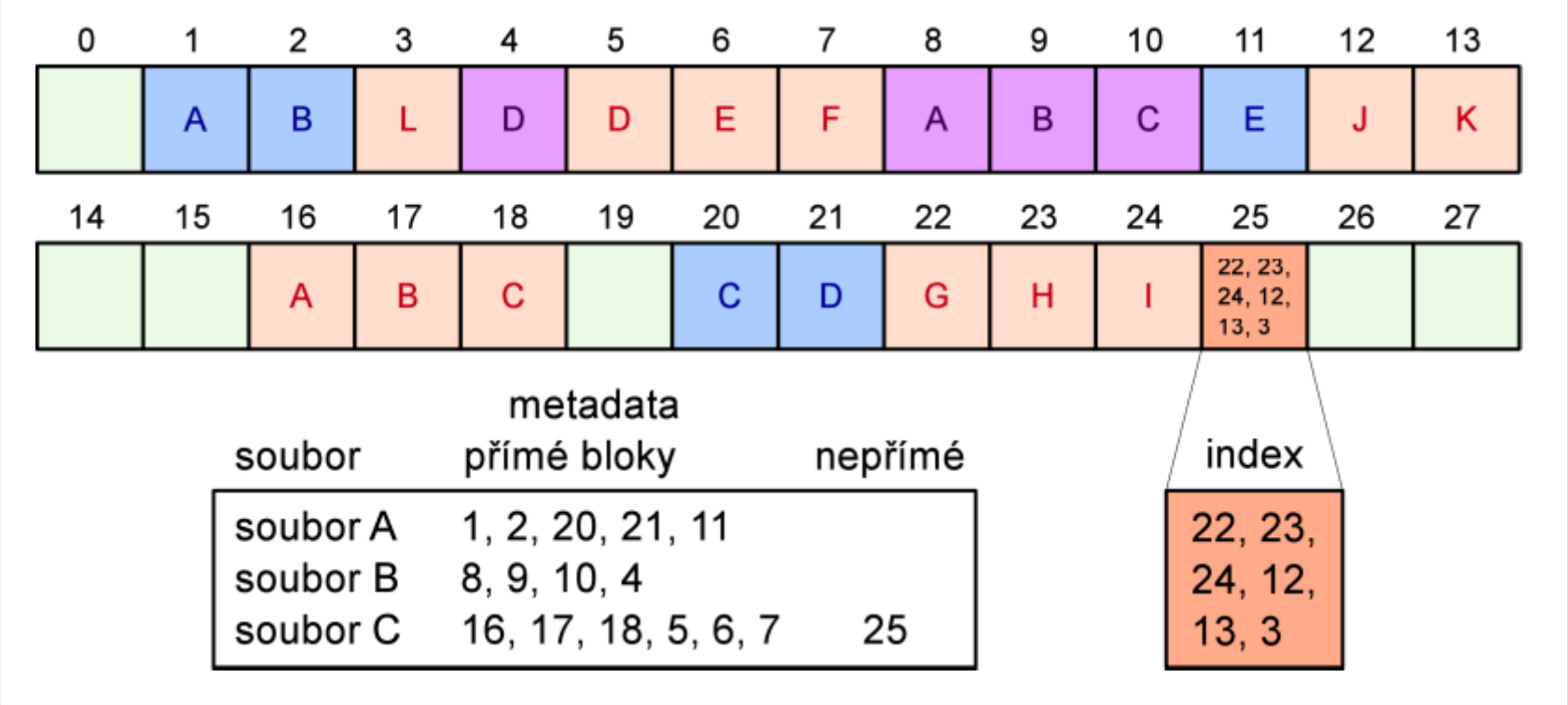
File Allocation Table (FAT)

- FAT12, FAT16, FAT32
  - číslo udává rozsah – počet bitů na položku
  - maximální počet alkokačních bloků (clusterů)
  - maximální velikost FS pro danou velikost clusteru
    - FAT16 a cluster osm 512B sektorů: $2^{16} \cdot 4$ KiB = 256 MiB
    - FAT12 a cluster čtyři 512B sektory: $2^{12} \cdot 2$ KiB = 8 MiB
  - FAT32 pro číslo clusteru užívá jen 28 bitů ze 32
    - max. počet clusterů je tím prakticky omezen na 2^{28}
  - velikost FAT je daná počtem clusterů na FS
    - velikost FS / velikost clusteru · velikost položky FAT

Indexová alokace

- index obsahuje seznam přidělených bloků
  - odstraněna vnější fragmentace
- index se může odkazovat na blok s indexem
  - nepřímý index (víceúrovňový)
- bloky mohou mít i různou velikost
  - redukce vnitřní fragmentace
- př.: unixové souborové systémy
  - i-uzel obsahuje několik přímých odkazů na bloky a (nepřímé) odkazy na další indexy bloků

Indexová alokace (obrázek)

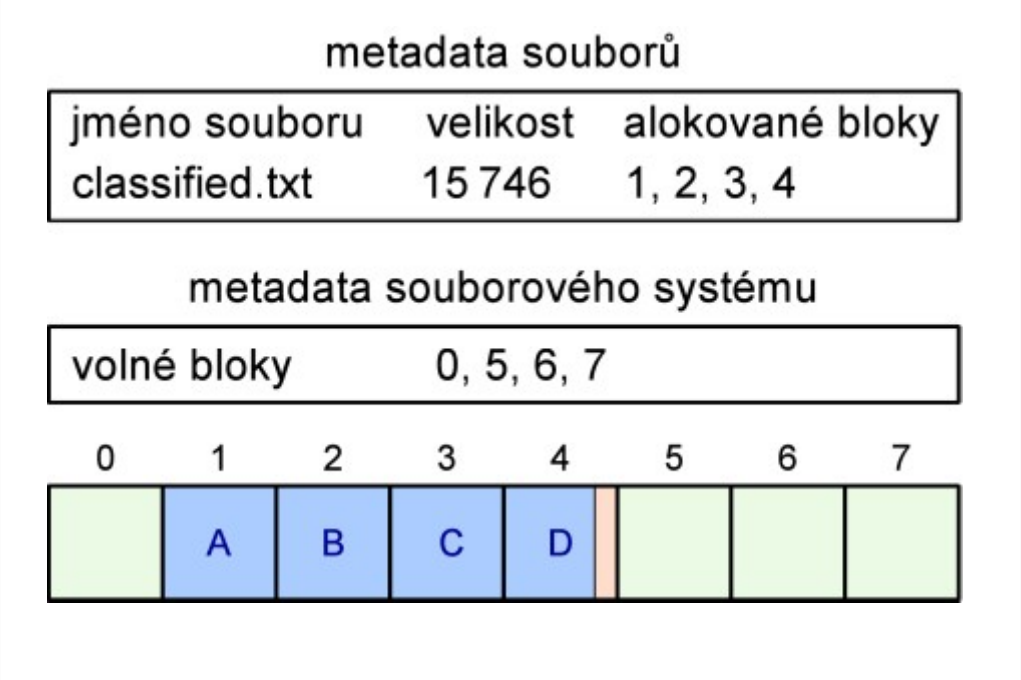


metadata obsahují index s odkazy na bloky,
některé odkazy mohou být nepřímé – na blok s indexem

Konzistence souborového systému

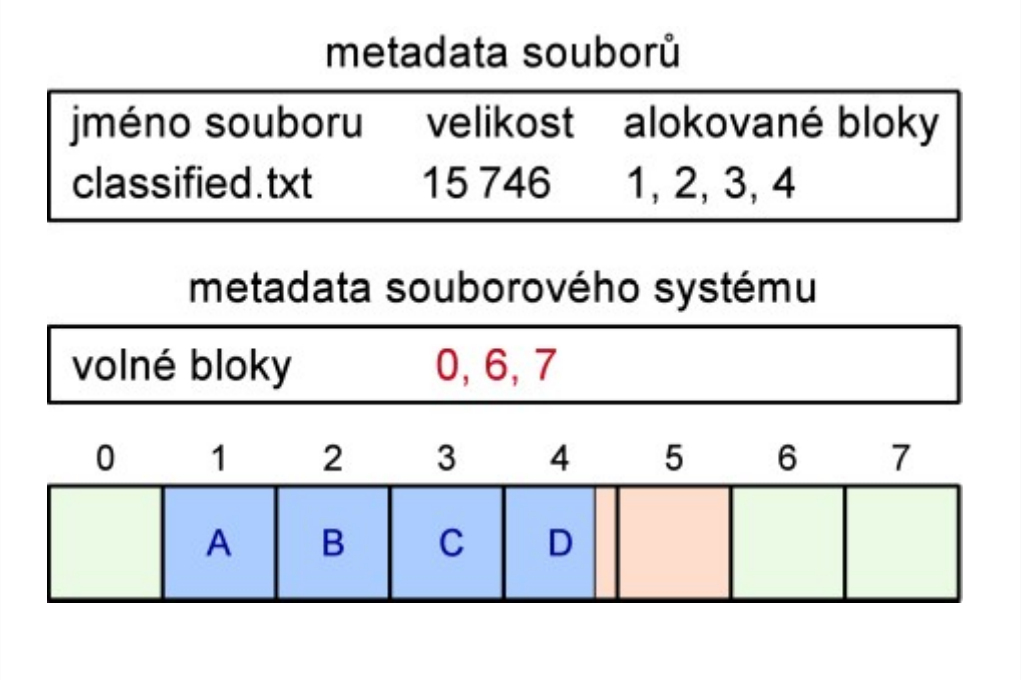
- zápis dat do souboru znamená provedení operací na různých místech na médiu
  - metadata souborového systému
    - alokace nových bloků, aktualizace seznamu volných bloků
  - datová část souborového systému
    - zápis nových dat souboru
  - metadata souboru
    - aktualizace přidělených bloků, velikosti, času změny
- změny probíhají postupně – vznik nekonzistence
  - po pádu systému je třeba provést kontrolu

Zápis do souboru (obrázek 1)



výchozí stav

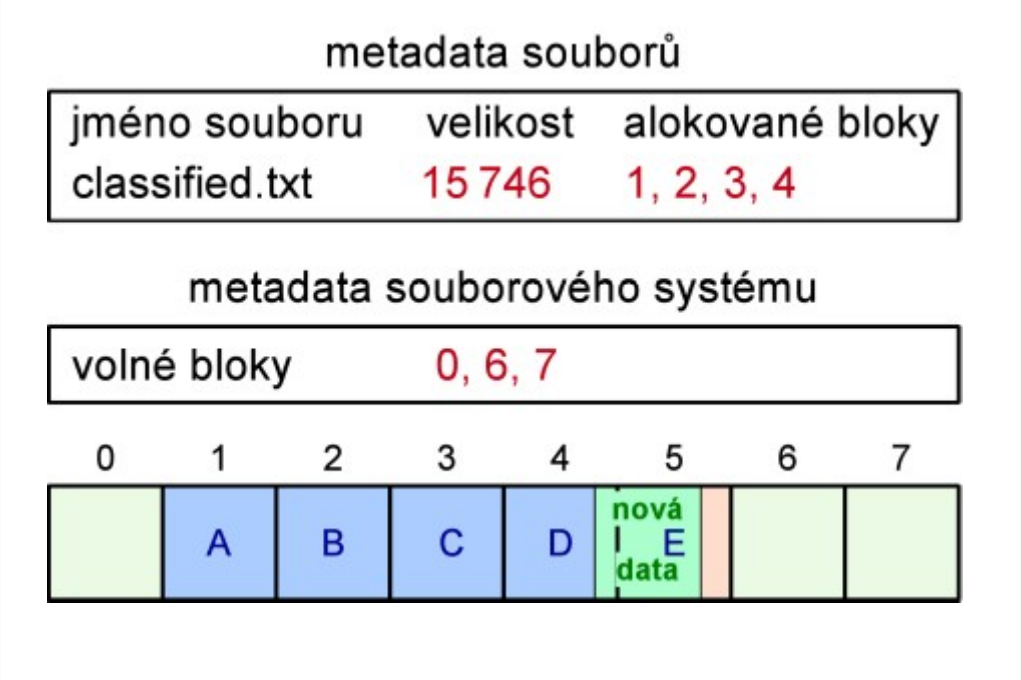
Zápis do souboru (obrázek 2)



alokace bloků na souborovém systému

červeně jsou uvedena nekonzistentní metadata

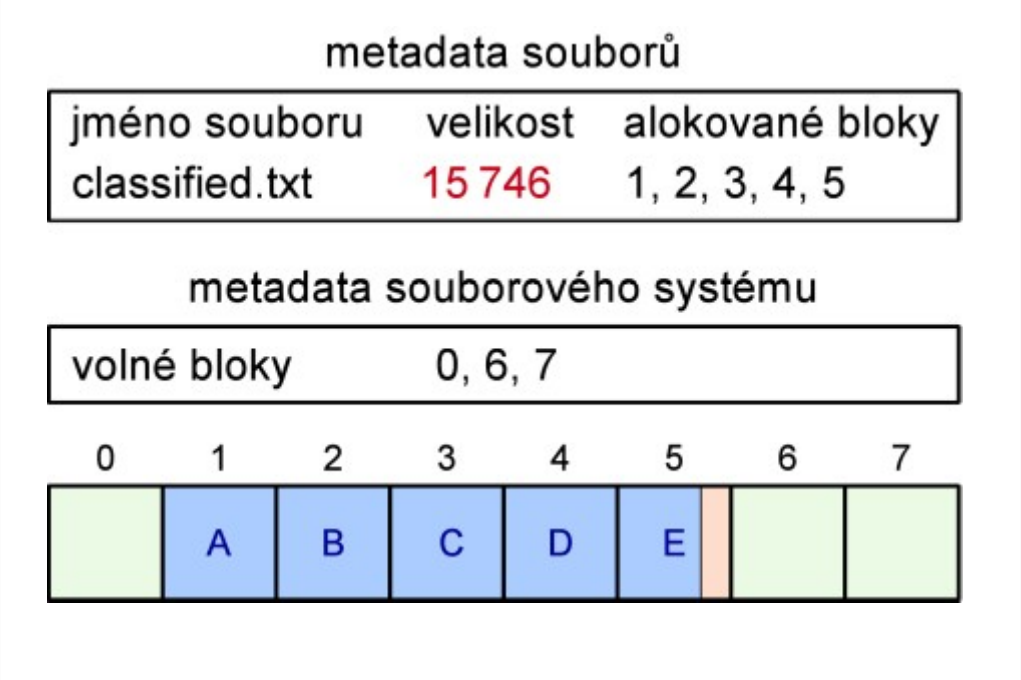
Zápis do souboru (obrázek 3)



zápis nových dat do souboru

červeně jsou uvedena nekonzistentní metadata

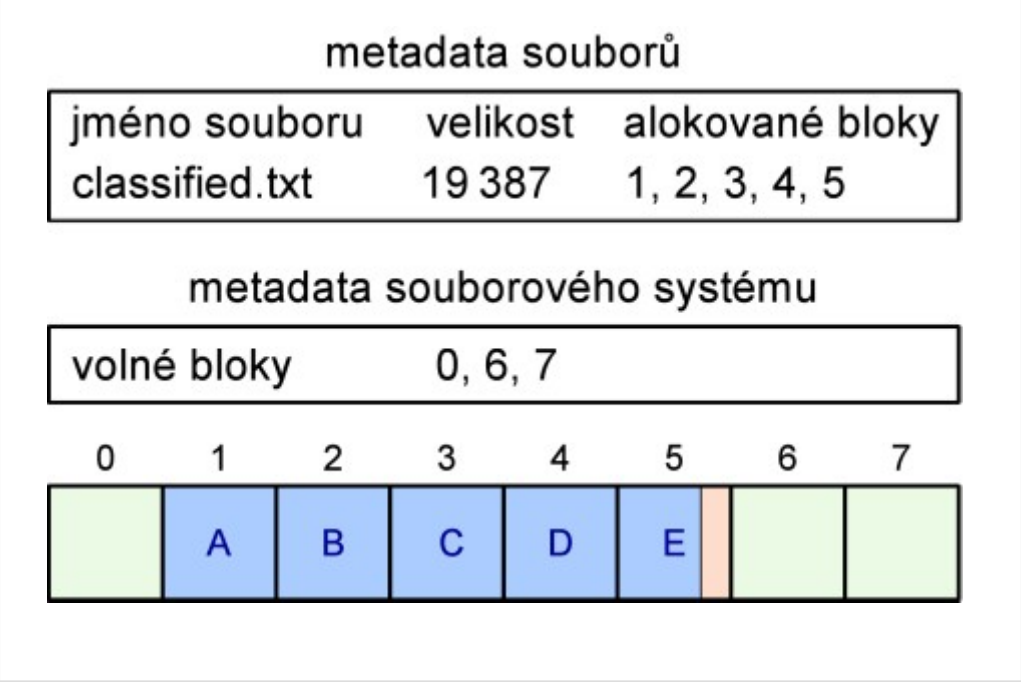
Zápis do souboru (obrázek 4)



aktualizace alokovaných bloků souboru

červeně jsou uvedena nekonzistentní metadata

Zápis do souboru (obrázek 5)



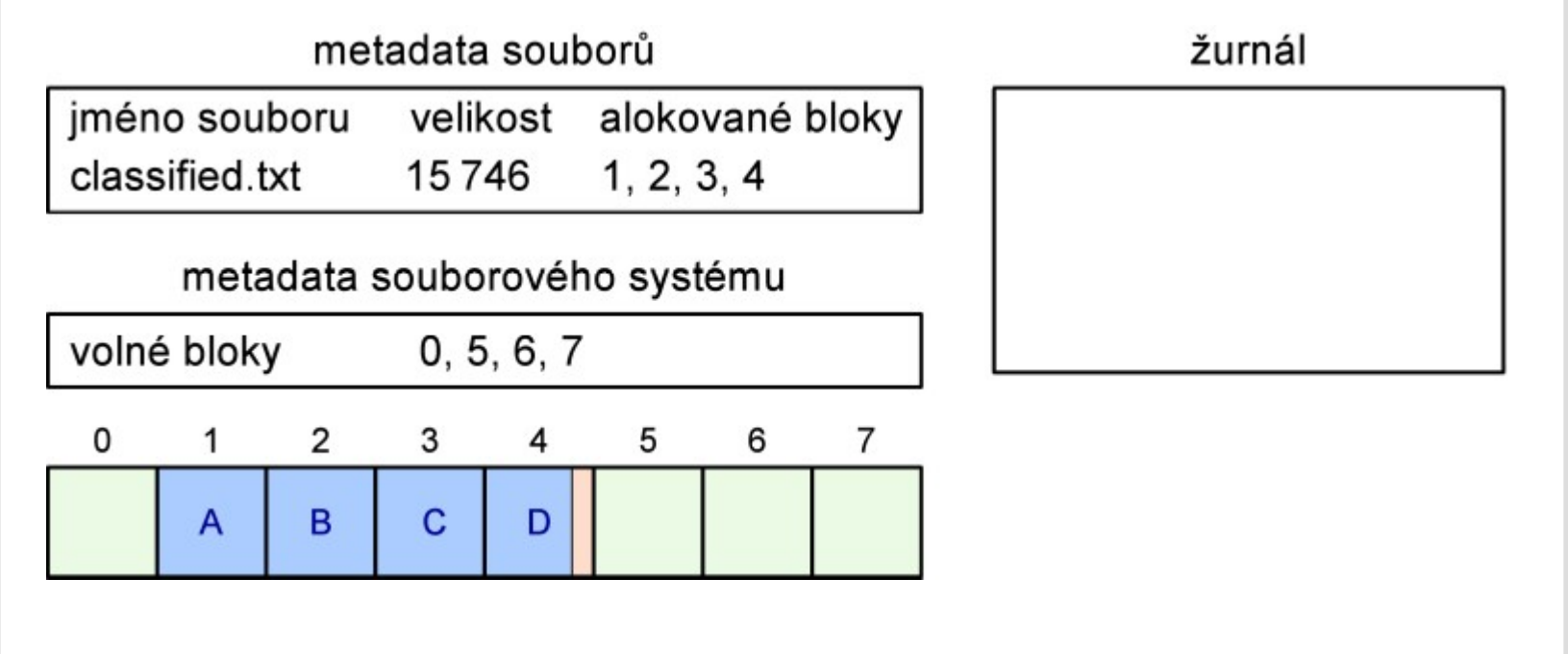
aktualizace velikosti souboru

konečný stav

Zachování konzistence – žurnál

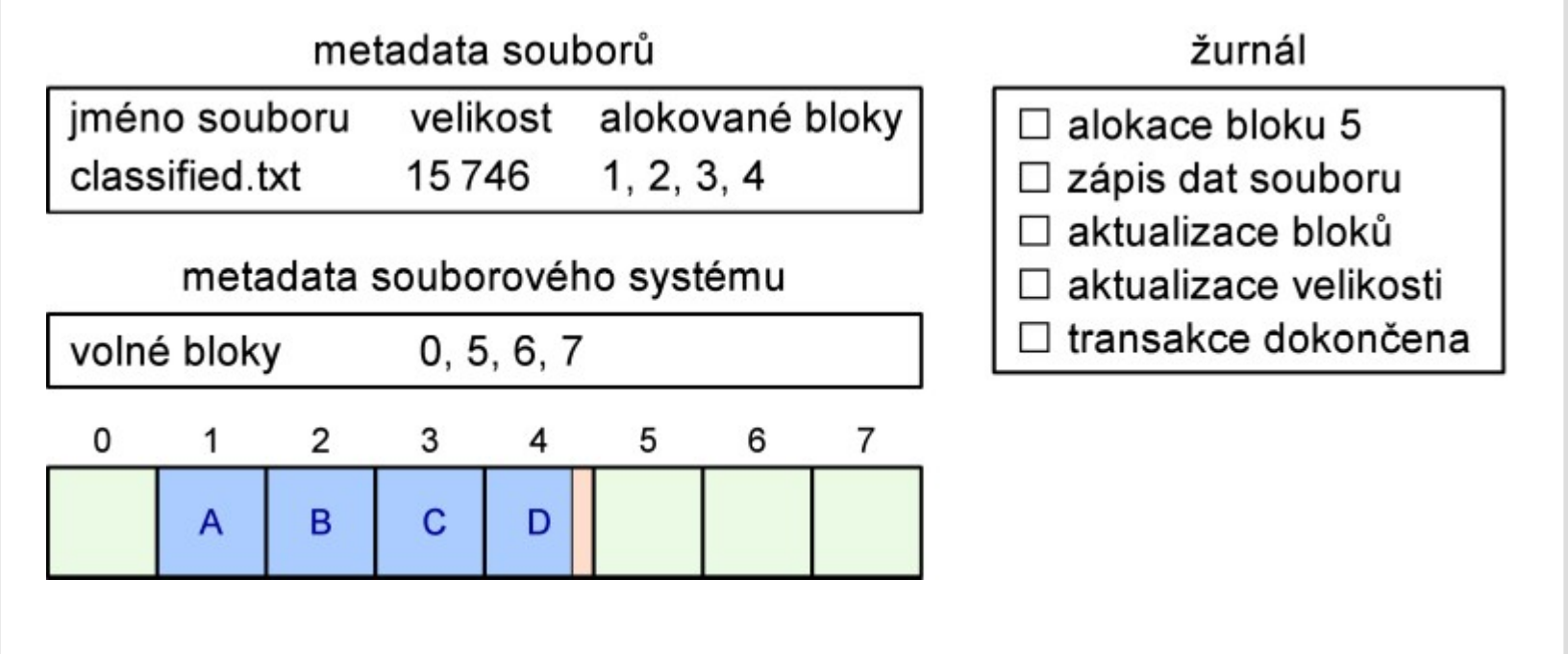
- žurnál (journal) – transakční log
  - zabraňuje ztrátě konzistence dat při pádu systému
    - žurnál nezabraňuje ztrátě dat při pádu systému
  - změny metadat (případně i dat) se zapisují do transakčního logu, což je kruhový buffer
    - teprve po potvrzení zápisu do žurnálu se provede patřičná změna souborového systému – dvojí zápis = zpomalení
  - při startu systému po pádu není třeba kontrolovat konzistenci celého souborového systému – zrychlení
    - prochází se žurnál a zkontroluje se (a případně se opraví) konzistence pouze na místech posledních změn

Zápis do souboru s použitím žurnálu (obrázek 1)



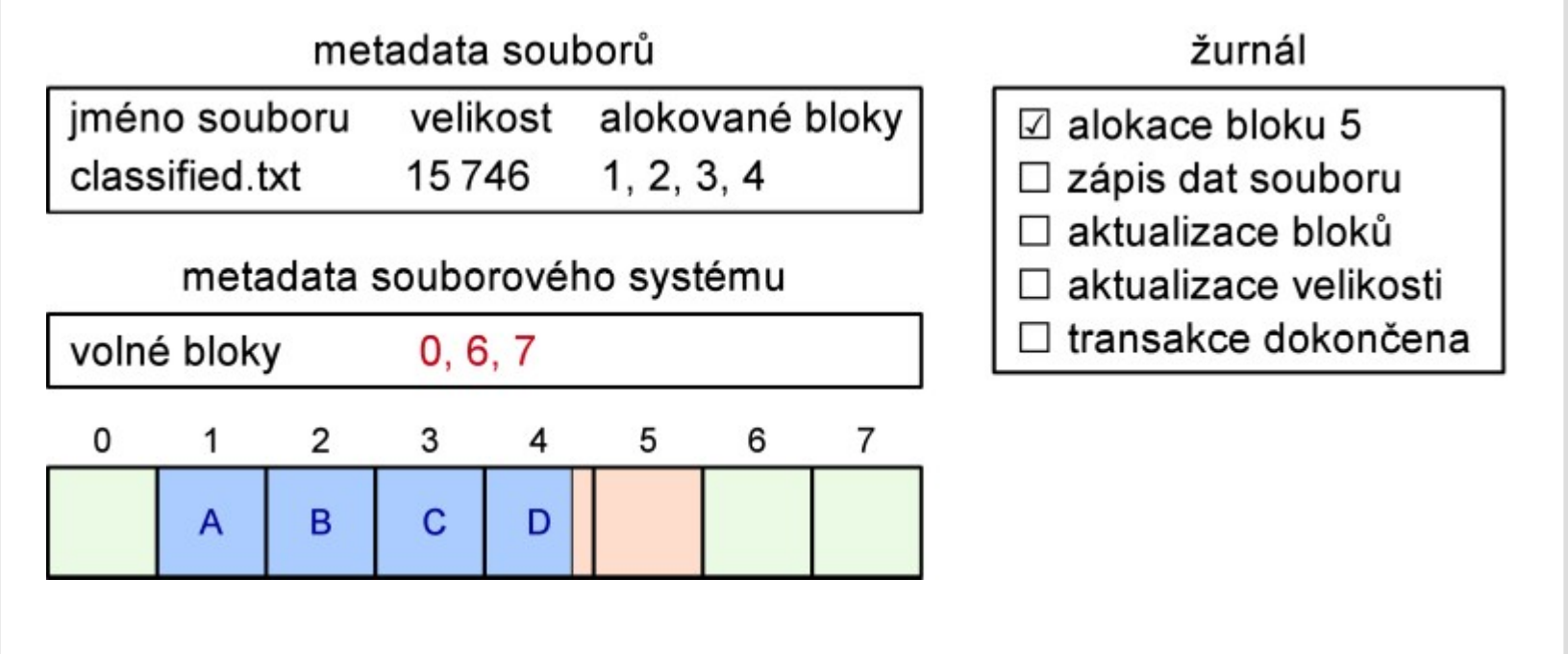
výchozí stav

Zápis do souboru s použitím žurnálu (obrázek 2)



zápis transakce do žurnálu

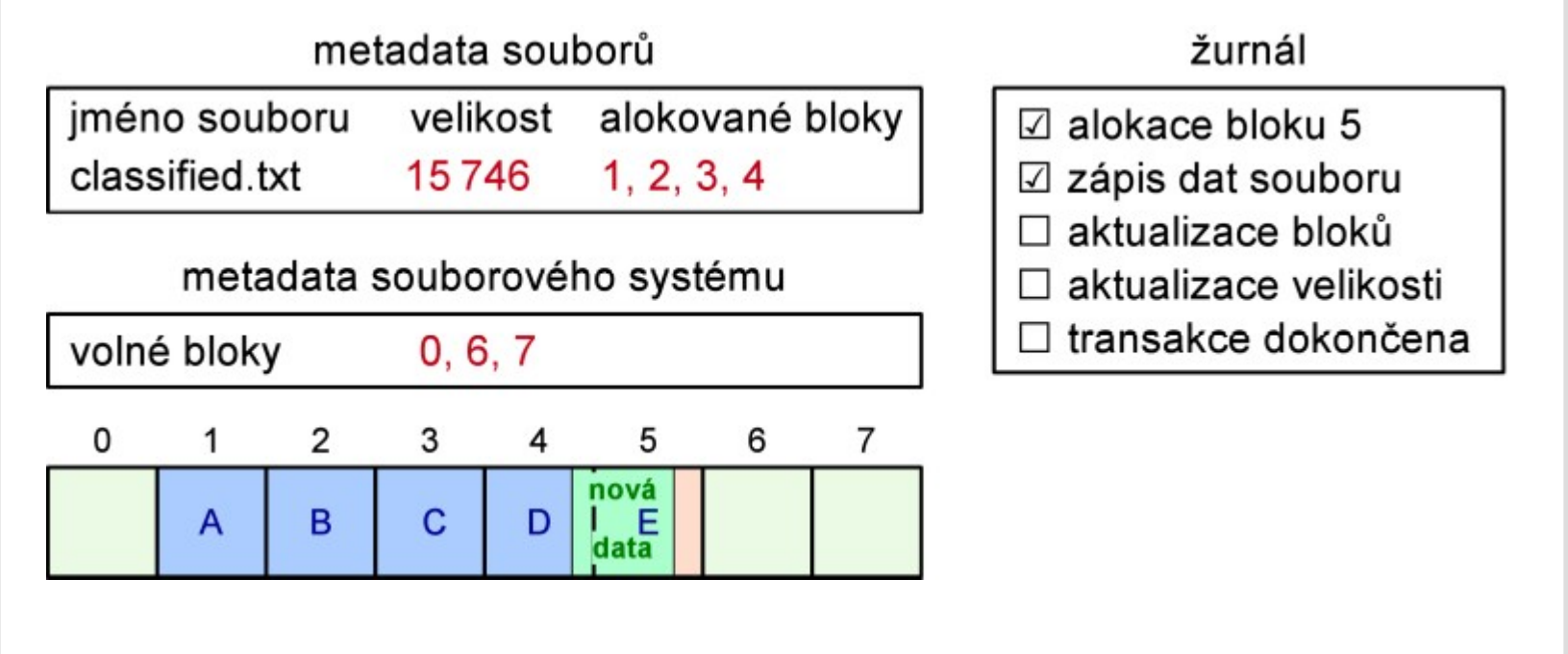
Zápis do souboru s použitím žurnálu (obrázek 3)



alokace bloků na souborovém systému

červeně jsou uvedena nekonzistentní metadata

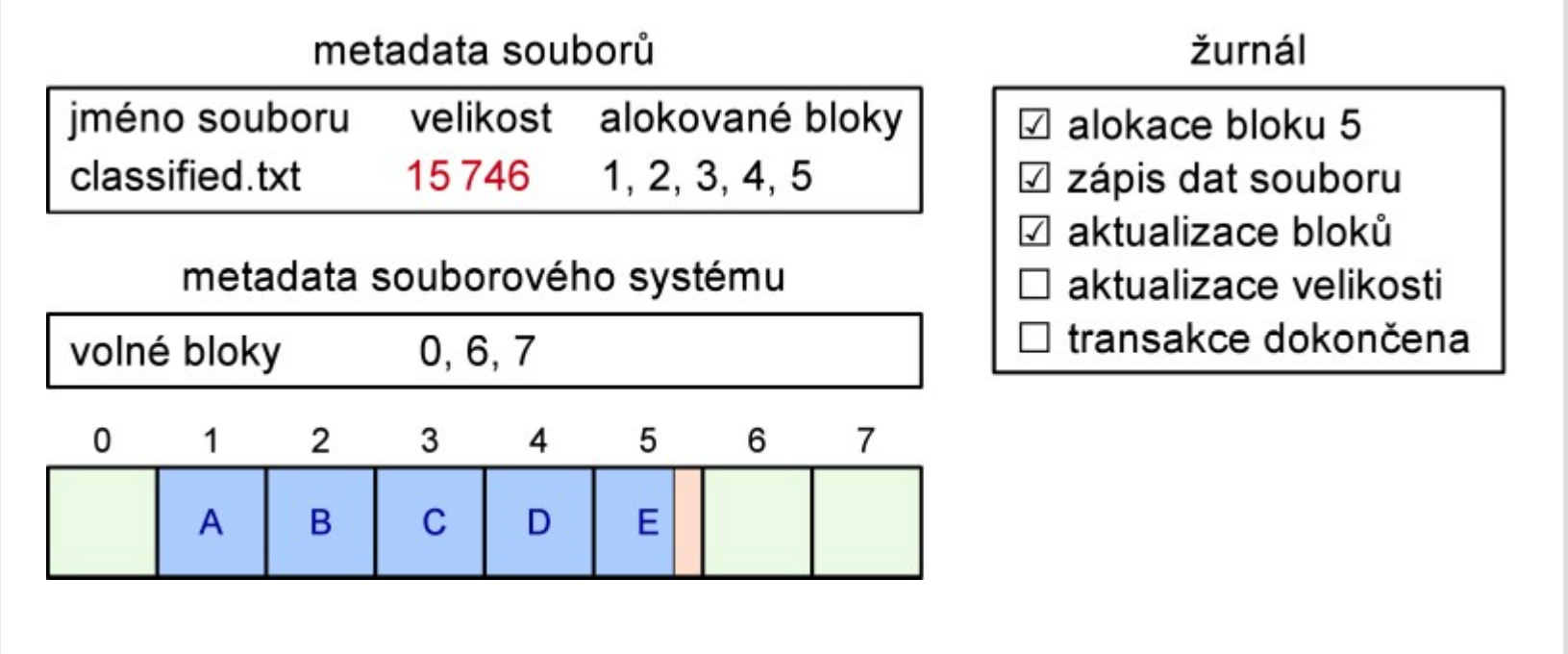
Zápis do souboru s použitím žurnálu (obrázek 4)



zápis nových dat do souboru

červeně jsou uvedena nekonzistentní metadata

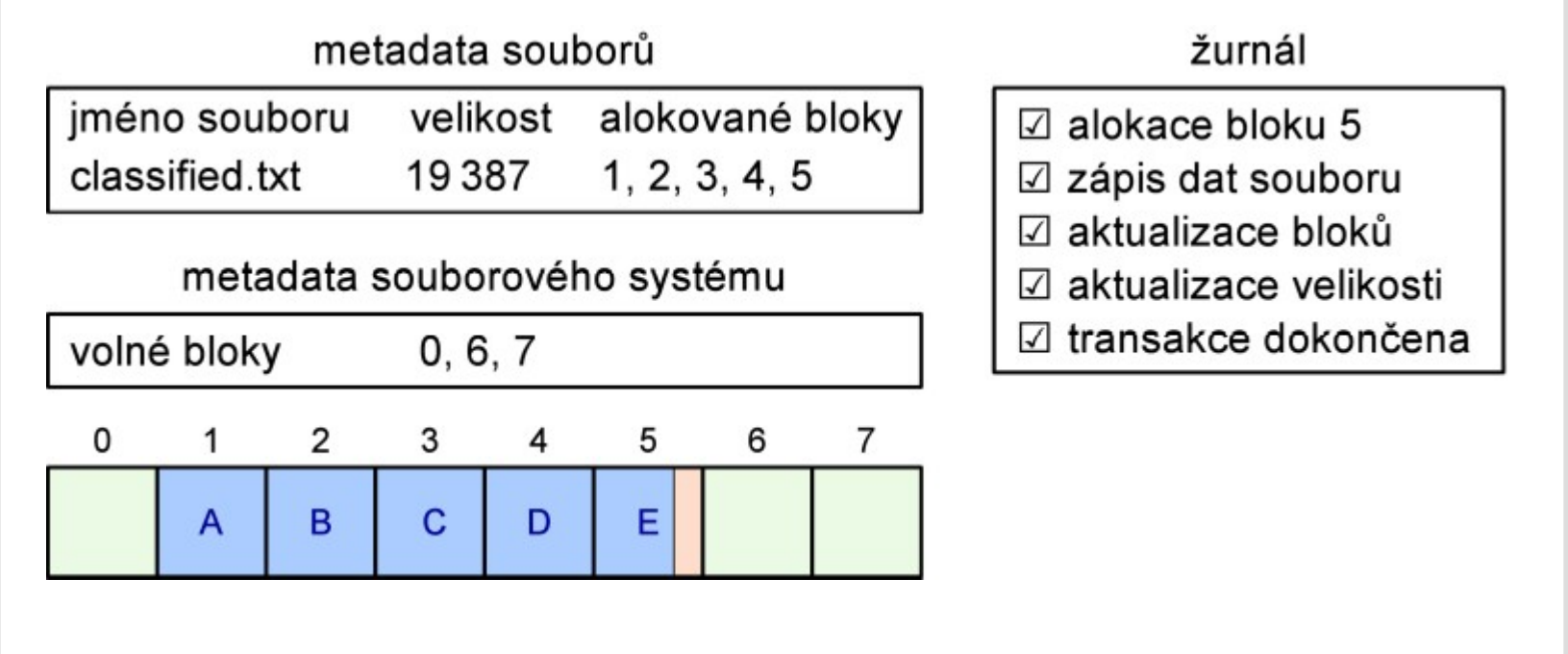
Zápis do souboru s použitím žurnálu (obrázek 5)



aktualizace alokovaných bloků souboru

červeně jsou uvedena nekonzistentní metadata

Zápis do souboru s použitím žurnálu (obrázek 6)



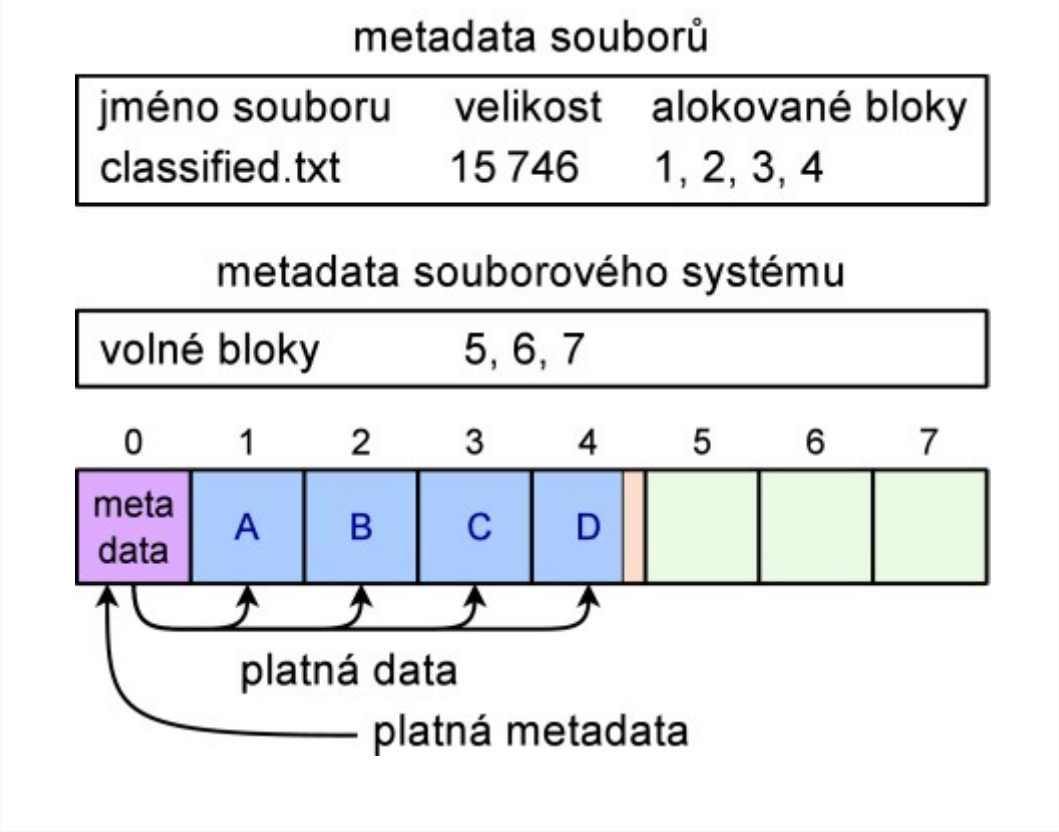
aktualizace velikosti souboru

dokončení transakce, konečný stav

Zachování konzistence – copy-on-write

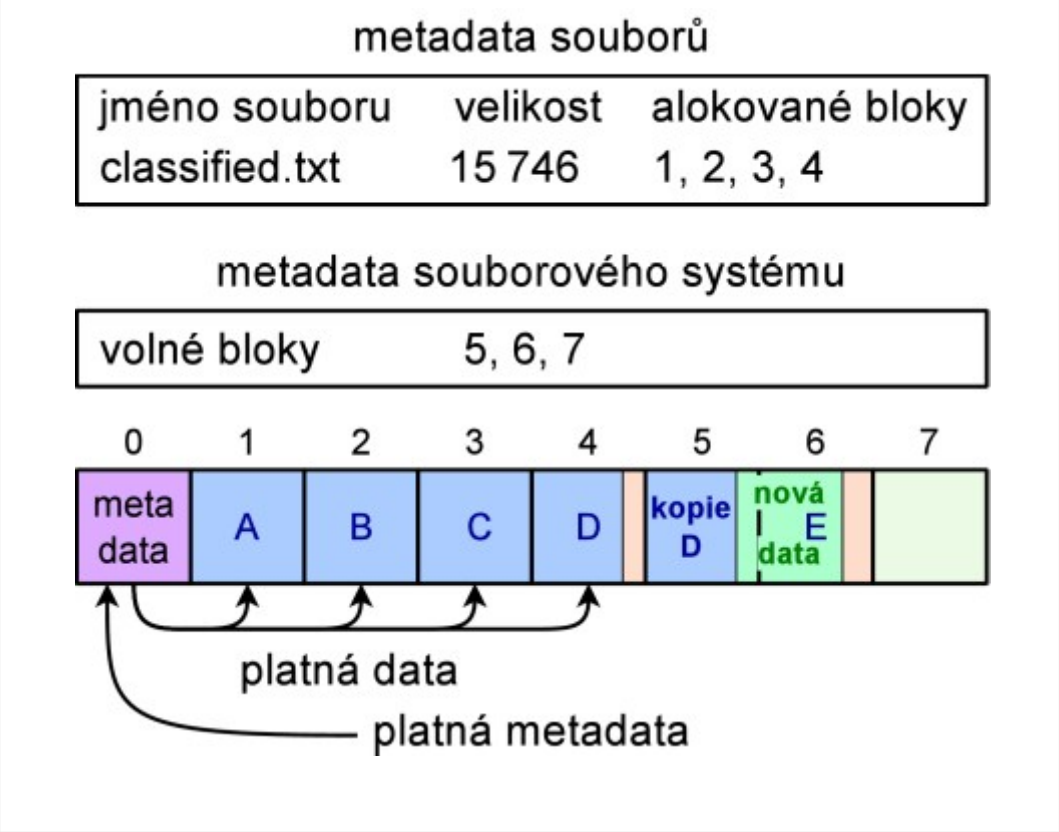
- copy-on-write – kopírování bloku při změně
  - nekonzistence vzniká při změně informací – zápisu
  - souborový systém nikdy nemodifikuje bloky na místě
    - při přepisu se modifikuje kopie bloku
    - stejným způsobem se aktualizují metadata
    - po dokončení zápisu se atomicky provede zneplatnění původních dat a metadat a potvrdí se platnost nových
    - souborový systém je tedy vždy konzistentní
    - data se nezapisují dvakrát (nevýhoda žurnálu odstraněna)
    - nevýhoda: větší datová fragmentace
  - souborové systémy: ZFS, BtrFS, NILFS, F2FS, ReFS

Zápis do souboru metodou copy-on-write (obrázek 1)



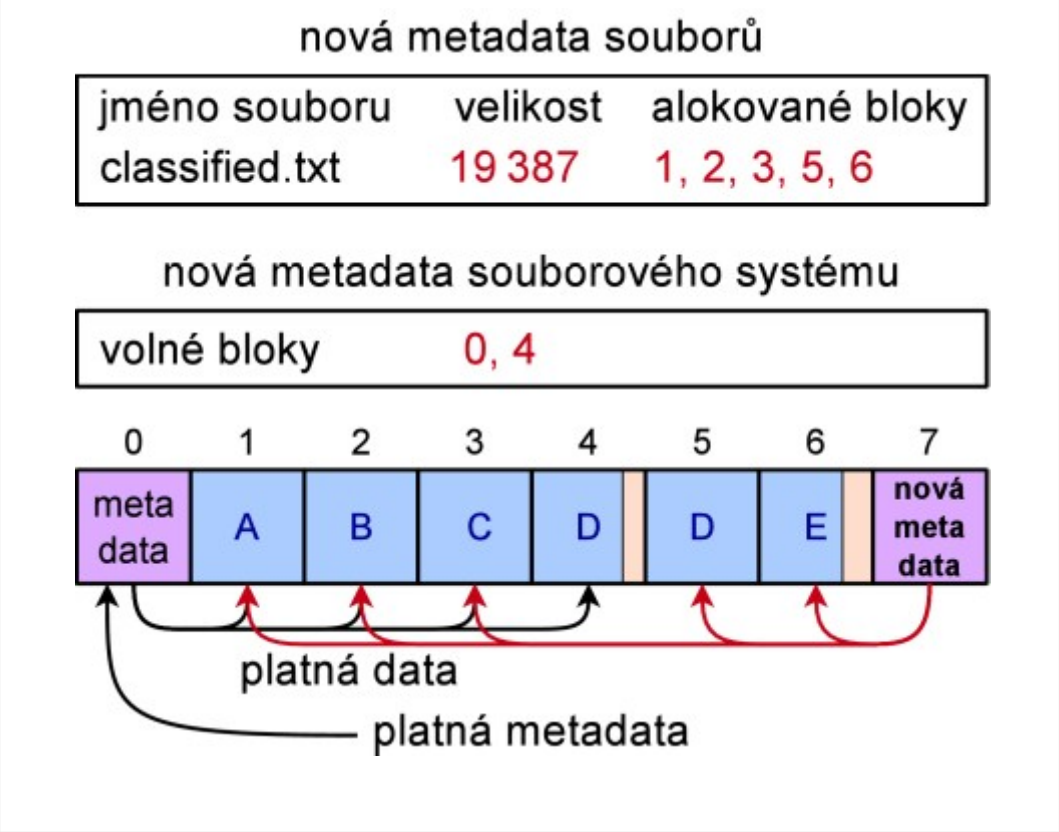
výchozí stav

Zápis do souboru metodou copy-on-write (obrázek 2)



zápis nových dat souboru

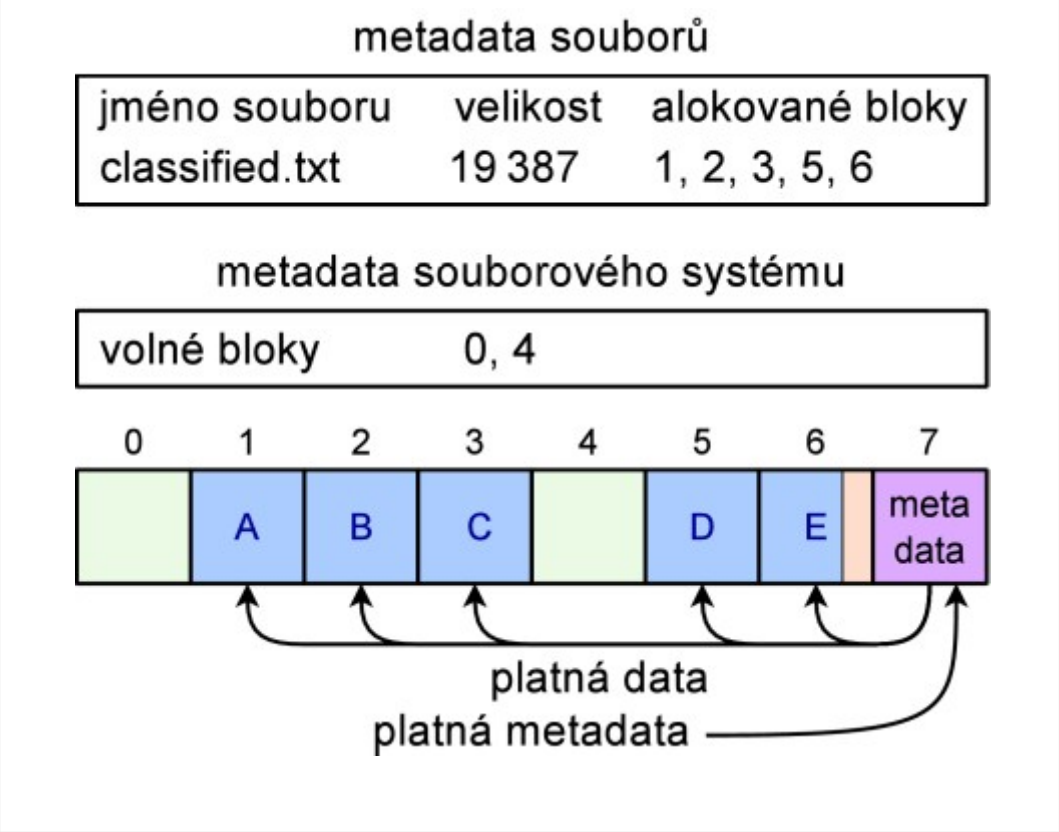
Zápis do souboru metodou copy-on-write (obrázek 3)



zápis nových metadat

zatím je stále platný blok s původními metadaty

Zápis do souboru metodou copy-on-write (obrázek 4)



nastavení platnosti nových dat a metadat

konečný stav

Konzistence souborového systému – příklady

- souborové systémy bez konzistence po pádu
  - FAT, exFAT, ext2, HFS (Apple), HPFS (IBM OS/2)
- žurnálovací souborové systémy
  - úplné: ext3, ext4, ReiserFS, UDF (transakční)
  - pouze metadata: NTFS (MS, transakční), JFS (IBM), XFS (SGI), HFS+ (od Mac OS X 10.2.2), UFS (od Solaris 7, FreeBSD 7.0), VMFS (VMware)
- souborové systémy s podporou copy-on-write
  - ZFS, BtrFS, NILFS, F2FS, ReFS

Speciální soubory – odkazy

- tvrdý (pevný) odkaz (hard link)
  - více jmen pro týž soubor (ne adresář) na stejném FS
  - data i metadata jsou na médiu uložena pouze jednou
  - ke smazání dojde při odstranění posledního odkazu
- symbolický odkaz (symbolic link), speciální typ
  - textová záměna za jiné jméno (absolutní či relativní)
  - změna jména je pro procesy transparentní
    - liší se od zástupce (shortcut), což je normální soubor
- NTFS junction point – odkaz na adresář

Speciální soubory – roura, socket, zařízení

- pojmenovaná roura (named pipe)
  - jednosměrný komunikační nástroj pro dva procesy
    - jeden proces zapisuje, jeden čte (má nezávislé ukazatele)
  - co bylo přečteno se odstraní
- pojmenovaný socket – adresa = jméno souboru
- zařízení (device file) – jednotný přístup k HW
  - blokový (s náhodným přístupem) – disky
  - znakový (proudový přístup po bajtech)
    - terminál, tiskárna, zvuková karta, skener

Speciální soubory – identifikace

- POSIX: první znak na řádku ve výpise `ls -l`
  - directory – adresář: jména souborů s čísly i-uzlů
  - link – symbolický odkaz: obsahuje jméno cíle
    - pozn.: hard link není speciální soubor
  - pipe – pojmenovaná roura
  - socket – pojmenovaný soket
  - block – blokový speciální soubor: bloková zařízení
  - char – znakový speciální soubor: znaková zařízení
    - identifikace zařízení pomocí čísel (hlavní a vedlejší)

Kritéria pro souborové systémy

- rychlost přístupu
- snadnost modifikace
- ekonomie využití paměti
  - minimalizace nevyužitelného prostoru (uvnitř bloků)
  - minimalizace redundance (indexace, žurnály)
- snadnost údržby
- spolehlivost – zachování konzistence
- podpora funkcí – práva, kvóty, speciální soubory

Oprávnění obecně (1)

- žádná – se souborem nelze nijak manipulovat
  - nelze zjistit ani existenci souboru
- znalost existence – bez dalších oprávnění
- provádění (execute) – soubor lze spustit
  - soubor nemusí být možné číst (tedy ani kopírovat)
- čtení (read) – lze zjistit obsah souboru
  - lze tedy soubor zkopírovat

Oprávnění obecně (2)

- přidávání (append) – lze přidávat data na konec
- přepisování, zápis (update, write)
  - do souboru lze zapisovat a přepisovat jeho data
- mazání (delete) – soubor lze odstranit
- vytvoření (create) – lze vytvořit nový soubor
- změna oprávnění (changing permissions)
  - změna přístupových práv jiným uživatelům

Příklady oprávnění – UNIX

- unixové systémy
  - zvlášť práva pro vlastníka, skupinu a ostatní
    - soubor je vlastněn jedním uživatelem a jednou skupinou
  - read – čtení obsahu souboru
    - adresář: zobrazení položek adresáře
  - write – změna obsahu souboru
    - adresář: vytváření a mazání položek (libovolného vlastníka) v adresáři
  - execute – spuštění programu
    - adresář: vstup do adresáře, čtení metadat položek

Příklady oprávnění – Novell Netware

- Novell Netware
  - supervisory – všechna práva
  - access control – přidělování práv ostatním
  - read – čtení, adresář: vypsat obsah
  - write – zápis, adresář: zápis do existujících souborů
  - create – adresář: vytváření souborů,
    soubor: obnovení smazaného souboru
  - erase – smazání
  - modify – změna atributů (nikoliv práv)
  - file scan – znalost existence souboru

Příklady oprávnění – NTFS

- NTFS (New Technology File System, Windows)
  - Full Control, Traverse 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, Change Ownership

Příklady atributů – FAT

- souborový systém FAT
  - read-only – soubor je pouze pro čtení
    - soubor nelze smazat ani přepsat
  - hidden – skrytý, nezobrazuje se ve výpise adresáře
  - system – systémový, skrytý + omezení odstranění
  - archive – soubor je připraven k archivaci (záloze)
    - atribut se nastavuje vždy při zápisu do souboru

Příklady atributů – NTFS

- souborový systém NTFS
  - read-only, hidden, system, archive – jako FAT
  - not content indexed – neindexovaný obsah souboru
  - off-line – data jsou na jiném (vzdáleném) zařízení
    - soubor není přístupný při provozu off-line
    - může též znamenat, že soubor je jen lokální kopií (cache)
  - temporary – dočasný soubor
  - compressed – automaticky komprimovaný soubor
  - encrypted – šifrovaný pomocí EFS
  - sparse – řídký soubor

Příklady atributů – extended FS (1)

- linuxový souborový systém ext2, ext3, ext4
  - append only – lze pouze přidávat na konec
  - compressed – komprimovaný (pouze ext4)
  - no dump – nearchivovat programem dump,
  - extent format – pro alokaci se používají velké souvislé oblasti bloků (extents, pouze ext4)
    - redukce datové fragmentace
  - immutable – soubor nelze nijak měnit ani odstranit
  - data journaling – žurnálovat i data souboru (od ext3)
  - no Atime updates – neaktualizuje se čas přístupu

Příklady atributů – extended FS (2)

- linuxový souborový systém ext2, ext3, ext4
  - secure deletion – bezpečné mazání (pouze ext4)
    - při mazání se obsah bloků přepíše nulami
  - undeletable – při mazání se uloží data (pouze ext4)
    - umožní obnovu smazaného souboru
  - no tail-merging – neslučovat poslední blok souboru
  - Synchronous updates – synchronní zápis do souboru
  - synchronous Directory updates – dtto do adresáře
  - Top of directory hierarchy – podadresáře nesouvisejí
    - alokace místa podadresářům na různých místech na médiu

Linuxové souborové systémy 1

- ext2/3/4 – nativní linuxový
  - ext3 (2001, jádro 2.4.15) – přidává žurnál
  - ext4 (2008, 2.6.19) – vyšší limity, extenty, …
- ReiserFS (Hans Reiser)
  - první žurnálovací FS v Linuxu (2001, 2.4.1)
  - efektivní pro mnoho malých souborů v adresáři
- JFS (IBM AIX a OS/2, Linux 2.4.20 2002)
  - JFS1: AIX 3.1 1990, JFS2: OS/2 1999, AIX 5L 2001
  - strom B+ pro rychlé hledání, extenty, komprese

Linuxové souborové systémy 2

- XFS (SGI IRIX 5.3 1993, Linux 2004)
  - patch dostupný od jádra 2.4.2 (2001)
  - efektivní pro paralelní ukládání, škálovatelný
    - strom B+ (var. B*) pro rychlé hledání, extenty, freeze
  - Red Hat: výchozí FS od RHEL 7 (2014)
    - Red Hat nadále investuje do jeho vylepšování
  - žurnálování založené na checkpoints
  - podpora sdílených extentů, svazků (subvolume)
    - copy-on-write – pouze pro data (2018)

Linuxové souborové systémy 3

- BtrFS (vývoj zahájen 2007: IBM, Oracle)
  - verze 1.0 v Linuxu 2.6.29 (2009)
    - „on-disk format“ prohlášen za stabilní až 2014
  - CoW, defragmentace a přidávání zařízení online, vícesvazkový, RAID, snímky, komprese, zálohování
    - plán: šifrování, kontrola online, RAID5, inkrementální zál.
  - má nahradit ext4 (ext4 lze povýšit na BtrFS)
  - RHEL 6.8 (2016) a 7.4 (2017) končí podporu

Linuxové souborové systémy 4

- NILFS (Linux 2.6.13 2005)
  - New Implementation of a Log-structured FS
    - specifiký typ CoW vhodný pro SSD a flash
  - continuous checkpoints, lze je připojit (read-only)
  - nízká latence
- F2FS (Samsung 2012, Linux 3.8 2012)
  - log-formát pro NAND flash (SSD, eMMC, SD)
  - vysoká propustnost, defragmentace online, atomické operace, checkpoints

Souborový systém ZFS (1)

- Zettabyte File System, pro Solaris 10, 2004
  - Sun: „*The last word in filesystems.*“
  - obsahuje vrstvu správy oddílů včetně podpory RAID
  - (téměř) neomezená kapacita
    - 256 ZiB – max. velikost zpool, max. 2^{64} zpools
    - 16 EiB – max. velikost souboru
    - až 2^{48} atributů souboru, max. velikost atributu 16 EiB
    - variabilní velikost alokačního bloku
  - téměř žádná nutná správa, automount, online scrub
  - dynamické rozložení zátěže (dynamic stripping)

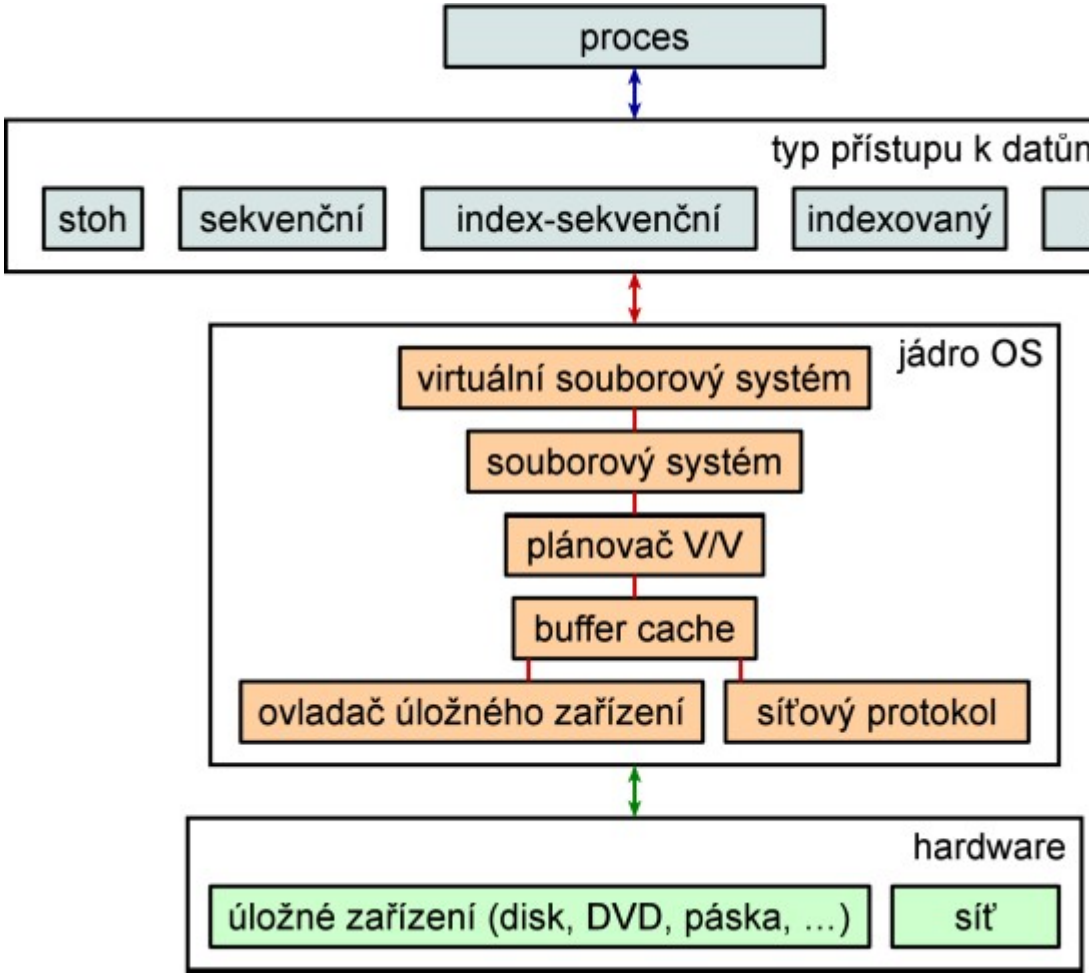
Souborový systém ZFS (2)

- Zettabyte File System
  - prověřená integrita dat – kontrolní součty bloků
    - možnost zapisovat data redundantně + self healing
  - transakční model copy-on-write + cache
    - ZIL (ZFS Intent Log, zápisová cache) lze umístit na SSD
    - ARC (Adaptive Replacement Cache, čtecí cache) v RAM + L2ARC na disku (SSD)
  - snapshots, clones (zapisovatelný snapshot), dump
  - deduplikace dat
  - komprese, šifrování, kvóty, rezervace místa

OpenZFS

- OpenSolaris – otevřená licence CDDL (2005)
  - 2008: ZFS zařazeno do FreeBSD 7.0
  - 2008: začala nativní implementace pro Linux (ZoL)
  - 2010: konec podpory (Oracle), fork OS: illumos
  - 2013: ZFS pro Linux stabilní
    - podporováno např. od Ubuntu 16.04 LTS (2016)
    - verze 0.8 (2019): šifrování, autotrim, projektové kvóty, pool checkpoints, direct IO
  - 2013–2020: sjednocení zdrojů FreeBSD + Linux
    - OpenZFS 2.0 (plán vydání 2020), v 3.0 + macOS (2021)

Architektura správy souborů (obrázek)



Architektura správy souborů (1)

- proces přistupuje k datům dle jejich struktury
- virtuální souborový systém implementuje základní operace se soubory + metadata
  - vytvoření, otevření, zavření, čtení/zápis, atributy
  - může uchovávat data v operační paměti (cache)
- skutečný souborový systém určuje jakým způsobem jsou data s metadaty uložena
  - udržuje řídicí struktury (volné místo, adresáře)
  - eviduje přiřazení logických bloků souborům

Architektura správy souborů (2)

- plánovač základních V/V operací odpovídá za
  - inicializaci a ukončení vstupně-výstupních operací
  - plánování a řazení nevyřízených operací
    - výkonová optimalizace
  - používá část operační paměti jako buffer cache pro urychlení operací – zapisuje data se zpožděním
- ovladač úložného zařízení
  - odpovídá za správné příkazy pro hardware
  - zpracovává hardwarové přerušovací signály vyvolané V/V operacemi

Organizace souborů – terminologie

- pole (field)
  - základní element dat (proměnná)
  - charakterizováno délkou a datovým typem
- záznam (record)
  - seskupení vzájemně souvisejících polí
  - jednotka dat (záznam zaměstnance, výrobku, …)
- soubor (file)
  - entita – posloupnost podobných záznamů
  - identifikován jménem (a cestou)

Organizace souborů – stoh

- stoh (pile)
  - soubor bez nutné vnitřní struktury
  - ukládání záznamů s různými poli
  - přístup k záznamu – úplné prohledání (exhaustive search)
  - příklady:
    - textové soubory – záznamy jsou řádky
    - ploché (flat) soubory – např. /etc/passwd

Organizace souborů – sekvenční soubor

- sekvenční soubor (sequential file)
  - všechny záznamy mají stejnou délku i strukturu všech polí (fixed format)
  - jedno pole je jednoznačným klíčem
  - záznamy mohou být uspořádány dle klíče
    - nové záznamy se ukládají do transakčního logu
    - hlavní soubor se aktualizuje dávkově po seřazení všech záznamů – šetří čas a snižuje riziko ztráty dat

Organizace souborů – index-sekvenční soubor

- index-sekvenční soubor (indexed sequential file)
  - sekvenční soubor + index klíčů s ukazatelem na záznam v hlavním souboru
    - úplný (exhaustive) index – nad neseřazeným souborem
    - neúplný index – záznam se dohledává sekvenčně
  - výrazná úspora času oproti sekvenčnímu hledání
    - např. milion záznamů – čistě sekvenční procházení potřebuje průměrně 500 000 přístupů
    - index pro každý 1 000. záznam – průměrně 1 000 přístupů (500 přístupů pro sekvenční hledání v indexu a 500 přístupů v hlavním souboru) ⇒ zlepšení 500×

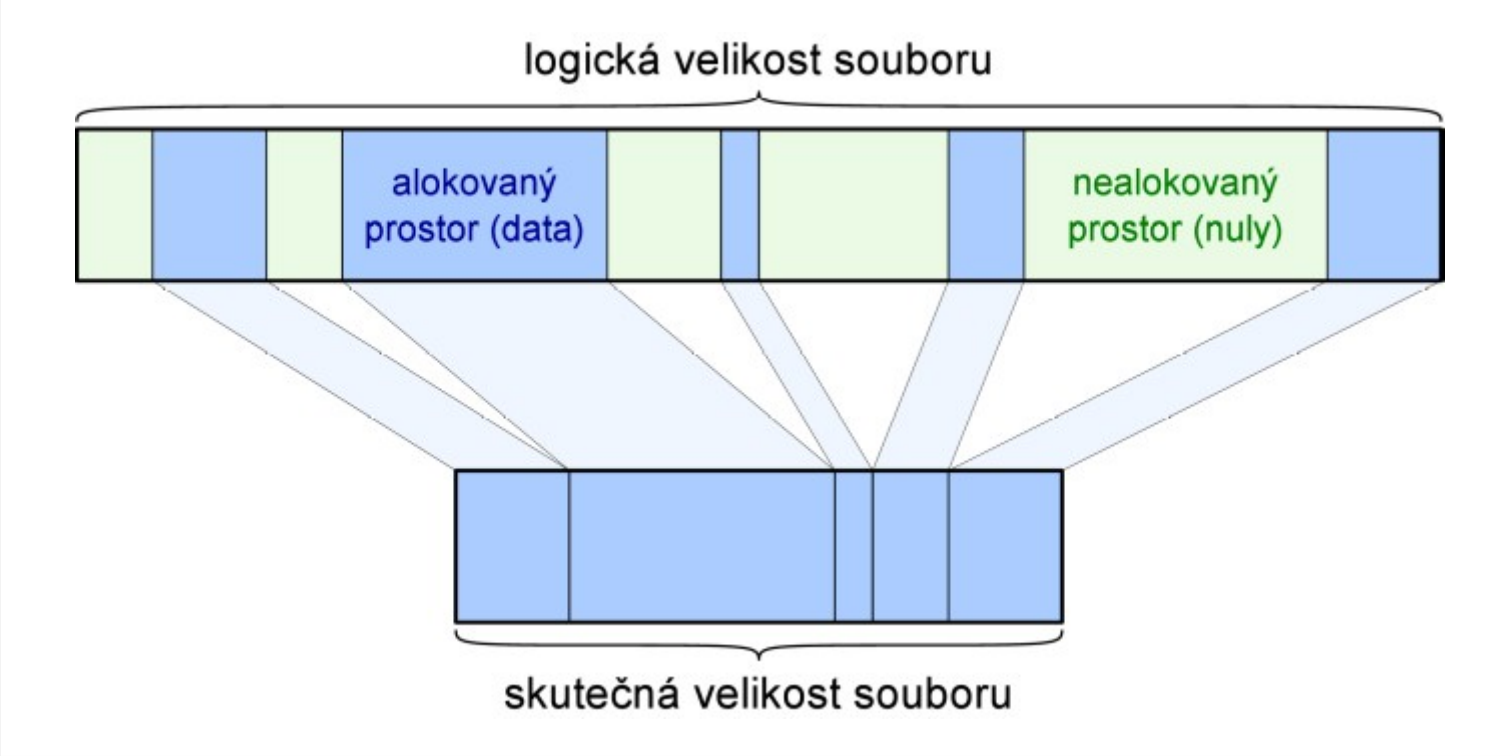
Organizace souborů – indexovaný soubor

- indexovaný soubor (indexed file)
  - soubor + indexy pro různá klíčová pole s ukazateli na záznamy v hlavním souboru
  - úplný index klíčových polí, podle kterých není soubor seřazen
  - nad seřazenými záznamy může být i neúplný index
    - konkrétní záznam je nutné (sekvenčně) dohledat

Organizace souborů – přímý soubor

- soubor s přímým přístupem (hashed / direct file)
  - adresa bloku, ve kterém je uložen záznam, je odvozena z klíče pomocí hešovací funkce
    - každý záznam musí mít klíč
  - více různých klíčů může dávat stejnou adresu (kolize hešovací funkce) – po zaplnění bloku jsou záznamy ukládány
    - na volná místa v následujících blocích nebo
    - do přeplňovacího souboru (overflow file)
  - lze využít tzv. řídké soubory (sparse files)

Řídký soubor (obrázek)



úseky nul jsou reprezentovány metadaty

nejsou pro ně alokovány žádné bloky