

Organizace diskového prostoru v systémech souborů

Z FITwiki

Obsah

- 1 Základní pojmy
- 2 Struktura souborů
 - 2.1 a) Logická struktura souboru
 - 2.2 b) Vnitřní struktura souboru
 - 2.3 Alokace alokačních bloků
 - 2.3.1 Souvislá alokace
 - 2.3.2 Lineární seznam
 - 2.3.3 Index-sekvenční organizace - pevné alokační bloky
 - 2.3.4 Index-sekvenční organizace - proměnné alokační bloky
- 3 Alokace diskového prostoru
- 4 Příklady implementace
 - 4.1 Unix file system (UFS)
 - 4.2 BSD Fast file system
 - 4.3 EXT2/EXT3
 - 4.4 Log structured file system (LFS)
 - 4.5 NTFS
 - 4.6 SSD

Základní pojmy

Soubor

je abstraktní jednotka obsahující data, která má jednoznačnou identifikaci.

Fyzická abstrakce souboru

I/O po sektorech disku, adresa fyzického sektoru, bez práv a ochran

Logická abstrakce souboru

I/O po bajtech (nebo záznamech), identifikace souboru, práva souboru

Systém souborů poskytuje

- strukturu souborů
- alokaci diskového prostoru
- souborové operace
- ochranu

Struktura souborů

a) Logická struktura souboru

je na úrovni rozhraní jádra OS

- **Bez struktury** - pouhé pole bajtů (Unix)

- **Logické záznamy s pevnou délkou** - rychlý náhodný přístup, lehké změny, velký prostor nevyužit pokud nejsou záznamy plné
- **Logické záznamy s proměnnou délkou** - pomalý přístup, špatná aktualizace, plné využití
- **Index-sekvenční soubory** - záznamy proměnné délky a index podle klíče (B-strom)

b) Vnitřní struktura souboru

je na úrovni I/O

- Soubor je pole alokačních bloků.
- Alokační blok je několik souvislých sektorů.
- Je potřeba překlad z logického adresování na adresy bloků.
- Velikost alokačního bloku ovlivňuje rychlost a využití místa.
- Úzce souvisí s alokací diskového souboru (viz dále)

Alokace alokačních bloků

- náhodný/sekvenční soubor
- malé soubory - hodně souborů, nesmí zabírat moc místa
- velké soubory - pokud možno sekvenčně kvůli rychlosti přístupu

Souvislá alokace

- celý soubor vcelku
- [+] rychlý přístup k celému souboru i při náhodném přístupu
- [-] nutná prealokace místa
- [-] externí fragmentace
- [-] problém je vkládání dat jinam než na konec

Lineární seznam

- jednosměrně nebo obousměrně vázané seznamy bloků
- [+] jednoduché vkládání, přidávání, rušení dat
- [+] nulová externí fragmentace
- [-] náhodný přístup pomalý (nutné sekvenční procházení seznamu)
- [-] slabá lokalita, náhodné adresy sousedních logických bloků

Index-sekvenční organizace - pevné alokační bloky

- Prostor alokovan po blocích pevné velikosti
- Index obsahuje seznam alokačních bloků
- [+] jednoduchá implementace
- [+] snadné zvětšování/zmenšování souborů
- [-] pro velké soubory pomalý průchod indexy

Problém velikosti tabulky

- **Globální tabulka (FAT)**
- pevně daný počet alokačních bloků
- **Víceúrovňová tabulka (UFS)**
 - je rychlejší, ale nevýhodná pro malé soubory

Index-sekvenční organizace - proměnné alokační bloky

- index alokačních bloků proměnné délky
- problém alokace bloků proměnné velikosti

- bloky mají velikost 2^n nejmenšího alokačního bloku

Alokace diskového prostoru

v OS musí nejen volit umístění a velikost jednotek, ale také mít přehled o volném místě a rozložení souborů.

Velikost alokačního bloku

- menší bloky - pomalejší, lepší využití prostoru
- větší bloky - rychlejší, horší využití prostoru

Vyhledání volného místa

- souvislá alokace nebo alokace pevných bloků eviduje volné bloky
- alokace proměnné délky musí evidovat volné úseky a spojovat je nebo dělit
- pro optimalizaci diskových operací by bloky měly být co nejbližší u sebe

Okamžik alokace diskového prostoru pro soubor

- **prealokace** omezuje fragmentaci souboru, ale nešetří místo
- **dynamická alokace** při zápisu může vyvolat konflikt s jiným souborem
- **odložená alokace** čeká až na uložení stránky paměti

Příklady implementace

Unix file system (UFS)

- Alokační blok: 1 sektor
- Volné bloky: index-sekvenční
- Alokační blok: první volný blok bez optimalizace
- Informace o souborech: i-node na začátku disku
- Víceúrovňová tabulka i-node
 - 10 prvních deset bloků přímo
 - 11. alokační blok - adresy alokačních bloků
 - 12. alokační blok - adresy alokačních bloků s adresami alokačních bloků
 - 13. alokační blok - adresy bloků s adresami bloků s adresami bloků

BSD Fast file system

- založen na UFS
- Alokační blok: větší (4-16 kB)
- Volné bloky: bitová mapa
- Alokační blok: optimalizovaná (nejbližší volný blok, ...)

EXT2/EXT3

- zjednodušená BSD FFS
- Alokační blok: menší (4kB)
- Volné bloky: bitová mapa
- Alokační blok: bez optimalizací (pouze prealokace při zápisu)
- ext3 doplňuje žurnál

Log structured file system (LFS)

- každá zápis dat se zapíše do logu
- data se zapisují po velkých blocích na konec
- data se čtou z logu
- při zaplnění disku spuštěn cleaner (snaže staré duplicitní bloky)
- odolnost proti výpadku (data jsou v logu vícekrát, ztrácí se jen poslední změny)

NTFS

- Všechny soubory a adresáře jsou popsány v MFT (Master File Table, obdoba tabulky inodů) na začátku disku
- Položka MFT (1KB) obsahuje hlavičku (42 byte) a seznam atributů pro daný soubor (atribut Attribute List pokud se seznam atributů nevejde)

SSD

- náhodné čtení stejně rychlé jako sekvenční (není třeba defragmentace)
- zápis obvykle řádově pomalejší
- problém velikosti stránky (min. blok zápisu) – 4KB (32nm), 8KB (25nm) - nesnáší zápis malých bloků (sektory, write amplification)
- blok pro mazání značně větší, musí se mazat najednou celý (~256 KB, cca. 2 ms)
- počet zápisů do stejné stránky omezen (10k/5k/3k)
- rozpoznání uvolněných bloků (operace TRIM)
- vadí časté zápisy do stejného sektoru (např. aktualizace timestampu posledního přístupu v i-nodech nebo změnový žurnál)

Citováno z „[http://wiki.fituska.eu/index.php?](http://wiki.fituska.eu/index.php?title=Organizace_diskov%C3%A9ho_prostoru_v_syst%C3%A9mech_soubor%C5%AF&oldid=10925)

title=Organizace_diskov%C3%A9ho_prostoru_v_syst%C3%A9mech_soubor%C5%AF&oldid=10925“

Kategorie: Státnice 2011 | Pokročilé operační systémy

-
- Stránka byla naposledy editována 10. 6. 2013 v 10:50.