

Správa paměti a virtuální paměť

13. ledna 2009

10:44

Algoritmy výměny stránek

- Chceme paměť, ale není prostor
- Když máme v cache množiny položek, a máme všechny položky plné a potřebujeme zapsat nová data, musíme z té množiny vybrat nějakou položku, kterou zahodíme - stejný princip jako u stránkování

Working set

- Množina právě používaných stránek
- Nějaké pracovní jádro, které je potřeba, aby program mohl běžet, odráží lokalitu při vykonávání programu
- Když máme více programů v paměti, snažíme se do paměti alespoň do paměti nacpat alespoň working sety
- Když by toto nebylo, tak bysme skoro pořád jen četli z disku

Algoritmy:

- FIFO
 - Nejjednodušší a **nejhloupější** strategie
 - Nahrazujeme rámce, který byl v paměti nejdéle, ten uvolníme a dáme ho k dispozici novému požadavku, který se objevil
 - Úplně ignoruje chování programu
 - Jedna, která vykazuje **Beladyovu anomálii** - zvýšení počtu rámců může vést ke zvýšení počtu výpadků
 - Pro nějaký proces máme vyhrazeno, že pro něj máme určitý počet vyhrazených rámců, a nahrazování budeme hledat právě v tomhle setu.. A anomálie znamená, že i když dáme větší počet rámců, tak se v průměru nemění počet pagefaultů
- Optimální stránka
 - V principu se snažíme vyhodit stránku, na kterou přistoupíme za nejdelsí dobu
 - Ale můžeme pracovat a zpětně říct, která to byla a použít na porovnání
 - Nelze implementovat online, ale lze simulovat ^
 - V principu jde o to mít v paměti vždy celý SW
- LRU - Least Recently Used
 - Dlouho nepoužívané stránky nebudou potřeba
 - Funguje docela dobře, ale realizace v SW není zrovna efektivní - při každém přístupu musíme aktualizovat informace o použití stránek
 - Vlastně bysme pak jen zapisovali uživatelnost
 - Realizace v HW je možná, ale obtížná/drahá - máme nějaký čítač, který CPU zpracovává; ve stránkovacích tabulkách je to problém (ale použitelné např. u webové proxy)
 - Problém aktualizace při každé změně - např spoják který by používané dával vždy nakonec, po čase by se nám na začátku hromadili málo používané
 - Nebo psát timestamp ke každému záznamu, ale lepší je spoják, je hned vidět kdo je obět
- NRU - Not recently Used
 - Stránka nebyla v blízké době použita
 - Každá položka má příznak Accessed - nastavuje hW při přístupu do paměti
 - OS bude periodicky procházet a bude nulovat příznak Accessed v seznamu stránek, tím říkáme, že uběhla perioda, kdy se stránka použila
 - Dirty = do stránky se zapisovalo, že jsem s ní něco prováděl; vlastnost HW
 - Když dojde k výpadku stránky tak vybíráme náhodně z těchto množin v tomto pořadí
 - Not accessed; not dirty - hned můžeme zapisovat
 - Not accessed; dirty - sice měli obsah, tak je zapíšem na disk, a zapíšem
 - Accessed, not dirty
 - Accessed; dirty - tyhle necháváme nejdéle, zde vypadá, že s nimi něco systém dělá
 - Pokud jen present, tak present nastavit na 0 - to vyvolá pagefault a zjistíme si, proč byl

- pagefault
 - Problém je u nulování příznaků - něco, co trvá dlouho; chceme je nulovat rozumně -> CLOCK
- CLOCK - Hodinový algoritmus
 - Modifikace NRU; "druhá šance"
 - Stránky jsou zařazeny v kruhovém seznamu
 - Funguje tak, že pokud nastane pagefault, takse podíváme, jesti ta stránaka, nak terou ukazujeme je volná, když je volná, máme oběť, pokud ne, posuneme se dále, dokud nenajdeme vhodnou oběť; pokud se otočím o celou obrátku dokola, tak vyhodím ten kde jsem začal (předtím jsem si ho v prvním kole hodil na accessed = 0) pokud se mezitím něco stalo (a se ten začáteční znovu hodí na accessed = 1)
 - Abychom nemuseli prohledávat celý spoják - museli bysme ho zamykat - a tím blokovat přístup
 - Zde nám stačí pointery které rotují
 - Také jde simulovat seznamem, kde bysme přehazovali prvky nakonec
 - Trošku vyšlo z FIFO
 - Pokud accessed = 0, oběť nalezena
 - Pokd accessed = 1, vynuluji a pokračuji dál v hledání
 - Dvě ručičky - pohybují se současně
 - Jednou koukáme na potencionální oběti, stejně jako v prvním případě
 - A druhou ručičkou vytváříme kandidáty (která jde napřed); jsme tím schopni, změnou úhlu, zpoždění, regulovat dobu, za kterou má ta stránka být znovu referencována - znovu použitá
 - Příznaky se ukládají do stránkovacích tabulek
 - Při procházení rámců se dívám do stránkovacích tabulek, kde jsou ty příznaky
- NFU - Not frequency used
 - SW řešení LRU
 - Čítač u stránek, periodicky přičítáme A, vybere se nejnižší
 - Problem - nezapomíná
- Trashing [!Slidy!]
- Segmentace [!Slidy!]
 - Bloky se popisují deskriptory
 - Dočíst!! Problémy segmentace