

# Virtualizace paměti, stránkovací a nahrazovací algoritmy

Z FITwiki

## Obsah

- 1 Organizace paměti
  - 1.1 Organizace LAP
    - 1.1.1 Jeden úsek paměti
    - 1.1.2 Společný adresový prostor
    - 1.1.3 Oddělené adresové prostory
  - 1.2 Mapování LAP na FAP
    - 1.2.1 Úseky pevné velikosti
    - 1.2.2 Úseky proměnné velikosti
    - 1.2.3 Segmentace
    - 1.2.4 Stránkování
    - 1.2.5 Segmentace se stránkováním
- 2 Stránkování
  - 2.1 Virtualizace a výpadek stránek
    - 2.1.1 Stránkovací algoritmus
  - 2.2 Nahrazovací algoritmy
    - 2.2.1 Nahrazovací algoritmy s pevným počtem rámců
    - 2.2.2 Nahrazovací algoritmy s proměnným počtem rámců

## Organizace paměti

Fyzický adresový prostor (FAP)

adresace paměti přímým přístupem, pohled na paměť z pohledu procesoru

Logický adresový prostor (LAP)

paměť z pohledu procesu, která se liší od fyzického přístupu k paměti

Virtualizace

při běhu procesu nemusí být celý obsah adresového prostoru trvale v paměti (některé úseky logické paměti mohou být např. swapovány na disku)

## Organizace LAP

### Jeden úsek paměti

- LAP=FAP
- jen jeden program v paměti
- OS má přidělenou část zbytek jednomu programu (MS-DOS)
- Monoprogramování bez ochrany paměti

### Společný adresový prostor

- všechny programy mají společný adresový prostor mapovaný na fyzickou paměť, pouze jsou zavedeny na jiná místa.
- je potřeba dynamická relokační (při zavedení programu se adresy do paměti změní podle místa zavedení)
- ale jednoduchá správa a přidělování paměti.
- Ochrana paměti není implicitní (programy mohou i do paměti jiných programů), lze řešit:

- mezní registry
- chráněný režim OS (programy mají ale omezený LAP).

## Oddělené adresové prostory

- každý program má k dispozici celý LAP, každý LAP mapován někam do FAP
- ochrana oddělením při mapování
- ale mapování složitější (podpora hardware - převod adres v každé instrukci)
- Adresový prostor jádra
  - a) oddělený LAP jádra (volání jádra musí přepočítat adresy),
  - b) sdílený s procesem volajícím jádro (horní část LAP rezervovaná pro jádro, v uživ režimu nepřístupná)

## Mapování LAP na FAP

### Úseky pevné velikosti

- FAP dělen na úseky pevné velikosti
- Programy pevnou velikost
- Programy jsou mapovány do vhodných volných úseků
  - Fronty procesů čekajících na úsek
  - Společné fronta - přidělen nejmenší postačující volný úsek

#### Interní fragmentace

nevyužitá část přiděleného úseku)

#### Odkládání (swapping)

pozastavený proces může být dočasně odložen do odkládacího prostoru aby se úsek uvolnil

### Úseky proměnné velikosti

- mění velikost úseků dynamicky dle požadavků
  - (spojování sousedních volných nebo dělení při zavedení programu)

#### Externí fragmentace

volná paměť není souvislá

#### Strategie přidělování úseků

- First fit - první dostačující
- Next fit - první dostačující za místem posledního přidělení
- Best fit - nejmenší dostačující
- Worst fit - největší volný (menší ext. fragmentace, ale fragmentuje největší úseky)

#### Alokace úseků o velikosti $2^n$

- používá malloc()
- přiděluje nejmenší postačující úsek o velikosti  $2^n$
- seznamy volných úseků jednotlivých velikostí - přiděluje se první na seznamu
- pokud není dostupný volný úsek alokuje se blok K nových úseků

#### Buddy systém

- Linux - systémová paměť
- alokace úseků o velikosti  $2^n$  + spojování úseků
- pokud není volný úsek - větší úsek se rozdělí na poloviny
- při uvolnění se sousední bloky spojují

## Segmentace

- LAP rozdělen do segmentů - úseků proměnné velikosti: kódový, datový a zásobník

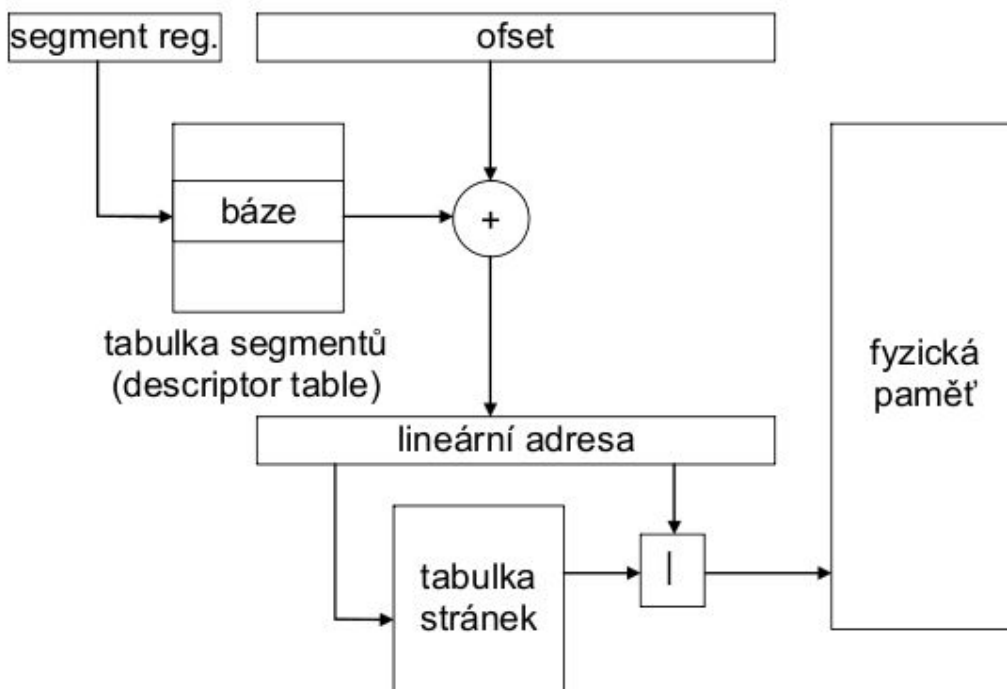
- logická adresa se skládá ze jména segmentu a posunu (offsetu) v něm
- Tabulka segmentů obsahuje pro každý segment bázi a limit (kde začíná a jak je velký) (tabulka pro každý proces nebo globální)
- umožňuje ochranu segmentů (zabraňuje programu zapisovat do kódového segmentu)
- namísto 1 spojitého bloku paměti jsou programu přidělovány jednotlivé segmenty - zmírňuje externí fragmentaci, ta ale přetrvává

## Stránkování

- dělí FAP na **rámce** stejné velikosti
- dělí LAP na **stránky** stejné velikosti
- velikost rámce = velikost stránky (obvykle 1kB až 16kB)
- Tabulka stránek pak určuje mapování stránek na rámce
- logická adresa - číslo stránky a offset
- Zamezuje externí fragmentaci (v LAP), interní fragmentace max o velikosti velikost stránky/2
- Ochrana musí být na úrovni stránek

## Segmentace se stránkováním

- segmenty jsou stránkovány



## Stránkování

### Tabulka stránek

- zobrazuje LAP na FAP
- pro každé číslo stránky obsahuje číslo rámce ve kterém je stránka umístěna

### Obsah tabulky stránek (i386)

- číslo rámce,
- flagy zápisu, systémové stránky, označení **dirty** (stránka byla modifikována) a přítomnosti v paměti, ...

### Inverzní tabulka stránek

- indexuje podle rámců ne podle stránek
- mnohem jednodušší tabulka a malá režie, ale hledání je komplikované a sdílení paměti ještě více
- PowerPC, HP PA RISC

## Rychlost stránkování

- překlad LAP a FAP se musí uskutečnit při každé instrukci adresující paměť (5-10ns asi 10% z času přístupu do paměti)

## Translation Look-Aside Buffer (TLB)

- je vyrovnávací paměť v HW pro překlad adres
- poslední/nejběžnější překlady adres jsou uloženy v rychlé (SRAM) asociativní paměti (přístup je zredukován z 50ns na 5ns)
- Čím větší úspěšnost TLB, tím rychlejší průměrný přístup.
- Při přepnutí kontextu se TLB vyprázdní (u SPARC ne, přidává položkám i číslo procesu)

## Velikost tabulky stránek

- pro 32bit adresu 1MB tabulka stránek

## Víceúrovňová organizace stránek

- adresa stránky se dělí na indexy jednotlivých tabulek, položky tabulek pak vybírají bázi následující tabulky
- (tj. *tabulka tabulek stránek*)
- Intel - 2 úrovně (Page Directory a Page Table)
- AMD64 - 4 úrovně

## Virtualizace a výpadek stránek

### Výpadek stránky

pokud požadovaná stránka není v paměti (díky virtualizaci)

- výpadek stránky výrazně zpomaluje -> je nutné minimalizovat počet výpadků

### Zpracování výpadku stránky

1. Výběr volného rámce
2. Pokud není žádný rámec volný, výběr stránky, která bude odstraněna – **nahrazovací algoritmy**
3. Zavedení požadované stránky a nastavení tabulky stránek
4. Restart instrukce, která způsobila výpadek

## Stránkovací algoritmus

udává, kdy a kolik stránek se zavede z disku do paměti, které rámce se jim přiřadí a případně, které stránky mají být z paměti odstraněny.

Snaha minimalizovat výpadky (nahrazením nepoužívaných apod.).

### 1) Výběr zaváděných stránek

- Kdy:
  - prefetchingem (dopředu)
  - zavádění na žádost (při výpadku)
- Počet zaváděných stránek:
  - celý LAP (neefektivní)
  - jediná stránka (neefektivní)
  - stránky a okolí (předvídá přístup na sousední stránky)

### 2) Umísťování stránek

- nemá vliv na výpadky, ale na rychlost odkládání (souvislý kus se odloží rychleji).

### 3) Nahrazovací algoritmy

- S pevným počtem stránek v paměti
  - OPT - Optimální - hypotetický, odstraňuje stránku která bude nejdéle nepoužita (čte budoucnost)

- LRU - Last Recently Used - odstraňuje nejdéle nepoužitou stránku
- NRU - Not Recently Used - odstraňuje v poslední době nepoužitou stránku (1-bitová aproximace LRU)
- FIFO - odstraňuje nedříve zavedenou stránku
- LIFO - odstraňuje nejpozději zavedenou stránku
- S proměnným počtem stránek v paměti
  - VMIN - Optimální - hypotetický, v paměti se drží stránky které budou použité v zadaném časovém intervalu v budoucnosti
  - WS - Working Set - v paměti se drží stránky použité v zadaném časovém intervalu - pracovní množina stránek (nepraktické - nutno často aktualizovat)
  - Page Fault Frequency
    - pokud stránky vypadávají častěji než je velikost pracovní množiny, pracovní množinu zvětší
    - pokud je interval mezi výpadky větší, jsou odebrány všechny stránky nepoužité od posledního výpadku - zmenšování pracovní množiny

Swapping = Odkládání stránek

je umísťování vyřazených stránek na disk do swapu (swap se nemusí použít, ale pak nelze stránky vyřadit a dojde paměť).

Thrashing

je stav, kdy počet výpadků překračuje přípustnou mez, většinu výkonu spotřebuje režie stránek (nejen algoritmus, ale i I/O swapu).

## Nahrazovací algoritmy

Nahrazovací algoritmus

určuje, které stránky vyřadit z paměti, snaha o minimalizaci následných výpadků. Problém je, že nezná následující sled stránek, takže pouze odhaduje z minulosti.

Klasifikace

- Dle rozsahu:
  - **Lokální** - vyřazuje pouze stránky procesu, který nyní chce stránku zavést
  - **Globální** - nerozlišuje stránky podle procesu
- Podle počtu rámců:
  - **Pevný počet rámců** - počet stránek v paměti zůstává stejný
  - **Proměnný počet rámců** - počet stránek v paměti se mění

Princip lokality

- **Prostorová lokality** - je vysoká pravděpodobnost, že následující adresa bude směřovat do stejné nebo sousední stránky (sekvenční kód, struktury a pole, proměnné jsou u sebe).
- **Časová lokality** - při běhu programu se některé proměnné používají opakovaně (cykly, funkční proměnné)
- Neplatí-li princip lokality (optimalizace kódu a dat), sebelepší používaný algoritmus nebude fungovat.

## Nahrazovací algoritmy s pevným počtem rámců

Optimální algoritmus

- Vybírá stránku, která bude nejdéle nepoužívána.
- Vyžaduje znalost budoucnosti (není reálný).
- Minimální možný počet výpadků stránek.

LRU (Least Recently Used)

- Vybírá stránku, která byla nejdéle nepoužita.
- Aproximuje optimální algoritmus díky lokalitě odkazů
- Implementace
  - ukládání čas posledního přístupu
  - zásobník, při použití přesun nahoru, vyhazuje se nejspodnější.

NRU (Not Recently Used)

- Aproximuje LRU jedním bitem.
- Při zavedení a použití je bit nastaven.
- Bit je nulován po uplynutí intervalu.

#### LFU (Least Frequently Used)

- Odstraněna nejméně používaná stránka.
- Může ale nahradit i právě použitou stránku.
- V případě rovnosti použije LRU.

#### FIFO

- Odstraní stránku, která byla nejdéle v paměti.
- Mohla tam být proto, že je neustále používána.
- Dochází k Beladyho anomálii (počet výpadků může růst s počet uchovávaných stránek v paměti)

#### LIFO

- Odstraní stránku, která je v paměti nejkratší dobu.
- Dojde do stavu, kdy se pouze vyhazuje poslední stránka a ostatní zůstávají.

#### Clock

- Cyklický seznam stránek s jednobitovým příznakem použití
- Ukazatel na aktuální pozici.
- Při hledání se testuje příznak použití
  - Je-li nastaven, vynuluje se a posune ukazatel.
  - Pokud není nastaven, stránka se použije (a posune se ukazatel).

#### Second Chance (NRU)

- Založen na Clock, ale má kromě příznaku použití i příznak modifikace.
- Není-li bit užití nastaven a je nastaven zápisový, stránka se pouze zapíše na disk, bit se vynuluje a posune se ukazatel.

### Nahrazovací algoritmy s proměnným počtem rámců

#### VMIN

- Udržuje pouze stránky, které budou potřeba
- Optimální nerealizovatelný algoritmus (znalost budoucnosti)

#### WS (Working Set)

- V paměti je pouze pracovní množina stránek, tedy stránky, které byly použity v daném intervalu.
- Složitá implementace, při každém odkazu stránky (nejen výpadku) se musí aktualizovat.
- Princip klouzavého okna

#### Page Fault Frequency

- Volí velikost pracovní množiny podle frekvence výpadků.
- Při vysoké frekvenci výpadků je možné přidat další prvek množiny.
- Při nízké frekvenci se odeberou všechny stránky nepoužité od posledního výpadku.

Citováno z „[http://wiki.fituska.eu/index.php?](http://wiki.fituska.eu/index.php?title=Virtualizace_pam%C4%9Bti,_str%C3%A1nkovac%C3%AD_a_nahrazovac%C3%AD_algoritmy&oldid=13235)

[title=Virtualizace\\_pam%C4%9Bti,\\_str%C3%A1nkovac%C3%AD\\_a\\_nahrazovac%C3%AD\\_algoritmy&oldid=13235](http://wiki.fituska.eu/index.php?title=Virtualizace_pam%C4%9Bti,_str%C3%A1nkovac%C3%AD_a_nahrazovac%C3%AD_algoritmy&oldid=13235)“

Kategorie: Státnice 2011 | Pokročilé operační systémy

- 
- Stránka byla naposledy editována 1. 6. 2016 v 08:04.