**Řízení operační paměti**

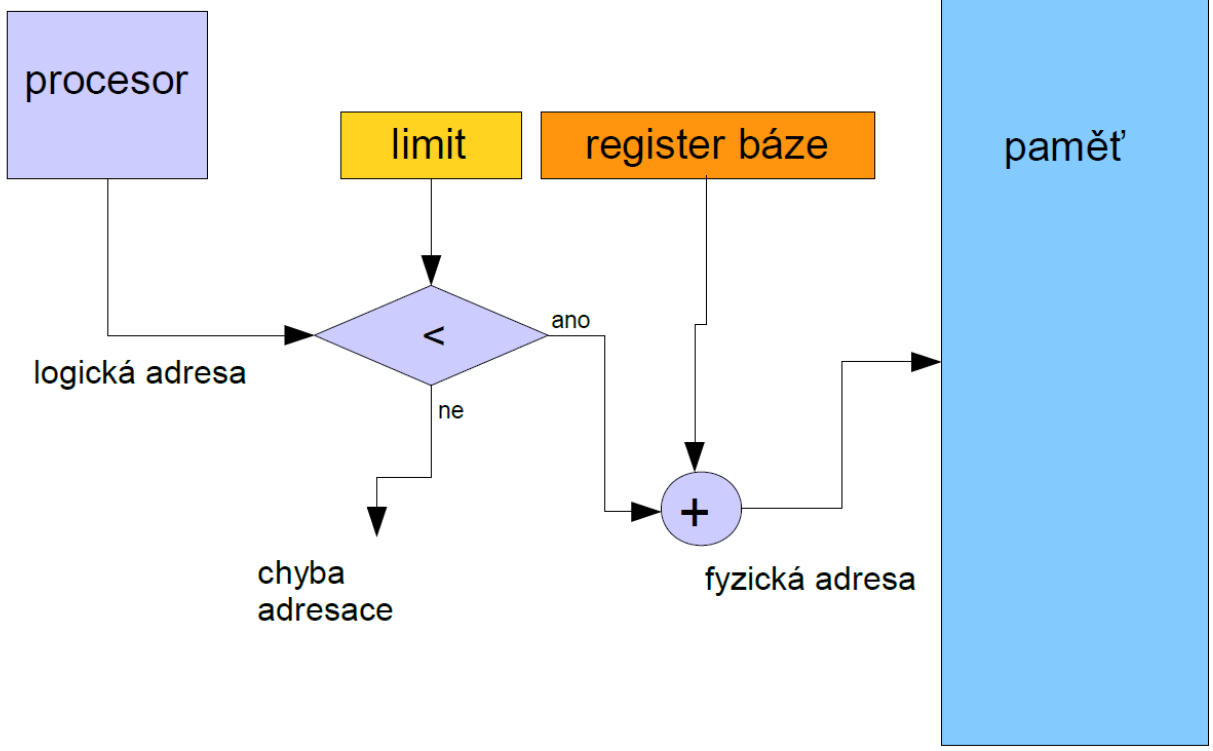
**K čemu slouží OP a co se v ní nachází**

* Slouží k zpracování instrukcí a dat procesů
* Je nutno ji spravovat -> správce paměti

1. Přidělovat OP jednotlivým procesorům, když si ji vyžádají
2. Udržovat informace o paměti (např. která část pamětí je obsazena, jakým procesem, zda část paměti není určena pro OS, adresy)
3. Zařazovat uvolněnou paměť zpátky do volně části (i spojování jednotlivých volných částí)
4. Odebírat pamět procesu v případě nutnosti (i kdyby proces spadnul)
5. Zajistit ochranu paměti(zamezit přístupu procesu mezi sebou ať nezačne zapisovat jinam)

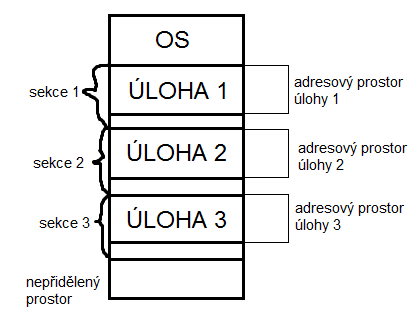
**Strategické přidělování OP**

1. Přidělování souvislé oblasti (veškeré volné paměti)

****

* Pouze 1 uživatelský proces (zbytek OS)
* V případě nutnosti více procesů je možné využít swapování (ostatní procesy jsou odloženy na disk)
* Pokud by byl program větší než přidělené místo je možné využít překrývání (overlays) -> nejnutnější část kódu je pořád v OP a zbytek se bude překrývat (o to se stará programátor ne OS!)

1. Přidělování paměti po blocích

****

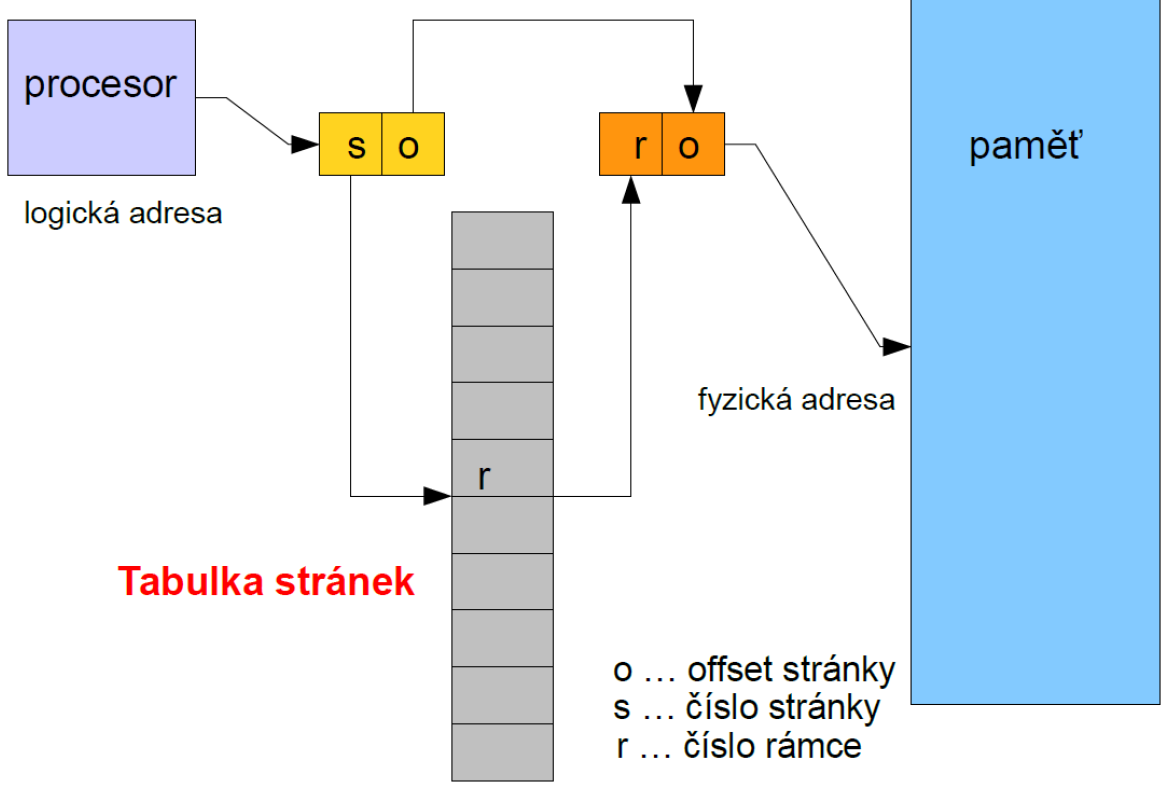
* Každý proces je zařazen do 1 bloku
* Nevýhoda: na začátku je nutné znát požadavky na velikost nejvyšší úlohy, aby bylo možno ji vložit do bloku
* Vzniká vnitřní fragmentace (nevyužitý prostor se nedá využít nikde jinde)
* Vzniká i externí fragmentace (nepřidělený prostor)
* Správce paměti je schopen spojit volné bloky, které na sebe navazují -> čímž je možno obsloužit i náročnější úlohy -> musí to být bloky vedle sebe!

Alokační strategie

1. First fit – první vyhovující dostatečně velký blok kam by se úloha mohla vejít tam se přidělí
   * Rychlé, nejčastěji používané
2. Last fit – umisťuje se do poslední vyhovující
3. Best fit – minimální volný prostor
4. Worst fit – nejvíce volného prostoru (plýtvání místa)

* Určitým vylepšením je dynamické přidělování sekcí – až za běhu zjistí kolik potřebuje místa a teprve potom si ji přidělí -> potlačení vnitřní fragmentace, pomalejší
* [Dynamické přemisťování sekcí](ODKAZY/dynamickeprempam.jpg)

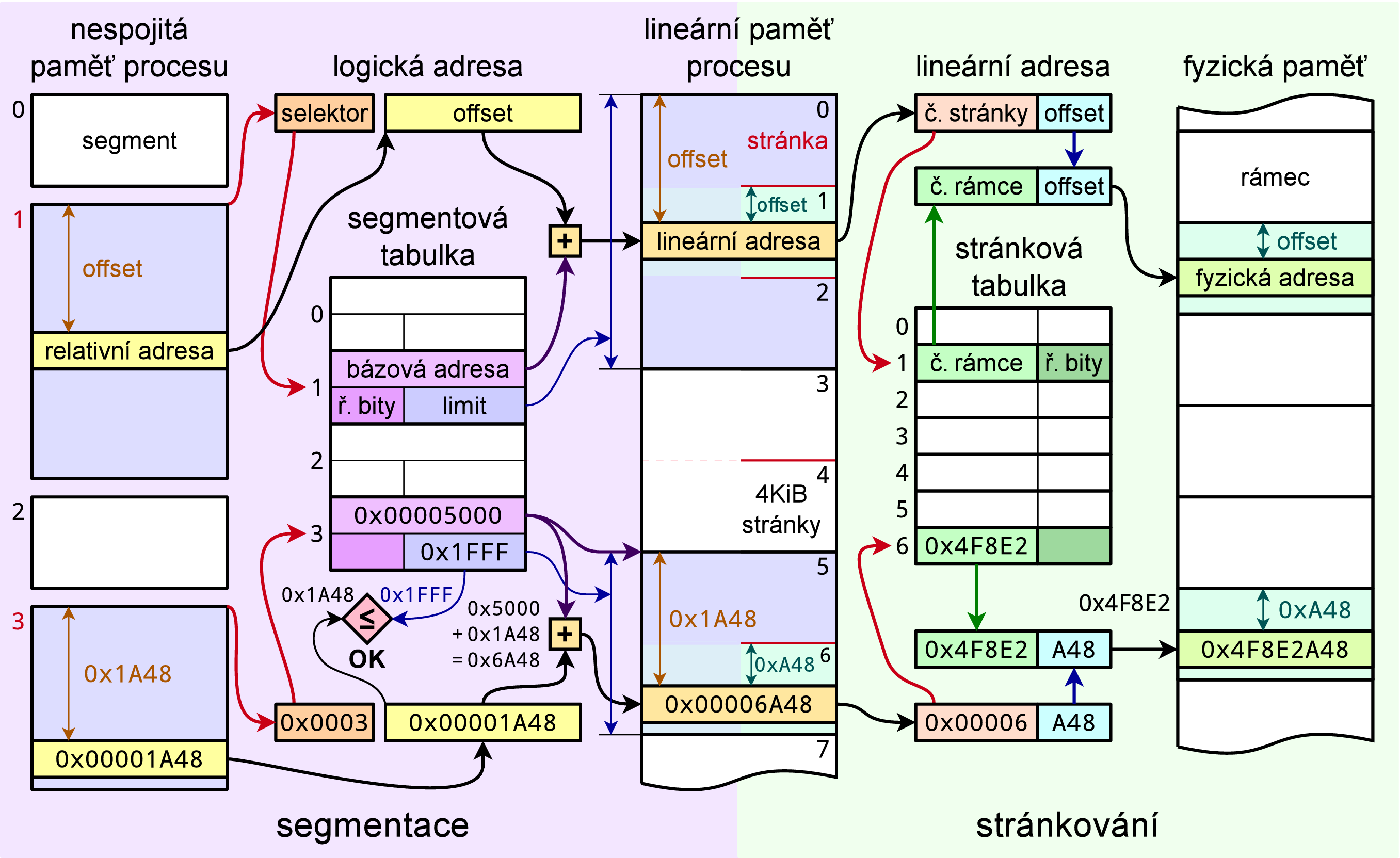
1. Stránkování



* Procesor vysílá požadavek logické adresy(číslo stránky a offset)
* Číslo stránky – index ve stránkovací tabulce, která náleží procesu
* Číslo rámce (r), je na indexu, to se vybere, společně s offsetem se spojí a vznikne konkrétní fyzická adresa
* RAM je rozdělena na rámce stejné velikosti
* Každý proces je rozdělen na stránky stejné velikosti
* Velikost stránky je stejná jako velikost rámce
* V 1 rámci uložena 1 stránka
* Offset – posun v rámci stránky nebo rámce
* Procesy pro svůj běh potřebují souvislý běh paměti, při jejich uvolňování však dochází k fragmentaci -> možno řešit přemisťováním bloků
* 2. způsob, ale ten je časově náročný
  + Stránkování umožňuje přidělit procesu několik nesouvislých úseků paměti a vytvořit pro proces iluzi, že se jedná o spojitou paměť
* [obrazek](ODKAZY/strankovani.jpg)
* Offset je 12 bitový
* Adresa má 32 bitů (12bit offset + 20 bit číslo stránky, rámce)
* V tabulce stránek se kromě adresy rámce nachází i atributy pro danou stránku (pouze pro čtení, je možno je modifikovat, jestli byla použita, jestli byla modifikována)
* Výhody
  + Potlačení přemisťování bloků
  + Zrušení externí fragmentace
* Nevýhody
  + Nutná HW podpora
  + Omezený počet použití
  + Vznik vnitřní fragmentace
* Zrychlení stránkování (vyhledávání, načítání) – TLB cache -> obsluhuje poslední použité stránky, kontroluje se jako první

1. Segmentace

* [obrazek](ODKAZY/segmentace.jpg)
* Segment je logické seskupení informací (hl. program, konstanty, mat. fce)
* Program se skládá z kolekce segmentů a ty mají k dispozici logický adresní prostor
* Segmenty se mohou lišit svou velikostí
* Každý proces obsahuje svou segmentovanou tabulku ve které najdeme limit (velikost segmentu, ochrana přístupu do paměti) a base (kde začíná segment)
* Číslujeme od 0
* Výpadky segmentů => obdobné algoritmy jako stránkování -> náročnější u větších segmentů
* Segmentace umožní aby procesy, které jsou řízeny stejným programem sdílely kód programu a konstanty (úspora vnitřní paměti)
* U systémů se sdílenou pamětí je vhodné sdílená data uložit do zvláštního segmentu a ten zpřístupnit všem procesům, které je používají (zlepšení ochrany paměti, protože procesy přistupují na konkrétní společná místa)
* Výhody
  + Velikost segmentu odpovídá skutečné potřebné části procesu -> odpadá vnitřní fragmentace
  + Možnost detekce chyb v programech (offset ukazuje mimo segment)
  + Možnost sdílení kódu mezi úlohami
* Nevýhody
  + Náročnost alokace segmentů (různé velikosti)
  + Režie přístupů do paměti -> převod z logické adresy na fyzickou
  + Nutná Hw podpora

1. Segmentace se stránkováním

* Rozdělení na segmenty(ty jsou různě velké) -> pak se rozdělí na stránky (stejně velké) -> následně se uloží do rámců
* Adresa 48 bit
  + 16 bit selektor
  + 32 bit offset
* Offset
  + Posun v rámci segmentu
  + Relativní adresa
* Selektor
  + 13 bit index
  + 2 bit pro úroveň ochrany segmentu
  + 1 bit popisovač (říká jestli segment pouze pro 1 proces nebo více procesů LDT, GOT, Local/Global description table)
  + Index
    - Bázová adresa = začátek segmentu
    - Řídící bity
    - Limit
* Lineární adresa vzniká součtem bázové adresy a offsetu (obě informace jsou 32 bit -> výsledek taky 32 bit)
* Lineární adresa
  + 10 bit stránkovací adresář
  + 10 bit stránkovací tabulka
  + 12 bit offset – posun v konkrétní stránce (rámci)
  + Prvních 10 bitů je nějakým indexem do stránkovacího adresáře -> str. tabulka
  + Druhých 10 bitů je indexem do stránkovací tabulky (č. rámce se vytáhne a spojí se s offsetem -> vzniká fyzická adresa)
* Fyzická 32 bit – můžeme přistoupit na konkrétní místo v paměti

**Přidělování paměti**

* Jak stránkování na žádost tak segmentace na žádost jsou také označovány za líné metody -> nedělají nic dopředu
* Při dopředném stránkování, nemůže systém vědět, která stránka bude potřeba a není možné mít všechny stránky všech procesů v OP

Čisté vs špínavé stránky

* Čisté = beze změny jsou v OP tak v disku
* Špinavé = došlo ke změně v OP, ale na disku jsou v předchozí verzi (nutno zálohovat/aktualizovat)

Precleaning (čistění)

* když má PC čas, zkopíruje stránky do virtuální paměti
* až dojde na požadavek

Výprask (thrashing)

* vyměňuje stránky z OP na disk -> špatně zvolený algoritmus nebo nedostatek OP (slap storm)

**Stránkování na žádost**

* z důvodu uvolnění fyzické paměti a umožnění tak využití pro další procesy jsou některé stránky odloženy na disk do tzv. swapovacího oddílu
* v tabulce stránek je umístěn údaj zda je stránka na disku nebo v RAM => stavový bit
* pokud je stránka na disku dojde k vyvolání výpadku stránky -> obslužný program pro toto přerušení musí zajistit nahrání stránky do OP, upravit údaj v tabulce stránek a zajistit opakované volání instrukce, která výpadek způsobila
* v případě obsazenosti paměti je nutné odložit jiný rámec na disk -> algoritmy nahrazování stránek

Algoritmy nahrazování stránek

Optimální algoritmus

* nahrazení stránky, která bude odkazována (volána) ze všech nejpozději
* [optimalni](ODKAZY/optimalni.jpg)
* Není možné ho naprogramovat (neznáme požadavky) -> slouží jako ideální příklad
* Výpadky – počet změn stránek (to zakroužkované)

FIFO algoritmus

* First in first out
* První příjde první odejde
* [FIFO](ODKAZY/fifo.jpg)

LRU

* List recenty used
* Nejdéle nepoužitá stránka jde pryč (nepoužitá v tom prvním řádku STR)
* [LRU](ODKAZY/lru.jpg)

Druhá šance

* Udělá se fronta (viz obrázek, je pod tabulkou), pokud má stránka hvězdičku neodstraní se z fronty ale odstraní se pouze hvězdička a dá se nakonec fronty, nahradí se pak další stránka ve frontě
* Hvězdička se přidá, když už je dané číslo v rámci
* Nově přidaná stránka jde VŽDY na konec fronty
* [Druha\_sance](ODKAZY/druhasance.jpg)

Hodiny

* Modifikovaná druhá šance
* Šipka ukazuje, která stránka vypadne
* Když je políčko, na které šipka ukazuje prázdné zapíše se do něj stránka a šipka se posune
* Pokud je číslo v políčku stejné jako stránka, která chce do fronty přidá se hvězdička (nebo exponent toho čísla to je jedno) a šipka zůstává na stejném místě
* Pokud pak šipka ukazuje na toto číslo a jde další stránka škrtne se hvězdička šipka se posune na další políčko a číslo na tom políčku se přepíše, potom se šípka opět přesune
* [hodiny](ODKAZY/hodiny.jpg)

Random

* Náhodné vyloučení stránky

NFU

* Not frequently used
* Čítač u stránek, který se periodicky navyšuje
* Vyloučí se ta s nejmenším číslem
* Poslední příchozí jsou nejvíce ohroženy (hrozí stárnutí)
* Řešení: pokud bude stránka opětovně volána navýší se její číslo

NUR

* Not used recently
* Nepoužitá stránka nahrazena
* Obsahuje 2 kontrolní bity pro každou stránku (00,11,10,01)
* Levý bit = modifikace
* Pravý bit = jestli byla použita
* 00….ani modifikovaná ani nečtená = šla by pryč první
* 01….pouze čtená
* 10….pouze modifikovaná
* 11….čtená i modifikovaná = vydrží nejdéle v OP
* Modifikace nastavována hardwarově