

Operační systémy 2



Virtuální paměť

Petr Krajča

Katedra informatiky
Univerzita Palackého v Olomouci

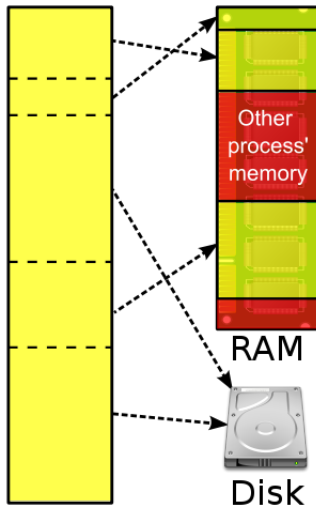
4. říjen, 2010

Motivace

- paměť RAM je relativně drahá \Rightarrow nemusí vždy dostačovat
- aktuálně používaná data (např. instrukce) musí být v RAM, nepoužívaná data nemusí (velké programy \Rightarrow nepoužívané funkce)
- je vhodné rozšířit primární paměť (RAM) o sekundární (např. HDD) 
- zvětšením dostupné paměti je možné zjednodušit vývoj aplikací (není potřeba se omezovat v množství použité paměti)
- sekundární paměť bývá řádově pomalejší
- k efektivní implementaci je potřeba spolupráce HW (MMU) a OS
- z pohledu aplikace musí být přístup k paměti transparentní
- virtuální paměť (VM) je součástí soudobých OS (swapování)
 - Windows NT – stránkový soubor (pagefile.sys)
 - Linux – swap partition (ale může být i soubor)
- bezpečnost dat v sekundární paměti? (např. po vypnutí počítače) 

Virtual Memory
(Per Process)

Physical
Memory



Inicializace procesu a jeho běh

- můžeme načíst celý proces do primární paměti (může být neefektivní)
- demand paging (stránkování na žádost)
- do paměti se načtou jen data (stránky), která jsou potřeba (případně související) \Rightarrow sekvenční čtení; prefetch)
- systém si eviduje, které stránky jsou v paměti a které ne (HW, stránkovací tabulka)
- přístup na stránku, která není v primární paměti \Rightarrow přerušení – výpadek stránky (page fault)
- přerušení načte stránku do paměti, aktualizuje stránkovací tabulku
- je-li primární paměť plná, je potřeba nějakou stránku přesunout do sekundární paměti (odswapovat)
- pokud je odsouvaná stránka sdílená (např. CoW) je potřeba aktualizovat všechny tabulky, kde se vyskytuje
- potřeba efektivně převádět rámce na stránky

Vlastnosti stránek

Rezervovaná stránka

- existují v adresním prostoru, ale nezapisovalo se do ní
- každá stránka je nejdříve rezervovaná
- vhodné pro velká pole, ke kterým se přistupuje postupně
- zásobník

Komitovaná stránka (Committed)

- stránka má rámec v primární nebo sekundární paměti
- musí řešit jádro
- paměť je často současně komitovaná i rezervovaná

Další vlastnosti

- dirty bit – 0 pokud má stránka přesnou kopii v sekundární paměti; 1 nastaveno při změně (nutná podpora HW)
- present/absent bit – přítomnost stránky v paměti (HW, nutné k detekci výpadků stránek)
- mohou mít přístupová práva (NX bit)

Výměna stránek

- page fault
- pokud není stránka v primární paměti, načte stránku do ní
- není-li volný rámec v primární paměti, je potřeba odsunout nějakou stránku do sekundární paměti
 - získáme volný rámec v sekundární paměti (pokud není volný rámec v sekundární paměti, najde se takový, který má kopii v primární paměti, nastaví se dirty bit a daný rámec se použije)
 - vybere se „oběť“ – stránka v primární paměti, která bude uvolněna
 - pokud má stránka nastavený dirty bit, přepíše se obsah rámce do sekundární paměti
 - načte se do primární paměti stránka ze sekundární
- zopakuje instrukci, která vyvolala page fault
- některé stránky je možné zamknout, aby nebyly odswapovány (nutné pro jádro, rámce sdílené s HW)

Výběr oběti (1/3)

- hledáme stránku, která nebude v budoucnu použita (případně v co nejvzdálenější budoucnosti)

FIFO

- velice jednoduchý algoritmus
- stačí udržovat frontu stránek
- při načtení nové stránky je stránka zařazena na konec fronty
- pokud je potřeba uvolnit stránku bere se první z fronty
- nevýhoda – odstraní i často užívané stránky
- Beladyho anomálie – za určitých okolností může zvětšení paměti znamenat více výpadků stránek

Least Frequently Used (LFU)

- málo používané stránky \Rightarrow nebudou potřeba
- problém se stránkami, které byly nějaký čas intenzivně využívány (např. inicializace)

Most Frequently Used (MFU)

- právě načtené stránky mají malý počet přístupů

Least Recently Used (LRU)

- jako oběť je zvolena stránka, která nebyla nejdýl používána
- je potřeba evidovat, kdy bylo ke stránce naposledy přistoupeno
- řešení:
 - 1 počítadlo v procesoru, inkrementované při každém přístupu a ukládané do tabulky stránek
 - 2 „zásobník“ stránek – naposledy použitá stránka se přesune navrchol
- nutná podpora hardwaru

LRU (přibližná varianta)

- každá stránka má přístupový bit (*reference bit*) nastavený na 1 pokud se ke stránce přistupovalo
- na počátku se nastaví reference bit na 0
- v případě hledání oběti je možné určit, které stránky se nepoužívaly
- varianta
 - možné mít několik přístupových bitů
 - nastavuje se nejvyšší bit
 - jednou za čas se bity posunou doprava
 - přehled o používání stránky \Rightarrow bity jako neznaménkové číslo \Rightarrow nejmenší = oběť

Výběr oběti (3/3)

Algoritmus druhé šance

- založen na FIFO
- pokud má stránka ve frontě nastavený přístupový bit, je nastaven na nula a stránka zařazena nakonec fronty
- pokud nemá je vybrána jako oběť
- lze vylepšit uvážením ještě dirty bitu

Buffer volných rámců (optimalizace)

- proces si udržuje seznam volných rámců
- přesun oběti je možné udělat se zpožděním
- případně, pokud je počítač nevytížený, je možné ukládat stránky s dirty bitem na disk a připravit se na výpadek (nemusí být vždy dobré)



Minimální počet rámců

- každý proces potřebuje určité množství rámců (např. movsd potřebuje v extrémním případě 6 rámců)
- stránkovací tabulka(y) musí být opět v rámci
- přidělování rámců procesům
 - rovnoměrně
 - podle velikosti adresního prostoru
 - podle priority
 - v případě výpadků stránek podle priority (globální alokace rámců)
- pokud počet rámců klesne pod nutnou mez, je potřeba celý proces odsounout z paměti
- hrozí thrashing (proces začne odsouvat stránky z paměti, které právě potřebuje)

Thrashing

- systém je ve stavu, kdy odvádí spoustu práce, ale bez rozumného efektu
- modelová situace:
 - pokud poklesne vytížení procesoru, systém spustí další proces
 - pokud je použitý algoritmus s globální alokací rámců, může odebírat rámce ostatním procesům
 - ostatní procesy můžou tyto rámce požadovat a brát je ostatním procesům
 - čeká se na sekundární paměť \Rightarrow sníží se využití CPU
 - procesor se pokusí spustit další proces, etc.
- viz Keprt p. 105
- lokální alokace rámců může thrashing omezit
- ideální je, aby měl proces tolik rámců kolik potřebuje

Řešení thrashingu

Pracovní množina rámců (working-set)

- vychází z principu lokality
- má-li proces tolik rámců kolik jich v nedávné době (lokalitě) použil
⇒ OK
- má-li jich více ⇒ neefektivní využití
- má-li jich méně ⇒ hrozí thrashing a je lepší celý proces odsunout z primární paměti
- hrozí hladovění velkých procesů
- náročný výpočet (např. podobný algoritmu druhé šance)

Frekvence výpadků stránek

- sledujeme, jak často dochází u procesu k výpadku stránky
- je potřeba stanovit horní a dolní mez
- pokud proces je mimo tyto meze ⇒ přidat/ubrat rámce

Velikost stránek

- stránky mají velikost 2^n , typicky v intervalu $2^{12} - 2^{22}$, i.e., 4 KB – 4 MB
- závisí na HW architektuře (může být i víc nebo méně)
- z pohledu fragmentace je vhodnější mít stránky menší
- více menších stránek zabírá místo v TLB \Rightarrow časté cache miss
- při přesunu do swapu může být velká stránka výhodnější (přístupová doba)
- některé systémy umožňují používat různé velikosti
- Windows NT \leq 5.1 & Solaris velké stránky pro jádro malé pro uživatelský prostor
- Windows Vista a novější – large pages
- Linux (hugetblfs)

Převrácená tabulka stránek (Inverted Page Table)

- někdy je potřeba namapovat rámce zpět na virtuální stránky
- prohledávat tabulky stránek je neefektivní (miliony záznamů)
- slouží k tomu převrácená tabulka stránek (tabulka pevné velikosti podle počtu rámců)
- k vyhledávání slouží pomocná hash tabulka (Hash Anchor Table)
- mapuje se adresa (případně PID)

Mapování souborů a I/O do paměti

Soubory

- operace `open`, `read`, `write` mohou být pomalé (systémové volání)
- mechanismus, který je použitý pro práci se sekundární pamětí, lze použít pro práci se soubory
- soubor se načítá do paměti po blocích velikosti stránky podle jednotlivých přístupů (demand paging)
- k souboru se přistupuje pomocí operací s pamětí (přiřazení, `memcpy`, ...)
- data se nemusí zapisovat okamžitě (ale až s odmapováním stránky/souboru)
- více procesů může sdílet jeden soubor \Rightarrow sdílená paměť (WinNT)
- možnost použít copy-on-write

I/O

- lze namapovat zařízení do paměti (specifické oblasti) a přistupovat k němu jako k paměti
- pohodlný přístup, rychlý přístup
- např. grafické karty

Poznámky na závěr

- jádro může požadovat souvislý blok rámců
- stránkovací tabulky jsou opět jen stránky \implies mohou být odsunuty?
- spolupráce s cachí
- základní algoritmy
- realně se implementují složitější metody (heuristiky)