

Principy počítačů a operačních systémů

Operační systémy
Souborové systémy

Zimní semestr 2011/2012

Poděkování

Při přípravě této prezentace jsem většinu materiálu převzal z prezentace

- Yaghob, J. **Základy operačních systémů**. Katedra SW inženýrství, Matematicko-fyzikální fakulta, Univerzita Karlova v Praze, 2007.



Proč potřebujeme souborové systémy?

Kapacita

- v hlavní paměti je možné pracovat pouze s omezeným množstvím informace
 - ♦ soubor umožňuje uložit obrovské množství dat

Perzistence dat

- obsah paměti je při ukončení procesu ztracen
 - ♦ informace v souboru přetrvává (perzistence)

Sdílení/výměna dat

- sdílení informace mezi více procesy
 - ♦ k souboru může přistupovat více procesů současně
- přenos informace mezi počítači



Soubory – obecné koncepty

Co je to soubor?

- soubor je perzistentní úložiště nestrukturovaných nebo strukturovaných dat, umístěné v pomocné paměti

Identifikace

- soubor má jméno, aby bylo možné se na něj odkazovat po dobu, která značně přesahuje dobu běhu programu, který ho vytvořil

Abstrakce

- soubor je abstrakce, která umožňuje uložit informaci na disk a později ji přečíst
 - ♦ odstiňuje uživatele od podrobností práce s disky



Soubory z pohledu OS

Informace o souboru (metadata)

- jméno
- atributy
- struktura
- typ

Práce se soubory

- druhy přístupu
- operace



Identifikace

- umožňuje lidskému uživateli přístup k jeho datům

Přesná pravidla pojmenování určuje OS

- délka jména
- malá vs. velká písmenka
- speciální znaky
- přípony a jejich význam



Atributy souboru

Ostatní informace

- definují vlastnosti a uchovávají informace o souboru

Atributy opět určuje OS

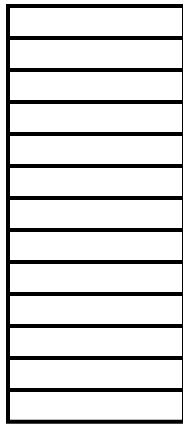
- jméno, typ
- velikost, umístění na disku
- vlastník, přístupová práva
- čas vytvoření, zápisu, přístupu



Struktura souboru

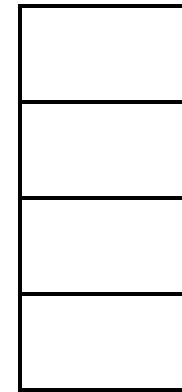
Sekvence bajtů

- Windows, UNIX

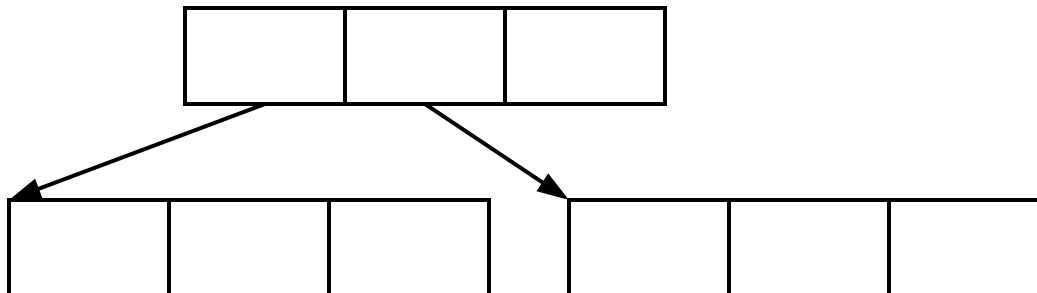


Sekvence záznamů

- IBM mainframes



Strom



Typ souboru

Běžné soubory

- obsahují data, z pohledu operačního systému se jedná pouze o pojmenovaný souvislý blok dat
- vnitřní struktura určena programem, který soubor vytváří

Adresáře

- systémové soubory vytvářející logickou strukturu souborového systému
- vnitřní struktura určena souborovým systémem

Speciální soubory

- znaková/bloková zařízení, roury, sockety, symbolické odkazy



Přístup k souborům

Sekvenční

- pouze pohyb vpřed, možný rewind
- umožňuje OS přednačítání

Přímý přístup

- umožňuje měnit aktuální pozici

Mapování do paměti

- využití stránkování



Soubory mapované do paměti

„pojmenovaná“ virtuální paměť

- program přistupuje k souboru pomocí běžných instrukcí pro práci s pamětí
 - ♦ soubor “vypadá” v paměti jako pole bajtů
- ušetří se kopírování souboru po paměti

Problémy

- přesná velikost souboru
- zvětšování souboru
- velikost souborů



Operace se soubory

CREATE	vytvoření souboru (jméno)
DELETE	smazání souboru (jméno)
OPEN	otevření souboru (jméno)
CLOSE	zavření souboru (id)
READ	čtení ze souboru (id)
WRITE	zápis do souboru (id)
SEEK	posun aktuální pozice (id)



Proč potřebujeme adresáře?

- udržení organizační struktury souborů
 - ♦ dnes typicky hierarchický systém
 - ♦ počátečním elementem je kořenový adresář
- uchovávání atributů souboru

Adresář – zvláštní typ souboru

- vnitřní struktura definována OS
- speciální operace nad adresáři
 - ♦ hledání souboru, vypsání adresáře
 - ♦ přejmenování, vytvoření, smazání souboru



Cesta k souboru

- pojmenování souboru v hierarchickém uspořádání
- absolutní cesta
 - ♦ cesta v grafu od kořene k souboru
- relativní cesta
 - ♦ cesta z aktuálního adresáře k souboru
 - ♦ pojmenování pro aktuální (.) a rodičovský (..) adresář

Aktuální adresář

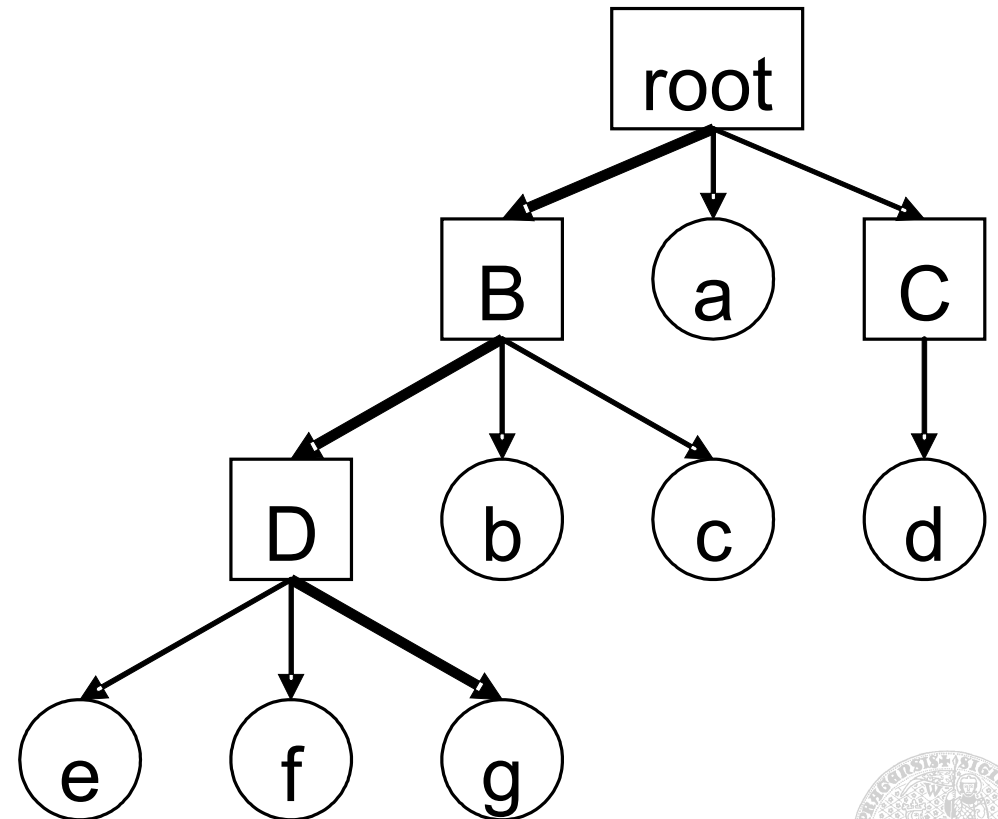
- vlastnost procesu
- jména souborů, která nezačínají kořenem, se hledají vzhledem k aktuálnímu adresáři



Hierarchická struktura

Strom

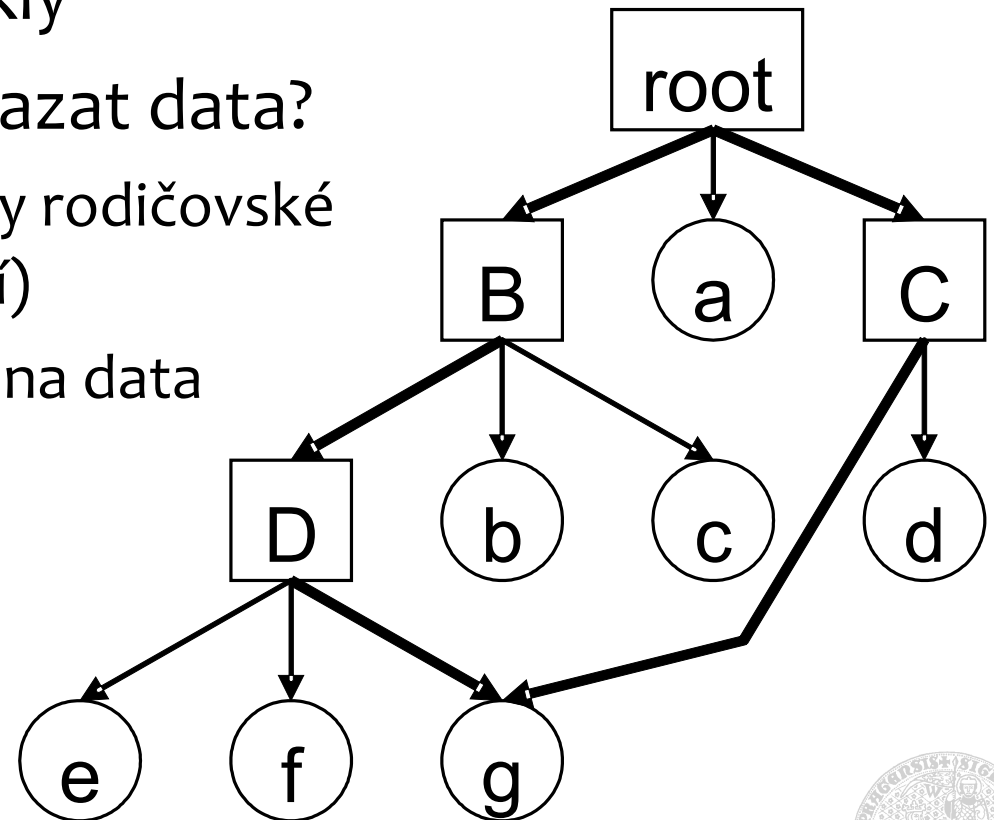
- jednoznačné pojmenování
 - ♦ /B/D/g



Hierarchická struktura

DAG – Directed Acyclic Graph

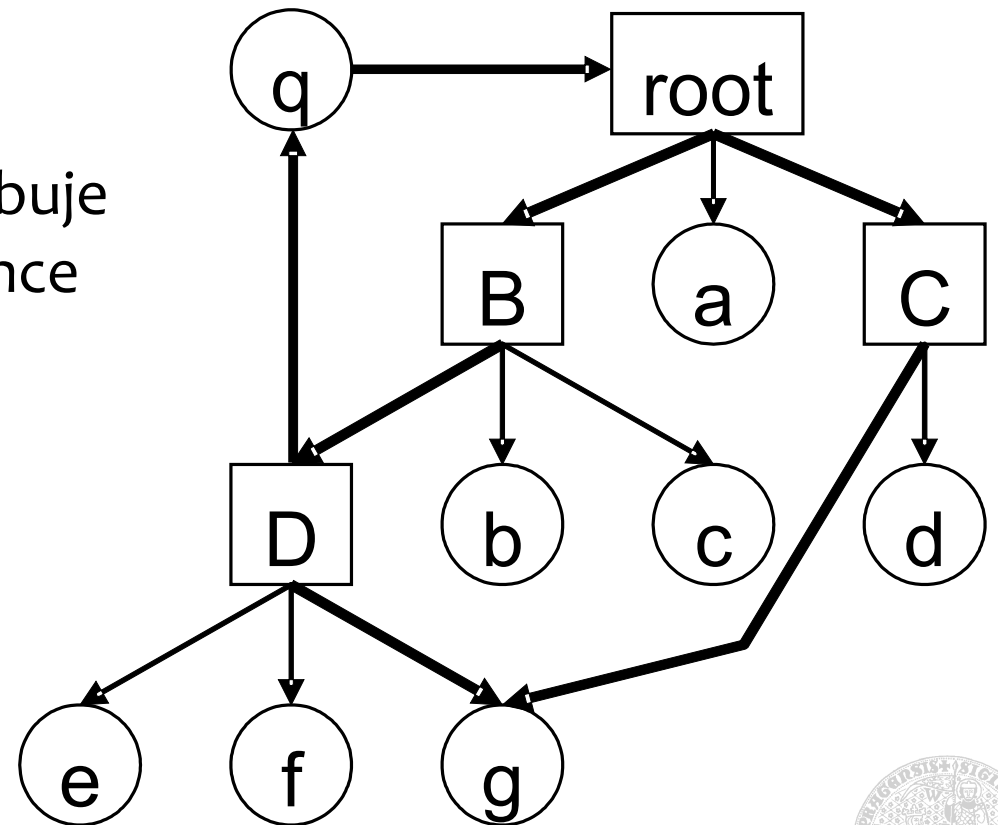
- víceznačné pojmenování
 - ♦ /B/D/g nebo /C/g
- hierarchie neobsahuje cykly
- mazání souboru – kdy smazat data?
 - ♦ nutno prozkoumat všechny rodičovské uzly (nedostatek informací)
 - ♦ použít reference counting na data souboru (obvyklé)



Hierarchická struktura

Obecný graf

- víceznačné pojmenování
 - ♦ /B/D/g nebo /B/D/q/C/g
- cykly vytváří problémy
 - ♦ při prohledávání
 - ♦ při mazání souborů (potřebuje garbage collection, reference counting nestačí)



Odkazy na soubory (links)

Hard link

- na jedna data souboru (nikoliv adresáře) se odkazuje z různých položek v (různých) adresářích
 - ♦ data mají více jmen
- souborový systém má DAG strukturu

Soft link (symbolický odkaz, symlink)

- speciální soubor obsahující jméno souboru/adresáře
 - ♦ při otevření se otevře soubor určený odkazem, při mazání se maže odkaz
- souborový systém má logicky strukturu obecného grafu bez problémů s mazáním



Implementace souborových systémů

Správa souborů

- kde na disku jsou umístěna data uložená v souboru

Správa adresářů

- mapování jména na jeho binární identifikaci
- uložení atributů

Správa volného místa

- které bloky jsou ještě volné



Implementace souborových systémů

Ukládání souboru na disk

- disky jsou organizovány po sektorech
- soubory se ukládají na disk po blocích
- velké bloky
 - ♦ rychlejší práce s diskem
 - ♦ nebezpečí velké vnitřní fragmentace
 - průměrná velikost souboru ~1500B
- malé bloky
 - ♦ pomalejší práce s diskem
 - ♦ větší režie na informaci o volných blocích
 - ♦ větší režie na informaci o umístění dat souboru



Uložení souborů na disku

Souvislá alokace

- souvislý sled bloků (run, extent)
- informace o uložení souboru sestává pouze z čísla prvního bloku
- lepší práce s diskem
- problém při hledání volného místa
- problém při zvětšování souborů

Run list

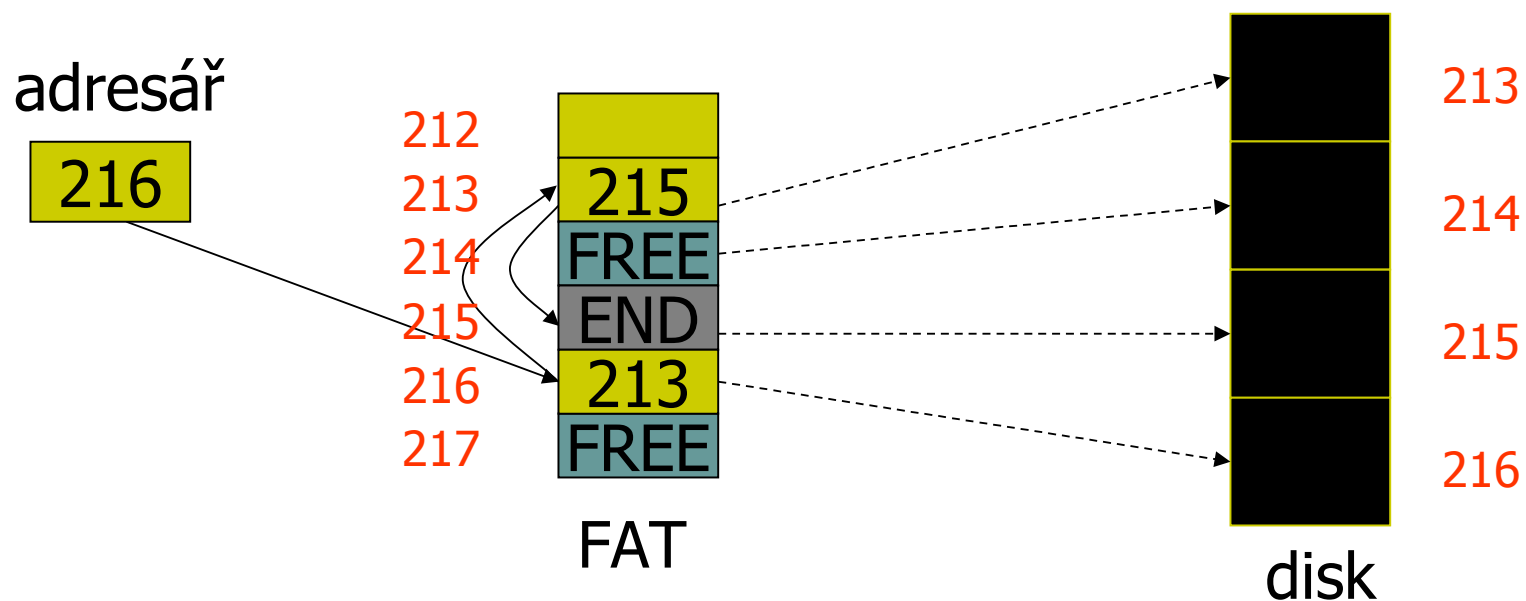
- seznam runů, ze kterých se skládá soubor
- umožňuje fragmentovat soubor – odpadají problémy s hledáním volného místa a při zvětšování souborů



Uložení souborů na disku

Spojovaná alokace

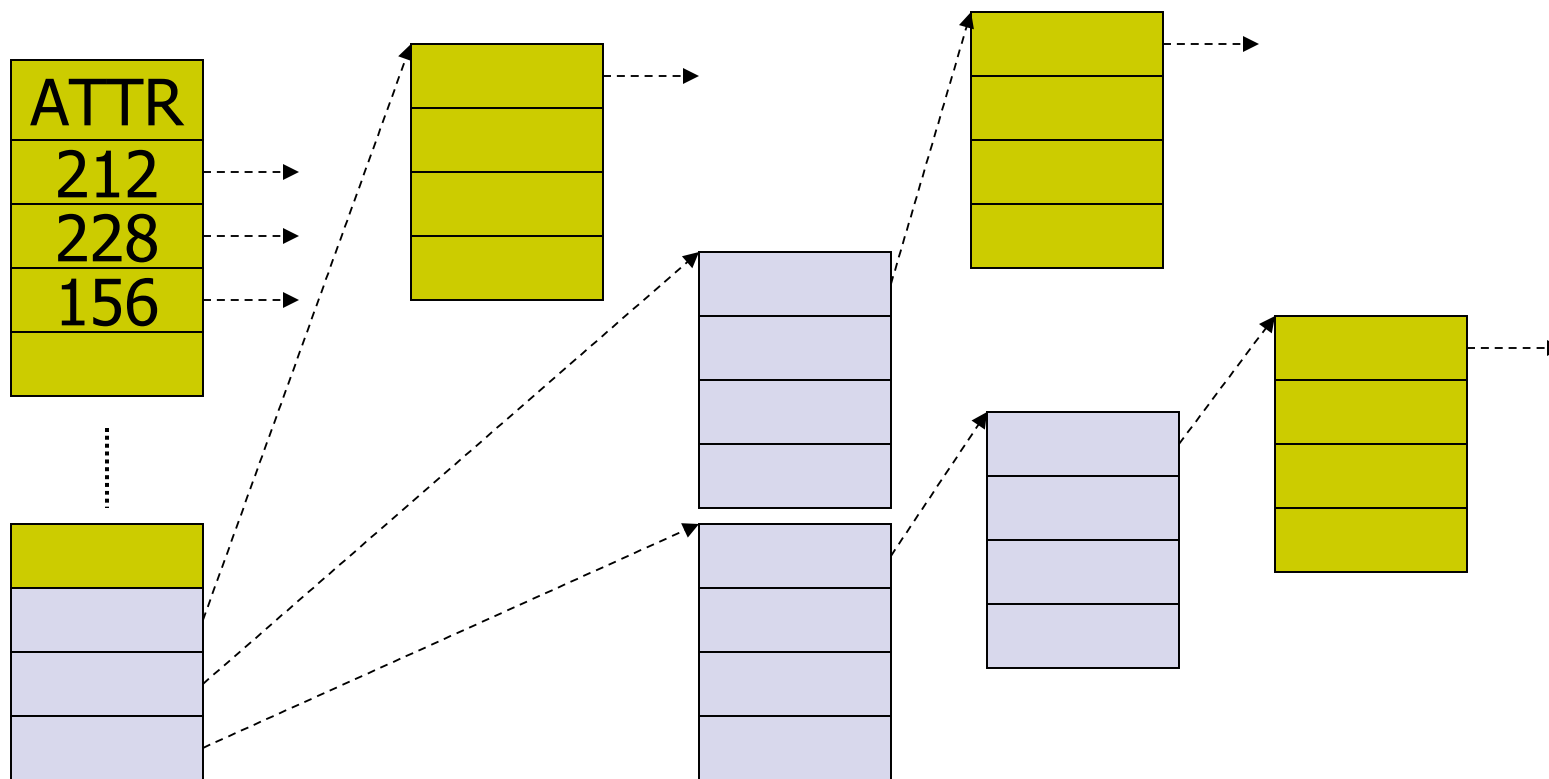
- pospojování bloků použitých pro soubor
- modifikace FAT – přemístění spojového seznamu do speciální oblasti disku



Uložení souborů na disku

Indexová alokace

- UNIX a i-node



Implementace adresářů

Záznamy pevné velikosti

- FAT

Spojový seznam

- delší jména
- pomalé hledání

B-stromy

- rychlé hledání
- Win NT



Volné místo na disku

Obdoba správy paměti

- souvislý adresový prostor rozdělený do bloků
- struktury nutno přizpůsobit práci s diskem
 - ♦ co to znamená?

Bitmapa

- NTFS, HPFS, NetWare

Spojový seznam volných bloků

- UNIX, Ext (ne Ext2)



Příklady souborových systémů

NTFS

NTFS (Windows NT, ...)

Základní charakteristika

- v podstatě neomezená velikost systému (16 exaB)
- v podstatě neomezený počet souborů
- jména v UCS-2 (podmnožina UTF-16), max. 255 znaků
- libovolné atributy – včetně přístupových práv
- vícenásobné streamy
- sparse files – díry ve streamech
- šifrování - od NTFS 5
- komprese
- automatická fault tolerance – žurnálování, transakce
- hard linky, symbolické linky – od NTFS 5
- adresáře jsou soubory s B-stromem s odkazy na soubory



Základní struktura

Oblast MFT (Master File Table)

- řídicí struktury obsaženy ve speciálním souboru MFT
 - ♦ vyhrazeno cca 12% kapacity, MFT rozdělena na záznamy (1 KB)
- prvních 16 souborů je speciálních – metasoubory
 - ♦ \$MFT – sama MFT
 - ♦ \$MFTmirr – kopie prvních 16 záznamů MFT
 - ♦ \$LogFile – žurnál
 - ♦ \$AttrDef – seznam std. atributů souboru na systému
 - ♦ \$Volume – informace o systému
 - ♦ \$. – kořenový adresář
 - ♦ \$Bitmap – bitmapa volného místa

Oblast dat



Pro každý soubor záznam v MFT

- počet odkazů
- příznaky
- seznam atributů (včetně streamů) – jméno, typ, data
- „Data“ souboru jsou taky stream s názvem :\$data
- přístup ke streamu je soubor:stream
- data rezidentní – přímo v MFT
- data nerezidentní – run list
- pokud pro popis dat nebo seznamu atributů nestačí základní záznam, tak se připojí další záznamy



Run list

VCN – virtual cluster number

- od začátku souboru

LCN – logical cluster number

- od začátku systému

Jeden run

- velikost runu, VCN, LCN, odkaz na další run



Příklady souborových systémů

Ext2/Ext3

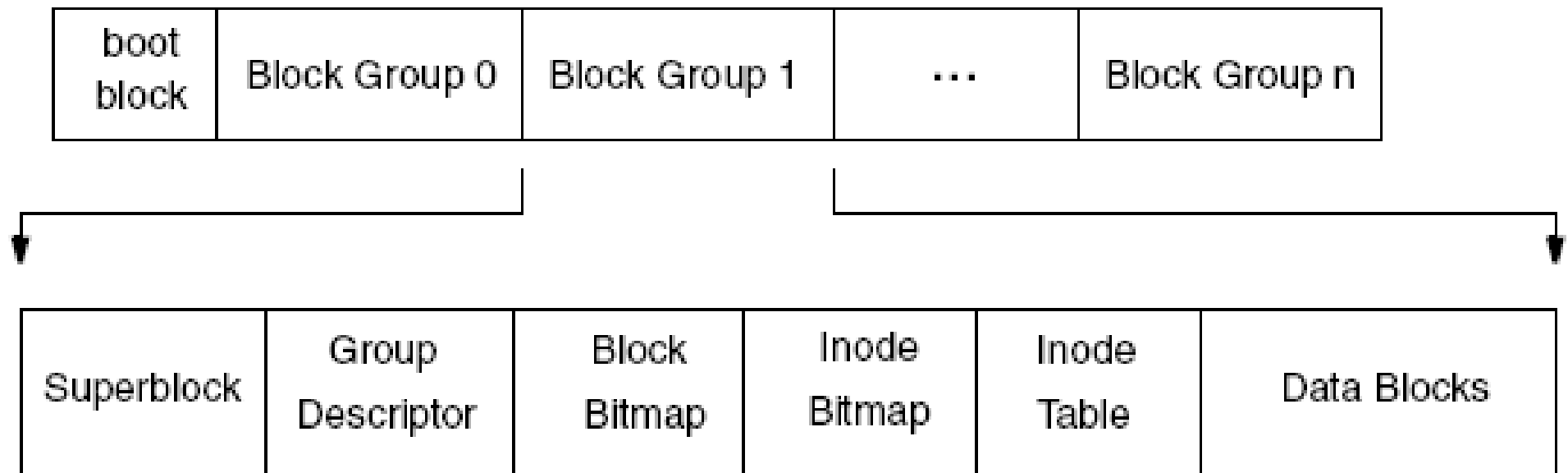
Ext2/Ext3 (Linux)

Základní charakteristika

- max. velikost souborového systému 16TiB
- jména max. 255 znaků
- rezervace volného místa pro vybraného uživatele
 - ♦ typicky root, ale může být zvolen jiný
- hard linky, symbolické linky
- prealokace po 8 souvislých blocích při zápisu, zbytek vrácen při zavření
- nezávislé žurnálování – EXT3
 - ♦ Souborový systém rozumí transakci, ale nedělá žurnálování
 - ♦ několik transakcí je cachováno, a pak uloženo najednou do žurnálu jako složená transakce



Struktura na disku



Soubory

Inode

- reprezentuje (pouze) data souboru
- jméno souboru s číslem inode uloženo v adresáři

Několik pevných atributů

- UID, GID, velikost, různé časy, počet odkazů
- 12 přímých odkazů na bloky
- Jeden nepřímý, jeden dvojitě- a jeden trojitě-nepřímý odkaz

Rezervovaná inode (první volný je 11)

- EXT2_ROOT_INO – kořenový adresář
- EXT2_BAD_INO – seznam špatných bloků

