

SPRÁVA PAMĚTI

Správa paměti je v informatice soubor metod, které operační systém používá při přidělování operační paměti jednotlivým procesům, které jsou v počítači spuštěny. Může zajišťovat i následné uvolňování paměti, nastavovat ochranu paměti a eventuálně i správu adresace paměti. Všechny tyto činnosti v operačním systému zajišťuje „**správce paměti**“ (MM – Memory Manager). Pro některé činnosti je obvykle nutná hardwarová podpora v procesoru, kterou zajišťuje **jednotka správy paměti** (MMU – Memory Management Unit).

OS jako správce paměti má tyto hlavní úkoly:

- **Přidělovat** operační paměť jednotlivým procesům, když si ji vyžádají.
- **Udržovat** informace o paměti, o tom, která část je volná a která přidělená (a komu).
- **Zařazovat** paměť, kterou procesy uvolní, opět do volné části.
- **Odebírat** paměť procesům, je-li to zapotřebí.
- **Zajistit ochranu** paměti
- V případě potřeby **využít sekundární paměti** (např. pevný disk) pro zvýšení kapacity OP

!!! Pozor, snaha uspokojit požadavky všech procesů nesmí vést k zablokování preempce (zablokování výměny procesů na CPU)

Ochrana paměti

Ochrana paměti je v informatice způsob řízení práv přístupu k paměti počítače a je součástí většiny moderních operačních systémů. Hlavním účelem této ochrany je zabránit procesu v přístupu k paměti, která pro něj nebyla alokována. Předchází se tak scénáři, kde chyba uvnitř jednoho procesu ovlivní běh dalších procesů či samotného operačního systému.

FAP versus LAP

Je-li operační paměť reprezentována pamětí s přímým přístupem označujeme adresový prostor jako **fyzický adresový prostor (FAP)**. Velikost tohoto prostoru je omezena buď fyzickou velikostí paměťových modulů anebo velikostí adresy, tj. adresa o velikosti „n“ bitů umožňuje adresovat „2 na n-tou“ paměťových míst.

Jednodušší procesory umožňují adresovat pouze paměť s přímým přístupem, tedy adresovat pouze fyzický adresový prostor. V dnešní době většina procesorů umožňuje adresovat i tzv. **logické adresové prostory (LAP)**. Jedná se o tzv. **virtuální paměť**. Jde v podstatě o speciální správu paměti, která umožňuje programům a aplikacím, přistupovat k místu na pevném disku stejně jako k operační paměti RAM, což umožňuje (na úkor snížení výkonu) snadno rozšířit kapacitu paměti počítače.

Když je **LAP** \leq **FAP**, tak jde o takzvané reálné přidělování paměti

Když je **LAP** $>$ **FAP**, tak jde o takzvané virtuální přidělování paměti

Reálné metody přidělování paměti

Každý proces potřebuje ke svému běhu přidělení určité části operační paměti. Proto má každý operační systém **modul správy paměti (MM)**, který zajišťuje přidělování a ochranu paměti. Podle implementace modulu správy paměti rozlišujeme:

1. přidělování souvislé (veškeré) volné paměti (**monolitické bloky**)
2. přidělování bloků paměti pevné velikosti (**statické bloky**)
3. přidělování bloků paměti proměnné velikosti (**dynamické bloky**)
4. **Segmentace** (metoda přidělování paměti, která zároveň řeší ochranu paměti)
5. **Stránkování** (metoda přidělování paměti, která zároveň řeší ochranu paměti)

1. MONOLITICKÉ BLOKY

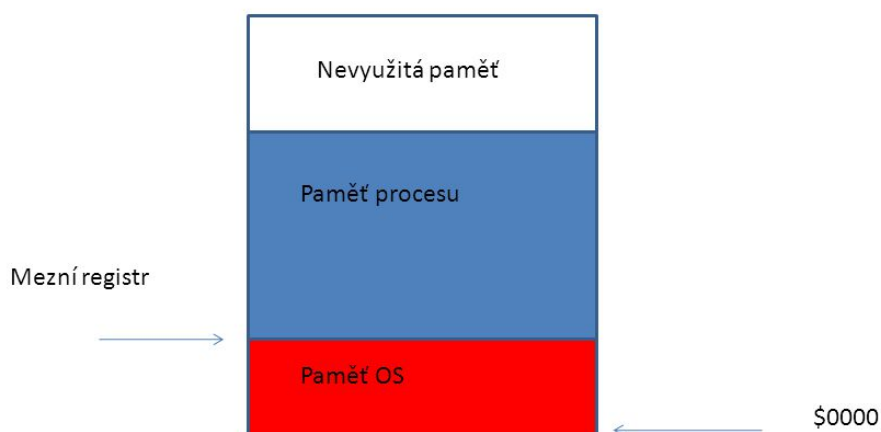
Jedná se o nejjednodušší správu operační paměti. Fyzický Adresový Prostor (**FAP**) je zde rozdělen na **dva bloky**, tj. na dvě souvislé části, jež jsou určeny počáteční adresou. První blok je přidělen rutinám jádra operačního systému a jeho datovým strukturám tj. „**Kernel memory**“. Druhý je pak přidělován na požádání ostatním procesům tj. „**Application memory**“. Paměť jádra sdílí všechny procesy, protože rutiny jádra včetně jeho datových struktur jsou využívány veškerými procesy při vykonávání služeb operačního systému. Paměť procesů neboli aplikační paměť je soukromého charakteru, tedy přístup k ní má pouze vlastník.

Regulace je prováděna velice triviálně. Je-li paměť **volná** tzn. není-li alokována **je přidělena procesu celá bez ohledu na požadovanou velikost**, přičemž nesmí přesáhnout velikost bloku. V případě obsazené paměti jiným procesem je požadavek na přidělení odmítnut. Toto zamítnutí může mít fatální důsledky pro proces žádající o paměť. K alokování paměti vlastně dochází jen při spuštění procesu a proces ji využívá po celou dobu svého životního cyklu. K uvolnění dojde po ukončení života procesu.

Tato strategie správy paměti se hodí pouze pro jedno-úlohové OS. I když se jeden čas používala ve více úlohových systémech. Kdy byl tento princip jediného bloku v paměti rozšířen o možnost odkládání paměťového regionu do sekundární neboli odkládací paměti, (např. pevný disk). Obnova procesu z pevného disku je velice pomalá, a proto je tato strategie v dnešních systémech téměř nepoužitelná. Jak je asi pochopitelné, vzniká zde problém s nedostatečným využitím aplikační paměti. V praxi to znamená, že proces s malou velikostí zabere celou aplikační paměť.

Ochrana paměti je zde realizována pouze v případě ochrany paměti jádra - (Kernel memory) pomocí tzv. **limitního (mezního) bázového registru**. V praxi jde o znemožnění použití nižších adres, než je báze. V registru je uložena báze bloku aplikační paměti a paměť jádra leží tedy před (pod) tímto blokem. Proces tedy ani nemůže adresovat paměť jádra. Registr se nastavuje pomocí **privilegované instrukce**, která se může volat pouze v režimu jádra.

Přidělování jedné souvislé oblasti paměti



2. STATICKE BLOKY

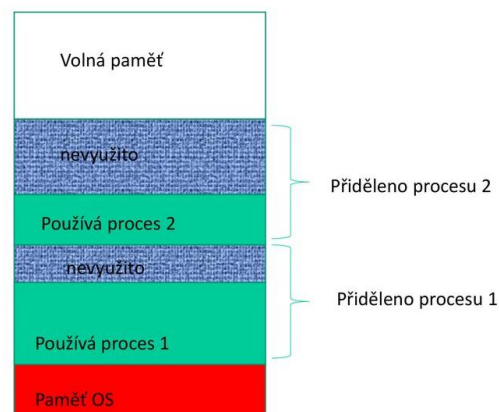
Jedná se o jedno z možných řešení nedostatečného využití aplikační paměti. **Aplikační paměť je rozdělena na několik samostatných bloků, které lze alokovat samostatně.** Tedy nepočítáme-li procesy odložené v sekundární paměti, **maximální počet procesů je omezen počtem bloků.** Jeden proces může zabírat i více nesouvislých bloků (bloky oddělené jiným blokem). Velikost a počet bloků vycházejí z druhu aplikací, které mají být na daném operačním systému provozovány.

Strategie přidělování bloků je založena na stejném principu jako u čistě monolitické paměti s rozdílem, že žádajícímu procesu o přidělení paměti je přidělen blok o nejmenší velikosti a zároveň dostatečné velikosti. Informace o obsazení bloků jsou udržovány poli o stanovené velikosti.

Jelikož je aplikační paměť rozdělena na již zmíněné bloky, je nutné zajistit jejich ochranu. Na **ochranu paměti** se používají také tzv. **limitní (mezní) registry procesoru.** Použitím dvou mezních registrů (nastavuje OS), které **uvádějí nejnižší a nejvyšší dostupnou adresu.** V limitním registru procesoru je uložena hodnota aktuálního paměťového regionu (bloku). Hodnota lokální adresy se porovnává s hodnotou limitního registru. Pokud je tato hodnota větší, následuje vyvolání výjimky – proces se pokouší zapsat mimo region (oblast pro něj alokovanou). Druhou metodou, jak zajistit ochranu paměti je: **Mechanismus zámku a klíčů** – spočívá v rozdělení paměti na stránky o pevné velikosti. Každé je přidělen zámek (celé číslo). Ve speciálním registru procesoru je klíč. Ty stránky, které mají stejnou hodnotu zámku jako je hodnota klíče přidělená konkrétnímu procesu, proces odemkne (může proces používat).

Adresový prostor procesu se většinou skládá ze tří regionů: **kódového, datového a zásobníkového regionu.** Kódový region většinou obsahuje kód programu. Datový region nese statická data programu. Zásobníkový region má charakter LIFO (Last In – First Out) a jsou zde obsaženy lokální datové proměnné a návratové adresy funkcí (viz předmět programování).

Přidělování bloků pevné velikosti



3. DYNAMICKÉ BLOKY

Aplikační paměť je rozdělena na bloky, jejichž velikost se dynamicky upravuje dle požadavků procesů. Před alokací prvního regionu (paměťového místa) tvoří paměť aplikačního prostoru jeden blok.

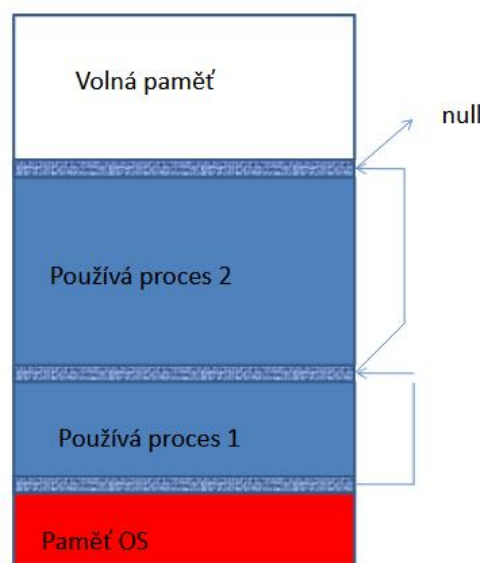
Při alokaci (přidělování) paměti se vyhledá opět první přípustný blok, pokud je jeho velikost rovna či blízká požadavku procesu, je blok použit celý. V ostatních případech je blok rozdělen na dva jeden o požadované velikosti procesu a druhý zůstává jako volné místo. (Existují i další metody přidělování bloků (First-fit, Last-fit, Worst-fit..., které zde už probírat nebudeme.) Když dochází k uvolnění bloků, je nutné provádět tzv. **zcelování volných bloků** - (Vznikne-li souvislá řada 2 či 3 bloků, tak jsou spojeny do jednoho.)

Obsazení paměti je v tomto případě realizováno na počátku každého bloku, **informace zde tvoří jakousi hlavičku**. Nutné je velice pečlivě ochránit hlavičky jednotlivých bloků. Nevýhodou tohoto systému je velký sklon k fragmentaci paměti, respektive vzniku velkého množství volných malých nesouvislých bloků. Je to velký problém zejména u víceúlohových systémů kde žádosti o alokování a dealokování přicházejí v libovolném a nezávislém pořadí.

Fragmentace paměti se dá řešit tzv. **setřásáním bloků**. V praxi se jedná o soustředění volné paměti do jediného bloku většinou na konci paměti.

Setřásání bloků bylo dříve používáno v kooperativních víceúlohových systémech (Windows 3, starší verze MacOS a Symbian).

Přidělování bloků proměnné velikosti (dynamické přidělování paměti)



METODY OCHRANY PAMĚTI U PAMĚTI DĚLENÉ NA DYNAMICKÉ BLOKY:

Metoda mezních registrů:

- paměť je rozdělena na bloky různých velikostí (**SEGMENTY**)
- každému programu je přiřazena limitní bazová adresa (privilegovanou operací)
- program pracuje s tzv. logickou adresou (offset)
- program nesmí adresou offsetu přesáhnout (jemu přidělený adresovatelný limit)
- při porušení ochrany → vnitřní přerušení (zařídí OS)
- metoda mezních registrů se běžně používá v architektuře x86

Metoda stránkování:

- podporuje virtualizaci paměti, multitasking a sdílení paměti mezi procesy
- proces pracuje s virtuální pamětí, jejíž velikost je rovna maximální možné adresovatelné hodnotě daného systému. (32b, 64b což je 2^{32} nebo 2^{64})
- paměť je rozdělena na bloky stejných velikostí (např. 4096 B) (vyspělejší architektury ovšem podporují různé, avšak předem definované, velikosti bloků)
- každý proces dostane přidělenou tabulku (tzv. primární tabulku), ve které pevně daný počet položek definuje:
 - překlad každé virtuální adresy na fyzickou adresu, nebo odkaz na tabulku vyšší úrovně – existuje-li překlad takové stránky pro proces, je tato část povinná
 - příznak existence stránky (stránka může být uložena na externím úložišti, nebo pro proces neexistuje (výpadek stránky) → zařídí OS)
 - přístupové práva (čtení, zápis, stránka obsahující spustitelný kód)
 - a další...

Mechanismus zámků a klíčů:

- paměť je rozdělena na bloky stejné velikosti (např. 4 KiB)
- každému programu přidělíme zámek (celé číslo)
- bloky patřící danému programu označujeme stejným číslem (zámek)
- v CPU je speciální registr s číslem běžícího procesu programu (klíč)
- CPU povolí jen typ operace s pamětí, kde zámek=klíč
- při porušení ochrany → vnitřní přerušení (zařídí OS)
- speciální klíč 0 slouží OS (bez ochrany)
- nastavení registru s klíčem je privilegovanou instrukcí
- zámky jsou uloženy v operační paměti jako pole
- metoda zámků a klíčů se používá dodnes

DOVYSVĚTLENÍ již zmiňované SEGMENTACE a STRÁNKOVÁNÍ paměti:

4. Segmentace

Segmentace je metoda správy paměti, kdy je procesu vytvořen virtuální adresní prostor začínající od nuly, čímž odpadá potřeba relokace (znovu umístění) použitého strojového kódu.

Každý proces (resp. jeho strojové instrukce) pak bude pracovat s logickými (virtuálními) adresami, které budou vyjadřovat vzdálenost od začátku tohoto vlastního vyhrazeného adresního prostoru (tj. logická adresa bude pro každý proces začínat od nuly). Logické adresy používané strojovými instrukcemi jsou pak v případě přístupů do paměti automaticky překládány na fyzické adresy a zpět.

Segmentace používá pro vytvoření logické (virtuální) adresy dva speciální registry – **segment** (místo v paměti) a **offset** (odsazení v paměti), přičemž jejich součet vyjadřuje fyzickou adresu.

Ochrana paměti je zajištěna **mezním registrem na offset** (procesor nedovolí použít offset větší, než je hodnota v mezním registru). Změna mezního registru a segment registru je privilegovaná instrukce, takže ji může provést pouze jádro operačního systému. Běžící proces tedy může v neprivilegovaném stavu volně pracovat pouze s offsetem. Pokusí-li se proces běžící v neprivilegovaném režimu provést privilegovanou instrukci (tj. změnu segmentu nebo mezního registru), vyvolá procesor vnitřní přerušení (přičemž obsluha přerušení se nachází uvnitř jádra operačního systému), které tak může dotyčný proces okamžitě ukončit.

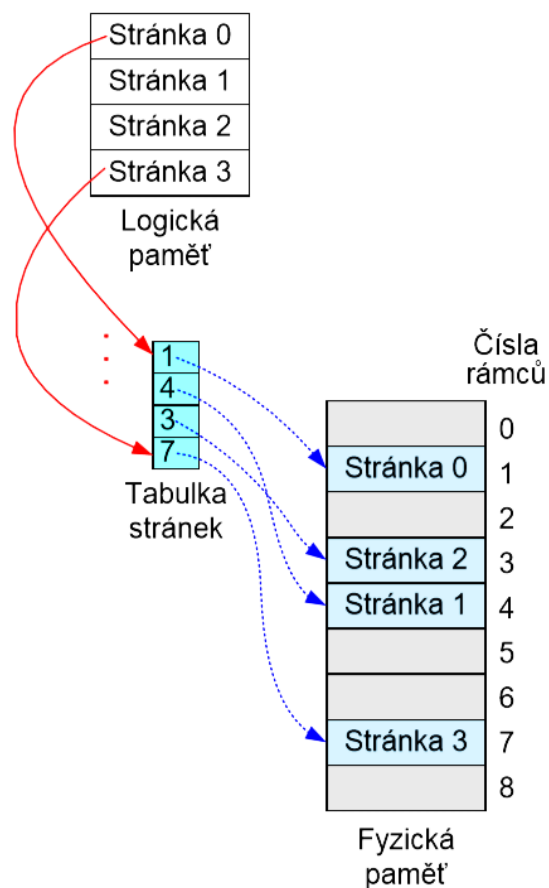
5. Stránkování (Stránkovaná virtuální paměť)

Při stránkování je paměťový adresní prostor rozdělen na malé, stejně velké, části, nazývané stránky. Následným zapojením mechanismu virtuální paměti je pak možné umístit každou stránku do kteréhokoliv místa ve fyzické paměti, či stránku označit jako chráněnou. Libovolné umístění stránek tak umožňuje zachovat lineární adresní prostor v rámci bloku, který je ve fyzické paměti složen z fragmentů.

Většina počítačových architektur založených na stránkování používá stránkování jako způsob ochrany paměti.

Tabulka stránek je použita k mapování virtuální paměti do fyzické. Tato tabulka je pro proces obvykle neviditelná. Tabulka stránek ulehčuje alokování nové paměti, protože každá nová stránka může být alokována z libovolného místa ve fyzické paměti.

Tímto způsobem je tak pro aplikaci nemožné přistoupit na stránku, která pro ni nebyla explicitně alokována a to proto, že žádná paměťová adresa, dokonce ani kompletně náhodná, kterou se aplikace pokusí použít, buď odkáže na stránku alokovanou pro tuto aplikaci, nebo dojde k vygenerování stavu **page-fault** (PF) (porucha stránky). Nealokované stránky a stránky alokované pro jiné aplikace jednoduše nemají z pohledu dané aplikace žádnou adresu.



ZÁVĚR

Existují ještě další metody správy a ochrany paměti (Simulovaná segmentace, Adresace založená na způsobilosti...)

Pro naše pochopení fungování správy operační paměti z pohledu operačního systému budou výše zmiňované metody stačit.