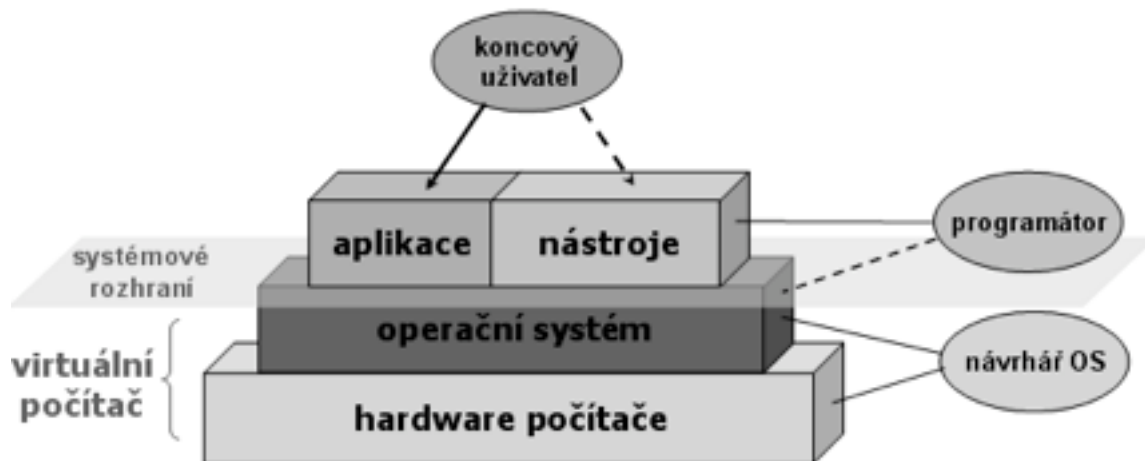


Úloha OS, prostředky počítače, představa virtuálního počítače

OS

- softwarová nadstavba HW
- společně s HW představuje virtuální počítač
- rozhraní mezi uživatelem a systémem
- prostředí pro provádění programů
 - předává a odebírá řízení uživatelským programům
 - izoluje programy a data jednotlivých uživatelů
- zobecněné rozhraní pro přístup k systémovým prostředkům
 - poskytuje uživatelům a programům služby virtuálního počítače, skrývá před aplikačními programy detaily hardwaru počítače
 - programátorům poskytuje jednoduchý, jednotný způsob přístupu k paměti, souborům a I/O zařízením

Vrstvy a vztahy v OS



Služby OS:

- prostředí pro tvorbu programů
- prostředí pro provádění programů
- přístup k I/O zařízením a souborům
- přístup k systému
- detekce chyb
- reakce na chyby
- sledování činnosti systému

zdroj wiki:

virtuální počítač:

Hardwarový virtuální stroj (QEMU, Microsoft Virtual PC, VirtualBox, VMware, Xen)

Aplikační virtuální stroj (Java Virtual Machine, Microsoft .NET Framework, Perl, Zend Engine - PHP)

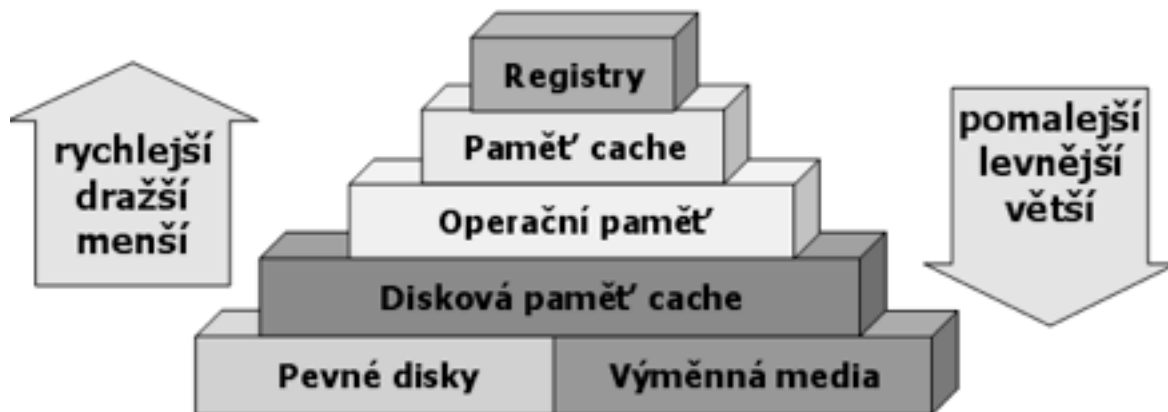
Hardware počítače, typ instrukčního cyklu procesoru, techniky programování vstupu a výstupu, DMA

Procesor (CPU)

- HW pro provádění programu
 - programový čítač PC (adresa následující instrukce)
 - instrukční registr IR (naposledy načtená instrukce)
 - dekodér instrukce a řídicí obvody
 - registr(y) s informacemi o stavu procesoru PSW (informace o stavu CPU)
- HW pro zpracování dat
 - registry pro krátkodobé uložení adres a dat
 - operační jednotka ALU
- Další pomocné registry a obvody
 - ukazatel zásobníku SP,...

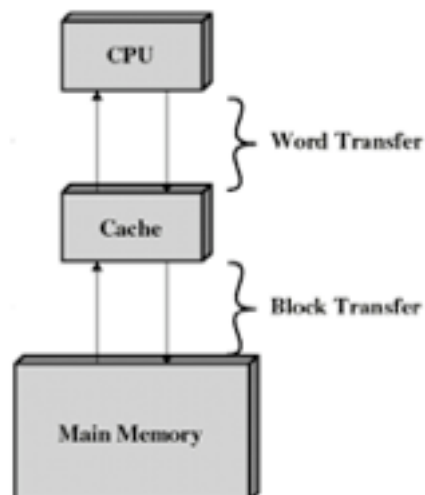
Operační paměť

HW sloužící k uložení programu a dat



Cache - vyrovnávací paměť zařazená mezi CPU a operační paměť

Procesor čte instrukce a data z paměti cache, pokud adresované slovo není v paměti cache, načte se z operační paměti do cache celý blok, obsahující požadované slovo; zápis: pokud procesor některé slovo v bloku modifikuje, musí se toto slovo (celý blok) zapsat zpět do operační paměti.



Základní instrukční cyklus

- v okamžiku zahájení provádění instrukce programový čítač PC obsahuje adresu příští instrukce
- CPU načte novou instrukci z operační paměti podle PC
- CPU dekóduje a provede načtenou instrukci
- obsah PC se automaticky zvýší tak, aby obsahoval adresu příští instrukce

Instrukční cyklus s přerušením

- CPU ověřuje požadavky na přerušení vždy před načtením nové instrukce
- pokud není přerušení povoleno nebo požadováno, CPU pokračuje normálně, tj. načte následující instrukci podle PC
- pokud je přerušení povoleno a požadováno, uloží se automaticky obsah PC a PSW např. do zásobníku a do PC a PSW se nahraje tzv. přerušovací vektor
- tím se změní adresa příští instrukce a nastavení přerušovacího systému

Typy přerušení:

I/O přerušení, časovač, programové chyby, chyby hardwaru

Techniky I/O komunikace

Programovaný I/O přenos

nepoužívá se přerušení, CPU musí čekat na dokončení každé I/O operace nevyžaduje od hardwaru, aby dokázal přerušit činnost CPU po dokončení I/O operace dokud I/O modul nedokončí akci, musí program čekat v době čekání na dokončení I/O operace zůstává procesor zaměstnán testování stavu I/O modulu

I/O přenos s přerušením

během provádění I/O operace může CPU pracovat - činnost CPU je přerušena po dokončení I/O operace

činnost procesoru je přerušena, jakmile je I/O modul připraven k další akci odpadá zbytečné čekání, po dobu provádění I/O operace je procesor volný pro jinou činnost

pro přenos se spotřebuje část času procesoru (procedura pro obsluhu přerušení musí obsloužit přenos každého slova dat)

I/O přenos s přímým přístupem do paměti (Direct Memory Access, DMA)

data se přenášejí po blocích přímo mezi I/O modulem a pamětí, bez účasti CPU program pouze inicializuje DMA přenos (odkud, kam, kolik,...)

procesor předá požadavek na přenos bloku dat DMA modulu

DMA modul přenese blok dat přímo mezi I/O modulem a pamětí, bez účasti CPU po dobu přenosu bloku je procesor zcela volný pro jiné použití, o dokončení přenosu bloku je procesor informován přerušením

Proces, stavy procesu, události, fronta procesů, odložení procesu, 7stavový diagram

Proces - konkrétní provedení určitého programu

Příklad: recept na dort = program -> pečení dortu podle tohoto receptu = proces

Operační systémy

Dispečer

- brání okupaci procesoru jediným procesem na nepřijatelně dlouhou dobu
- na základě plánovacího algoritmu rozhoduje, který proces bude prováděn jako další

Vznik procesu

- spuštěním úlohy v dávce
- přihlášením uživatele do systému
- spuštěním služby pro obsluhu požadavků např. tiskový server, www server,...
- vytvořením potomka existujícího procesu

Postup při vytvoření procesu

- přidělit procesu jedinečný identifikátor
- alokovat pro proces místo v paměti
- inicializovat řídicí blok procesu (PCB)
- nastavit odpovídající odkazy
- uložení záznamu o aktivaci procesu

Přepínání procesu

- přerušeni
 - proces vyčerpal přidělenou časovou kvótu
 - I/O operace vyžádaná procesem byla dokončena

Chyba přístupu k paměti

- adresované paměťové místo nebylo nalezeno v reálné paměti
- odpovídající část virtuální paměti musí být přenesena do operační paměti

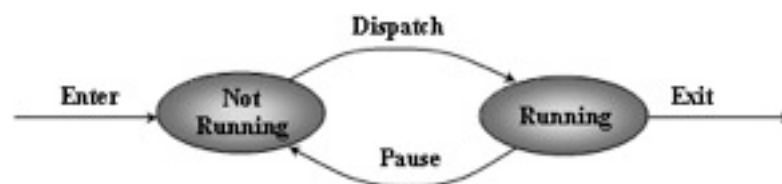
Ukončení procesu

- úloha v dávce vydá příkaz HALT
- uživatel se odhlásí
- proces předá systému požadavek na ukončení
- dojde k chybě

Příčiny ukončení procesu

normální dokončení, vyčerpání časového limitu (Time Out), nedostatek paměti, překročení mezí, porušení ochrany, chybné použití dat, požadavek rodičovského procesu, ukončení rodičovského procesu,...

Dvojstavový model



(a) State transition diagram

Správa proces - typická funkce jádra

- vytváření a ukončování procesů
- plánování a spouštění procesů
- přepínání procesů
- synchronizace procesů
- podpora komunikace procesů
- udržování řídicích bloků procesů

Sdílení prostředků, soupeření procesů, kritická sekce, vzájemné vylučování, deadlock, vyhladovění, hardwarová řešení, semafor, synchronizace a komunikace procesů

Procesy si konkurují při:

- sdílení prostředků
- alokaci času procesoru
- komunikace procesů
- synchronizace více procesů

Soutěžení procesů o prostředky

- provádění jednoho procesu může ovlivnit chování soupeřících procesů
- pokud přístup k jednomu prostředku požaduje více procesů současně, lze prostředek přidělit jen jednomu z nich
- nelze vždy zaručit, že blokový proces získá přístup k prostředku v budoucnu

Kritická sekce - problémy s konkurencí

- pokud přístup ke sdíleným prostředkům není řízen, může dojít k porušení konzistence dat
- výsledek operací prováděných procesy může být závislý na pořadí, v jakém procesy dostanou přidělen procesor - nepřijatelné
- přístup ke sdíleným prostředkům je nutné řídit

Sdílení prostředků - problémy s řízením přístupu

- *vzájemné vylučování* - v každém okamžiku má přístup k prostředkům pouze jeden proces
- *deadlock* - procesy vzájemně čekají na uvolnění prostředků přidělených jiným procesům, odstranění deadlock - násilné ukončení zablokovaných procesů, odebrání prostředků procesům, obnova ze zálohy; vyloučení - nespustit procesy s požadavky, které mohou vyvolat deadlock
- *vyhladovění* - proces nemůže získat přístup k požadovanému prostředku v důsledku obsazení prostředku jinými procesy

Kooperace procesů sdílením

- procesy používají a mění sdílená data
- v daném čase může data změnit pouze jeden proces
- k zajištění integrity se používají kritické sekce

Kritická sekce

- proces je ve své kritické sekci, jestliže provádí část programu, manipulující se sdíleným prostředkem
- provádění kritické sekce musí být vzájemně vylučné
 - v každém okamžiku smí být v kritické sekci pouze jeden proces
- každý proces musí žádat o vstup do kritické sekce

Kooperace procesů komunikací - synchronizace

- vzájemná komunikace procesům umožňuje synchronizovat různé aktivity
- může nastat smrtící objekt
 - každý proces čeká na zprávu od některého jiného procesu
- může nastat vyhladovění
 - dva procesy si stále vyměňují zprávy, zatímco jiný proces čeká na zprávu

Hardwarová řešení

opírá se o některé základní principy

- proces běží v procesoru kontinuálně dokud nevyvolá službu OS nebo není přerušen
- k přerušení procesu může dojít pouze na hranicích instrukcí - mezi dokončením jedné a zahájením další instrukce
- přístup k paměťovému místu obvykle vylučuje současný přístup ostatních procesů k tomuto místu

- Zákaz přerušení

- Speciální instrukce

- Instrukce test-and-set
 - nastavený příznak - kritická sekce je obsazena; aktivní čekání; může dojít k vyhladovění i k deadlock

- Semaforey
 - semaforey jsou výkonným nástrojem pro řešení vzájemného vylučování a koordinace procesů
 - synchronizační nástroj nevyžadující aktivní čekání - proces, který musí čekat je blokován a zařazen do fronty procesů čekajících na stejnou událost - změnu stavu příslušného semaforu - wait/signal

Dlouhodobé, střednědobé a krátkodobé plánování procesů, plánovací kritéria, plánovací algoritmy

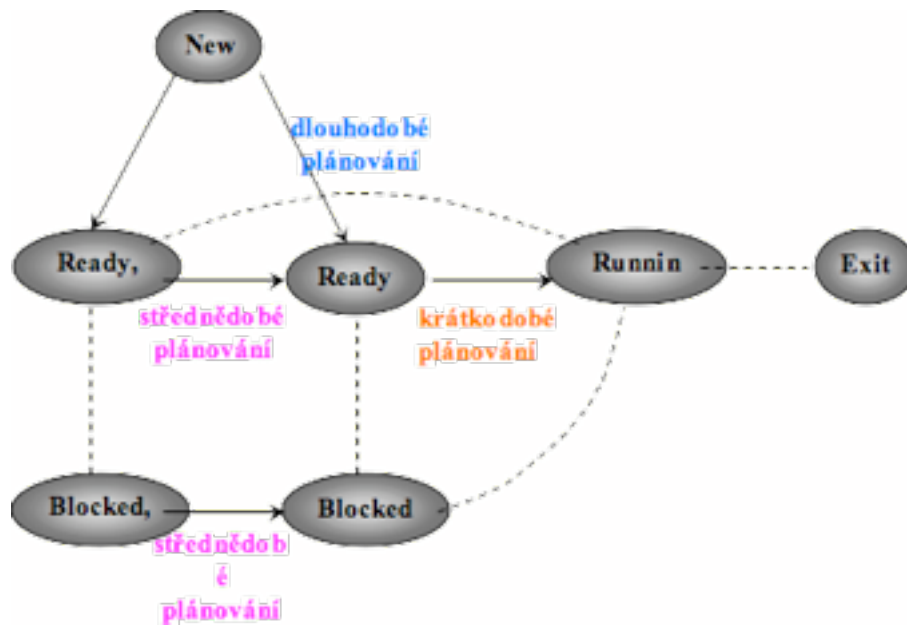
dlouhodobé - provádí se při vytváření nového procesu

střednědobé - odkládání procesů

krátkodobé - který z procesů připravených ke spuštění se má provádět jako další který z požadavků procesů na I/O má být obsloužen dostupným I/O zařízením

Dlouhodobé - určuje, které programy systém zařadí do zpracování, ovládá stupeň multiprogramování, omezuje maximální počet procesů

Čím více procesů, tím méně času procesoru připadá na každý proces, provedení programu trvá déle, v rozporu s požadavkem na vysokou propustnost.



Střednědobé - vychází z potřeb multiprogramování, optimalizuje odkládání procesů (swap), určuje, které procesy mají zůstat v operační paměti a které mají být odloženy.

Maximální počet procesů v systému je omezen velikostí operační paměti - čím více procesů, tím menší riziko, že procesor zůstane nevyužitý (provádí plánovač)

Plánovací kritéria:

Uživatelsky orientovaná kritéria: minimální oba odezvy (doba, která uplyne od předložení požadavku do vydání výstupu)

Systémově orientovaná kritéria: efektivní využitá procesoru, paměti a periférií

Výkonově orientovaná kritéria: výkon měřitelný pomocí doby odezvy a propustnosti, počet provedených úloh za časovou jednotku

Plánovací algoritmy:

Prioritní plánování:

Plánovač vždy dá přednost procesu s vyšší prioritou před procesem s nižší prioritou, obvykle se používá pro každou prioritní úroveň samostatná fronta připravených procesů. Nízká priorita může být příčinou vyhladovění procesu, je nutné umožnit změnu priority čekajících procesů v závislosti na době čekání nebo historii provádění (dynamická fronta)

Rozhodování plánování:

Nepreemptivní - jakmile je proces ve stavu Running, běží, dokud se sám neukončí nebo dokud se nezablokuje čekáním na I/O

- FCFS - Fronta
- SPN - nejprve jsou nejkratší procesy, pak dlouhé

Operační systémy

Preemptivní - OS může právě běžící proces kdykoliv přerušit a přesunout do stavu Ready, umožňuje lepší ovládání, protože jeden proces nemůže na dlouhou dobu monopolně obsadit procesor

- RR - na principu časovače
- SRT - preemptivní varianta SPN - vyžaduje odhad doby provádění

Spravedlivé plánování

Správa paměti, relopace programů, alokace paměti, odkládání obsahu paměti, virtuální paměť, stránkování

rozdělení paměti pro použití více procesy, paměť je nutné alokovat efektivně, aby mohlo být spuštěno co nejvíce procesů, procesy musí mít přístup pouze do té části paměti, která jim byla přidělena

Relokace (přemístění) programu programátor neví, ve které oblasti operační paměti bude program umístěn při provádění, program může být pozastaven, odložen na disk, a pak vrácen do jiné oblasti operační paměti, paměťové odkazy v programu musí být převedeny z logických adres (uváděných v kódu programu) na aktuální adresy fyzické operační paměti, aktuální umístění programu v paměti je určeno při zavedení programu proces může být umístěn v různých dílech, a proto se mohou absolutní adresy během provádění měnit, když se novému procesu nebo procesu odloženému na disk přiděluje paměť, když se provede setřesení paměti

Ochrana - proces nesmí být schopen pracovat s paměťovými místy jiného procesu bez povolení

Sdílení - někdy je nutné dovolit více procesům přístup ke stejné oblasti paměti, je výhodné, jestliže procesy provádějící tentýž program používají tutéž kopii programu v paměti, výrazná úspora paměti, méně časté odkládání na disk

Pevné dělení: dostupná paměť je rozdělena na oddíly s pevnými hranicemi
všechny oddíly jsou stejně velké

- jakýkoliv proces vyžadující oblast paměti menší nebo rovnou velikosti oddílu lze zavést do kteréhokoliv volného oddílu
- když jsou obsazeny všechny oddíly, může OS některé oddíly uvolnit odložením procesu na disk
- pokud se program nevejde do jednoho oddílu, musí programátor použít překrývání
- jakkoliv krátký program obsadí celý paměťový oddíl
- operační paměť nelze využívat efektivně - vnitřní fragmentace

oddíly mají různou velikost

- procesům se přiděluje paměť tak, aby se minimalizovala *vnitřní fragmentace*
- procesům se přiděluje nejmenší oddíl, do kterého se proces ještě vejde
- procesy čekající na přidělení paměti tvoří frontu
 - Samostatná fronta pro každý oddíl
 - proces lze zavést do paměti pouze když je příslušný oddíl volný, zbytečně blokuje procesy, které by se mohly provádět ve větších právě volných oddílech

Společná fronta pro všechny procesy

- při zavádění procesu do paměti je použit nejmenší volný oddíl, do kterého se proces ještě vejde ale nemusí to být nejmenší použitelný oddíl rychlejší ale za cenu větší fragmentace

Dynamické dělení paměti: proměnný počet i velikost oddílů, procesu je přiděleno přesně tolik paměti, kolik potřebuje, při ukončení nebo odložení procesu vznikají v paměti díry uvolněnou část lze použít pouze pro proces, který potřebuje stejný nebo menší objem paměti, při použití procesem s menšími požadavky na paměť představuje zbytek nepoužitá paměť opět díru menší než původní, a proto obtížněji využitelnou, tomuto jevu říkáme *vnější fragmentace*, lze odstranit defragmentací (sražení všech oblastí obsazených procesy těsně k sobě, aby vznikla souvislá oblast paměti)

Umísťovací algoritmy:

algoritmus nejlépe padnoucí (Best-fit) vyber blok, jehož velikost je nejbližší požadavku stejný nebo nejbližší, nejméně výkonná metoda, dosahuje se nejmenší možná fragmentace, protože je vždy použit nejmenší vyhovující blok fragmenty jsou malé, ale rychle přibývají proto se musí často provádět setřesení obsazené paměti

algoritmus první padnoucí (First-fit) paměť se prohledává od začátku a procesu se přidělí první volný blok, který vyhovuje požadavku, rychlejší než best-fit, prohledávání zpomaluje výskyt velkého počtu obsazených bloků, tato oblast se pokaždé zbytečně prohledává

algoritmus další padnoucí (Next-fit) paměť se prohledává od místa, kam se naposledy umísťovalo, a procesu se přidělí první volný blok, který vyhovuje požadavku, umísťuje další proces hned za předchozí, blok volné paměti na konci lze získat setřesením paměti

Logická adresa - odkaz na paměťové místo nezávislý na aktuálním přidělení paměti k získání fyzické adresy je nutný překlad

Relativní adresa - vyjádřená jako relativní pozice vzhledem k nějakému známému místu

Fyzická adresa absolutní nebo aktuální adresa - absolutní adresa umožňuje trvale přístup k určitému místu fyzické paměti, aktuální adresa umožňuje přístup k určitému místu fyzické paměti k získání

Virtuální pam

logický paměťový prostor, jehož část je uložena v části operační paměti a zbytek v sekundární paměti

- odkazy na paměť v programu adresují virtuální paměť
- v reálné operační paměti se nacházejí pouze
 - právě prováděná část programu
 - data, se kterými se právě pracuje
- ostatní části programu a dat mohou být uloženy v sekundární paměti - odkládací oblast
- přístup programu k virtuální paměti je zcela transparentní
 - všechny problémy spojené s vyhledáním aktuálního umístění instrukcí a dat řeší OS
- virtuální paměť může být uspořádána jako
 - lineární adresový prostor - jednorozměrné pole

- množina segmentů - bloky pevné nebo proměnné délky
- mapování adres virtuální paměti na adresy reálné paměti zajišťuje hardware
- pokud program adresuje místo, které se právě nenachází v reálné operační paměti, pak pokud v paměti není volné místo, uloží se obsah bloku reálné paměti na disk
- do volné oblasti reálné paměti se z disku načte blok obsahující požadované paměťové místo

Stránkování

paměť se rozdělí na malé části stejné velikosti - rámce

logická paměťová oblast procesu se rozdělí na stejně velké části jako paměť - stránky

operační systém udržuje pro každý proces tabulku přiřazení stránek a rámců, tabulka obsahuje údaje o umístění rámce v paměti

Segmentace

všechny segmenty všech programů nemusí mít stejnou délku, maximální délka segmentu je omezená, protože segmenty nemají stejnou délku, podobá se segmentace dynamickému dělení paměti

- odkazy na paměť se dynamicky převádějí na fyzické adresy při provádění
 - proces může být odložen a znovu vrácen do paměti, takže jeho umístění v paměti se může kdykoliv změnit
- proces může být rozdělen na části, které nemusí být v paměti uloženy souvisle
 - během provádění se v operační paměti nemusí nacházet všechny části procesu současně
- OS zavede do operační paměti několik částí programu
 - část procesu, která se nachází v operační paměti, označujeme Resident set
- pokud se logicky adresované paměťové místo nenachází v operační paměti, generuje se přerušení
 - OS převede proces do blokovaného stavu
- část procesu obsluhující požadované paměťové místo se přesune do operační paměti
 - OS vydá požadavek na čtení z disku
 - během provádění diskové operace je naplánován další proces
 - dokončení čtení z disku je ohlášeno přerušením, jehož důsledkem je převedení původního procesu do stavu Ready

Stránky

- čím menší stránky, tím menší vnitřní fragmentace, tím více stránek procesy potřebují
- čím více stránek na proces, tím rozsáhlejší jsou stránkovací tabulky
- čím větší stránkovací tabulky, tím větší podíl virtuální paměti tabulky zabírají
- sekundární paměť je navržena pro efektivní přenos dat po blocích, takže větší stránky jsou výhodnější
- možnost práce se stránkami rozdílné velikosti poskytuje pružnost pro efektivní využití TLB
- velké stránky mohou být využity pro kód programu
- malé stránky mohou být použity pro vlákna

Soubor, atributy souboru, omezení přístupu k souboru, Hierarchická organizace souborů, alokace prostoru na disku, souborový systém

Soubory: univerzální forma dlouhodobého uložení dat v sekundární paměti, má unikátní jméno, přístup k souboru může být omezený

Stromová struktura:

- jeden hlavní adresář, obsahující adresáře, podadresáře,...
- počet úrovní může být omezen (např. ISO9660)
- více souborů může mít stejné jméno, pokud se liší cestou
- při práci se soubory lze zvolit libovolný adresář jako pracovní
- cestu k souboru lze vyjádřit relativně vzhledem k pracovnímu adresáři

Přístupová práva:

- *žádná* - uživatel neví o existenci souboru, není dovoleno číst obsah adresáře
- *znalost existence* - uživatel zjistí existenci souboru a jeho vlastníka
- *provádění* - uživatel smí soubor provádět, ale nesmí ho kopírovat
- *tení* - uživatel může obsah souboru číst pro jakékoliv účely (provádění, výpis, kopírování), ale nesmí ho modifikovat nebo mazat
- *přidávání* - uživatel může soubor číst a navíc smí přidat data na konec
- *přepisování* - uživatel může soubor číst/provádět/modifikovat/smazat
- *změna ochrany* - uživatel může měnit nastavení přístupových práv souboru jiného uživatele
- *mazání* - odstranění souboru

Přidělování paměti:

- v době vytváření souboru je nutné znát maximální velikost souboru (je to obtížné)
- ve snaze vyhnout se vyčerpání rezervovaného prostoru je odhad obvykle zbytečně velký
- souvislé obsazení paměťového prostoru zvyšuje rychlost zpracování
- velký počet malých bloků zvyšuje požadavky na velikost tabulek
- pevná velikost bloků zjednodušuje novou alokaci uvolněného prostoru - po smazání souboru lze uvolněné bloky bez problémů použít pro nový soubor
- proměnná velikost bloků - minimalizuje vnitřní fragmentaci
- **souvislá alokace**
 - při vytváření je souboru přidělena jedna sada po sobě následujících bloků
 - stačí jedna položka v alokační tabulce - počáteční blok souboru, délka souboru
 - dochází k vnější fragmentaci
 - po zrušení souboru se obvykle nepodaří do uvolněné oblasti umístit soubor stejné velikosti
 - po zaplnění disku je obtížné nalézt souvislou sadu bloků potřebné délky
- **řetěžená alokace**
 - alokují se jednotlivé bloky
 - každý blok obsahuje ukazatel na následující blok v řetězci
 - stačí jedna položka v alokační tabulce - počáteční blok souboru, délka souboru
 - nedochází k vnější fragmentaci
 - kterýkoliv volný blok lze připojit na konec souboru
 - v případě potřeby lze snadno zvětšit velikost souboru
 - nebere ohled na princip lokality
 - logicky sousední bloky mohou být fyzicky umístěny značně daleko od sebe

• **indexová alokace**

- alokační tabulka obsahuje samostatný jednoúrovňový index pro každý soubor
- index má jednu položku pro každý blok alokovaný souboru
- alokační tabulka obsahuje číslo bloku pro indexaci
- lze použít proměnnou velikost bloků

Souborový systém

- slouží k dlouhodobému uložení programů a dat
- informace se ukládají v pojmenovaných objektech zvaných soubory
 - pro OS snadno použitelná jednotka z hlediska přístupu a ochrany
- programy při manipulaci s obsahem souborů mohou soubory (či jejich části) kopírovat do virtuální paměti
 - soubor může sloužit jako zdroj vstupních dat nebo cíl vstupních dat pro program
 - soubor může obsahovat program

Různé koncepce OS, monolitický systém, vrstvený systém, mikrojádro, vlákna, OS pro práci v reálném čase, distribuovaný OS, multiprocesorové systémy

Monolitický operační systém

obsahuje všechny funkce jádra v sobě. Velké neschopné přenosné nerozšiřitelné.

Mezi výhody patří centralizovaný charakter, který umožňoval relativně snadnou administraci systému a poměrně snadnou implementaci aplikací. Nevýhodou byl často monolitický software - uživatelské rozhraní, výkonná část a přístup k datům byly obsaženy v jedné velké aplikaci, a proto jakákoliv změna mohla ovlivnit celou aplikaci.

Vrstvený systém

- vzhledem k velké složitosti OS obvykle systém dělíme na vrstvy
- dělení na vrstvy umožňuje rozklad složitých problémů na větší počet snadněji zvládnutelných dílčích problémů
- každá vrstva využívá (primitivnějších) funkcí následující nižší vrstvy

Mikrojádro

- jádro OS plní pouze malý počet základních funkcí
 - primitivní řízení paměti
 - komunikace procesů (Inter-Process Communication)
 - základní plánování
- ostatní služby OS jsou poskytovány procesy běžícími v uživatelském režimu (servery)
 - ovladače zařízení, souborové systémy, virtuální paměť
- výhody: větší pružnost, rozšiřitelnost, přenositelnost
- nevýhody: snížení výkonnosti v důsledku náhrady systémových volání výměnnou zpráv mezi procesy

Vlákna

- proces se skládá z jednoho nebo více vláken, která lze provádět souběžně
- užitečné, když se aplikace skládá z několika vzájemně nezávislých akcí (nemusí být prováděny po sobě)
- všechna vlákna jednoho procesu sdílejí data, prostředky a část kontextu procesu
- je snadnější vytvořit vlákno, zrušit vlákno nebo přejít z jednoho vlákna do druhého, než udělat totéž s procesy

Operační systémy

- Operační systém může podporovat provádění více vláken v jednom procesu
 - MS-DOS - jednovláknové procesy
 - UNIX - více uživatelských procesů (multitasking), procesy jsou pouze jednovláknové
 - Windows NT - vícevláknové procesy

OS pro práci v reálném čase

Real Time Operating Systém - RTOS je charakterizován deterministickým chováním, krátkou dobou odezvy, vysokou spolehlivostí

- Deterministické chování
 - operace se provádějí v pevných, předem určených časech nebo v předem určených časových intervalech
 - u každé operace je předem známo, kdy nejpozději skončí
- Doba odezvy
 - čas, ve kterém musí systém přiměřeně reagovat na událost, časové měřítko je relativní
 - závisí na aplikační oblasti stovky mikrosekund např. řízení reaktoru
- Spolehlivost
 - systém, jehož porucha může mít katastrofální důsledky
 - systém natolik spolehlivý a bezpečný, že na něm můžeme být zcela závislí
 - systém odolný proti poruchám, porucha může snížit výkonnost systému, ale nesmí ho vyřadit z funkce, přednost mají úlohy kritické pro funkci systému, úlohy s nižší prioritou se provádějí jen když na ně zbývá čas.

Distribuovaný operační systém

jako NFS, umožňuje vzdálené připojování pomocí mount, systém klient-server

Tento model představuje aplikaci jako objekty, které mohou používat služby nabízené jinými objekty v systému nebo samy tyto služby nabízet. Distribuované systémy jsou v podstatě vícevrstevné klient/server systémy, kde počet různých klientů a serverů může být obrovský.

Multiprocesorové systémy

Single Instruction Single Data; Single Instruction Multiple Data; Multiple Instruction Single Data

Multiple Instruction Multiple Data - kombinace procesorů, které souběžně zpracovávají odlišnými posloupnostmi instrukcí odlišné množiny dat

podle stupně vazby

MP systém s volnou vazbou

- každý procesor má vlastní operační paměť a I/O subsystém
- různé typy vazby

MP systém s těsnou vazbou

- procesory sdílejí operační paměť
- řízení jedním operačním systémem

podle symetrie

symetrický MP systém

- jádro OS může provádět kterýkoliv procesor

Operační systémy

- spustitelné procesy či vlákna lze provádět v kterémkoliv volném procesoru
- asymetrický MP systém
- procesory jsou funkčně specializované např. I/O procesory
- systém je řízen centrálním procesorem