

Stránkování



Virtuální adresový prostor rozdělen na bloky stejné velikosti, které se nazývají **stránky** (pages). Fyzický adresový prostor je rozdělen na bloky stejné velikosti jako stránky, a ty se nazývají **rámce** (frames).

Adresa ve virtuálním adresovém prostoru je vyjádřena jako uspořádaná dvojice (p, d) , kde p je číslo stránky a d je posunutí od začátku stránky.

Mechanismus stránkování pomocí **stránkovacích tabulek** převede číslo stránky p na odpovídající číslo rámce p' , pokud takový převod existuje. Tento převod se nazývá **mapování**. Výsledná adresa do fyzického adresového prostoru je pak tvořena dvojicí (p', d) .

Pokud neexistuje mapování, došlo k **výpadku stránky** (page fault), který je hlášen OS. Ten pak má možnost mapování změnit.

Stránkovací tabulky

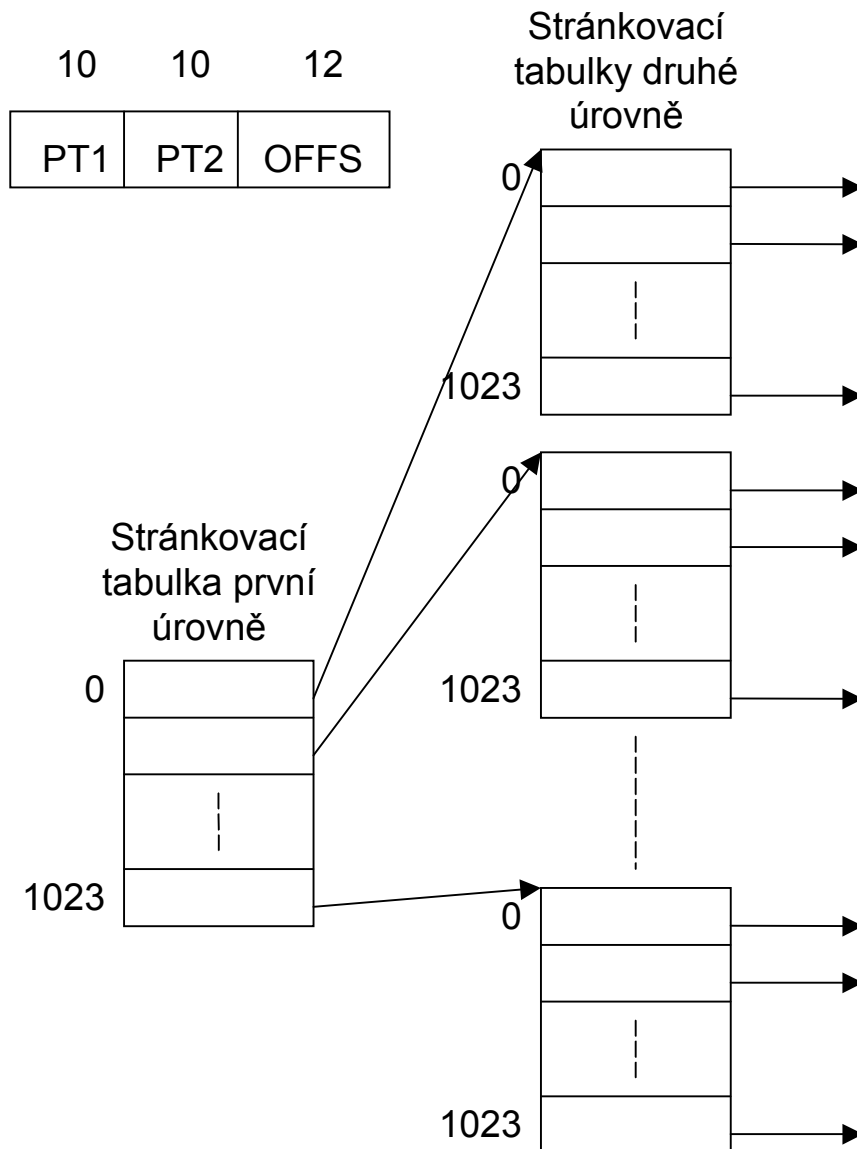
Zajišťují informace nutné pro mapování včetně příznaku existence tohoto.

Stránkovací tabulky přináší dva problémy:

1. Velikost stránkovacích tabulek (32-bitový virtuální adresový prostor obsáhne 4GB, při velikosti stránky 4KB se jedná o tabulku s 1MB položek, při velikosti 4B na každou položku to jsou 4MB stránkovací tabulky).
2. Rychlost přístupu do stránkovacích tabulek (pokud instrukce má paměťový operand, tak se ještě musí použít stránkovací tabulky obvykle v paměti)

Víceúrovňové tabulky stránek

Řeší problém s velikostí stránkových tabulek. Virtuální adresa se rozdělí na několik polí, které pak slouží jako indexy do jednotlivých úrovní stránkových tabulek.



Stránkovácí tabulky druhé úrovně tam nemusí být namapované (stejně jako stránky).

Asociativní paměť

Řeší problém rychlosti přístupu do stránkovacích tabulek v paměti. Využívá lokality programů, tj. program v jistém časovém úseku využívá pouze několik stránek paměti.

Asociativní paměť zabraňuje průchodu stránkovacími tabulkami při každém paměťovém odkazu. Obsahuje několik položek (řádově desítky) a každá položka je dělena na dvě části: **klíč** a **hodnotu**. V klíči jsou čísla stránek a hodnoty jsou čísla rámců.

Při převodu virtuální adresy na fyzickou adresu se nejprve paralelně prohledává asociativní paměť na číslo stránky podle klíče. Pokud tento dotaz na asociativní paměť uspěje, již se nepřistupuje do stránkovacích tabulek. Pokud v asociativní paměti hledané číslo stránky není, použije procesor stránkovací tabulky pro nalezení mapování. Pokud mapování uspěje, je zapsáno do asociativní paměti (na úkor jiné položky). Pokud ne, nastane výpadek stránky.

Nulaúrovňové stránkování

Nejsou žádné stránkovací tabulky, pouze větší asociativní paměť. Neúspěšné hledání v asociativní paměti je hlášeno OS jako výpadek stránky a OS plní přímo asociativní paměť.

Inverzní stránkovací tabulky

U větších virtuálních adresových prostorů (64 bitů) není možné používat klasické stránkovací tabulky pro jejich velikost. Fyzický adresový prostor je ale stále malý. Inverzní stránkovací tabulky jsou organizovány podél rámců a prohledávají se např. hašováním na mapování.

Algoritmy výměny stránek

Při výpadku stránky musí OS rozhodnout, který rámec uvolní pro nově mapovanou stránku.

Optimální stránka

V okamžiku výpadku stránky je jistá množina stránek v paměti. Každá tato stránka může být označena číslem, které udává počet instrukcí provedených před prvním přístupem na tuto stránku. Odstraněna je pak stránka s největším takovým číslem. Nelze ho implementovat, protože OS nemůže predikovat chování programů.

NRU (Not Recently Used)

Ke každé stránce jsou k dispozici dva příznaky: **A (accessed)** je nastaven při každém přístupu do dané stránky, **D (dirty)** je nastaven při zápisu do dané stránky. Nastavuje je HW a zůstávají nastaveny, dokud je nevynuluje OS. Pokud HW tyto příznaky nepodporuje, lze je simulovat.

Algoritmus: periodicky jsou příznaky A nulovány pro všechny stránky. Při výpadku stránky jsou pak stránky rozděleny do čtyř tříd:

	A	D
0	0	0
1	0	1
2	1	0
3	1	1

NRU pak vybere náhodně stránku z nejnižší neprázdné třídy.

Hodinový algoritmus

Stránky jsou v kruhovém seznamu. Při výpadku stránky algoritmus zkoumá stránku, na kterou ukazuje "ručička". Pokud je příznak A nastaven, vynuluje se a ručička postoupí na další stránku. Pokud je příznak A nulový, tato stránka je vybrána.

LRU (Least Recently Used)

Pozorování: Stránky často používané v předchozím časovém úseku (krátkém) budou v nejbližších okamžicích opět použity. Stránky dlouho nepoužívané nebudou v nejbližší době použity.

Pro plnou implementaci seznam s nejdéle nepoužitou na začátku a naposledy použitou na konci. Při každém přístupu na stránku je pak nutné tento seznam pozměnit.

Pomocí HW existují dvě metody:

1. CPU má 64-bitový čítač, který zvětšuje při vykonání každé instrukce. Každá položka stránkovacích tabulek má místo pro uložení tohoto čítače. Při každém přístupu na stránku je tento čítač zapsán do stránkovacích tabulek. Při výpadku stránky se vybere stránka s nejnižší hodnotou tohoto čítače.
2. Počítač s n rámci obsahuje matici $n \times n$ bitů. Na počátku je matice vynulována. Při každém přístupu do rámce k se nastaví celý řádek k na 1 a sloupec k se vynuluje. Nejdéle nepoužitý rámec je ten s nejnižší binární hodnotou řádku.

NFU (Not Frequently Used)

SW řešení LRU. SW čítač pro každou stránku, na začátku je nulový. Periodicky je k čítačům u stránek přičítán příznak A. Při výpadku se vybere stránka s nejnižším čítačem.

Nevýhodou je to, že NFU "nezapomíná", tj. v minulosti často používané stránky mají velkou hodnotu čítače a ačkoliv se již nepoužívají, algoritmus je nevybere.

Úprava: Před přičtením příznaku A se čítač posune o jeden bit doprava a příznak A se přičítá na nejvýše významný bit čítače (doleva). Tato úprava se nazývá **stárnutí** (aging).

Alokační strategie

Lokální	každý proces má přidělenou svoji množinu rámců a v ní stránkuje
Globální	všechny procesy používají všechny rámce

Implementace stránkování

Znovuspuštění instrukce

Sdílené stránky - snadné ve víceúrovňových stránkovacích tabulkách

Odstranění položky z asociativní paměti při rušení mapování

Ošetření výpadku stránky

1. Proces přistoupí na adresu ve virtuálním paměťovém prostoru, pro kterou neexistuje mapování. Vyvolá se OS (typicky přerušením) pro obsluhu výpadku stránky.
2. Uloží se stav přerušného procesu (registry, ...).
3. OS zjistí adresu ve virtuální paměti, která způsobila výpadek (často ve speciálních registrech CPU).
4. OS zkontroluje platnost adresy a přístupová práva. Pokud se jedná o neoprávněný přístup, proces je potrestán. Pokud je adresa platná, OS se snaží najít volný rámec. Pokud takový neexistuje, spustí se algoritmus výměny stránek.
5. OS zruší mapování na vyhazovanou stránku.
6. Je-li vyhazovaná stránka modifikována (D je nastaven), je třeba tuto stránku uložit na disk.
7. OS zjistí polohu požadované stránky na disku a načte ji do volného rámce.
8. OS zavede mapování na načtenou stránku.
9. Obnoví se stav přerušného procesu a ten má možnost se rozběhnout, jako by se nic nestalo.

Segmentace



Stránkování chápe virtuální adresový prostor jako jednorozměrný. Segmentace zavádí dvojrozměrný virtuální adresový prostor.

Segment je nezávislý adresový prostor sestávající z adres od 0 do limitu segmentu. Segmenty mohou mít různé velikosti, a ty se mohou měnit v průběhu času. Segment má své umístění ve fyzické paměti (neviditelné pro procesy).

Každá adresa se pak sestává z dvojice (s, d) , kde s je číslo segmentu a d je adresa v segmentu.

Segmenty mohou být přítomny nebo nepřítomny a dochází k výpadkům segmentu (obdoba výpadku stránky).

Při alokaci segmentu se používají algoritmy typu first-fit.

Je možná i kombinace stránkování a segmentace.