Operační systémy – přehled

PB 152 ◇ Operační systémy

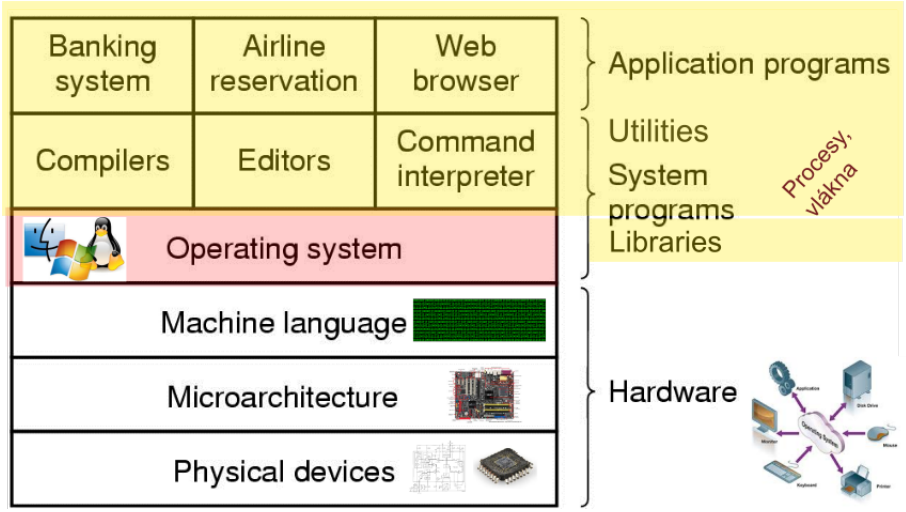
Jan Staudek
http://www.fi.muni.cz/usr/staudek/vyuka/

Verze: jaro 2017

Komponenty počítačového systému

- uživatelé (lidé, stroje, jiné počítače, ...)
- aplikační programy
  - ✓ definují způsoby kterými se používají zdroje systému pro řešení výpočetních uživatelských problémů (vědecké úlohy, video hry, byznys programy, ...)
- systémové programy, pomocné programy, knihovní programy
  - ✓ řeší standardní, systémově orientované úlohy (zálohování, editace, kompilace, řízení databáze, standardní matematika, ...)
- operační systém
  - ✓ program pro řízení a správu používání hardware různými aplikačními výpočty různých uživatelů – souběžně, v reálném čase, ...
- hardware
  - ✓ bázové výpočetní zdroje (CPU / procesor, paměť, I/O zařízení komunikační spoje), CPU – *Central Processing Unit*

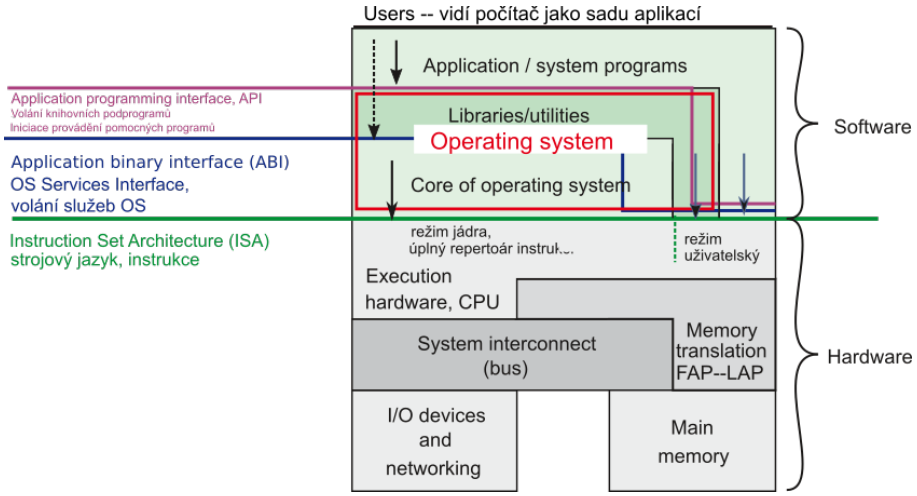
Komponenty počítačového systému, hierarchie



Operační systém, cíle a funkce

- Program, který řídí provádění aplikačních a systémových programů v počítači a efektivně jim zpřístupňuje zdroje
- Reprezentuje rozhraní
  aplikační systémy a systémové programy × hardware
- Cíle OS
  - ✓ pohodlnost používání počítače
  - ✓ efektivnost využívání zdrojů počítačového systému
  - ✓ schopnost rozvoje – umožnit efektivní vývoj, testování a zavádění nových funkcí bez interference s poskytovanými službami
- Z cílů plynou role
  - ✓ OS – rozhraní uživatel (aplikační a systémové procesy)/počítač
  - ✓ OS – správce zdrojů
  - ✓ OS – vývojeschopný organismus

Struktura hardware a software počítače detailněji



Generická rozhraní OS

- Instruction set architecture (ISA), strojový jazyk
- Application programming interface (API),
  knihovny volání služeb OS na úrovni vyšších programovacích jazyků, pomocné programy
  – OS může poskytovat pro různé třídy aplikací různá API
  sockets pro přenos dat, .NET, transakce, ...
- Application binary interface (ABI),
  volání služeb OS na úrovní strojového jazyka (asembleru), reprezentace OS a počítače vůči všem aplikacím

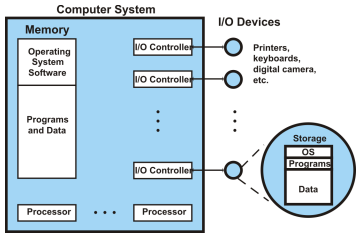
OS v roli rozhraní uživatel/počítač

- Koncoví uživatelé vidí počítač jako sestavu aplikačních programů
  - ✓ Aplikační programy se píší v programovacích jazycích aplikačními programátory
- Pro složité ovládání počítače mají k dispozici funkčnost součástí operačního systému – systémových, knihovních, pomocných programů, *utilities*
- Maskování detailů hardware a funkčnost nutnou pro efektivní a pohodlné používání počítače aplikačními, systémovými, knihovními, pomocnými programy zajišťují programy soustředěné v jádru operačním systému, *core*, *kernel*, pomocí poskytovaných služeb OS, *OS services*

Typické oblasti služeb poskytovaných OS

- Vývoj programů, editory, ladící systémy, ...
  typicky poskytované pomocnými (systémovými) programy
- Provádění programů, vše co je nutné zajistit pro činnosti řízené programy – plánování, zavádění, ovládání IO, ...
- Přístup k IO zařízením – jednotné API pro různá zařízení
- Přístup k souborům dat na vnějších pamětech
- Přístup k systémovým zdrojům, bezpečnost, řešení konfliktů
- Chybové řízení, automatizované reakce na nestandardní stavy v hardware, v software a v případech kdy OS nemůže uspokojit požadavek aplikace
- Protokolování, info o tom co se dělo, základ pro účtování, základ pro odhady budoucího vylepšování, ...

OS v roli správce zdrojů



- ✓ Počítač je sestavou zdrojů / prostředků pro přesun, uchovávání a zpracování dat
- ✓ OS je odpovědný za řádnou správu těchto zdrojů
- ✓ OS je implementovaný v software,
  procesor ho chápe jako každý jiný řádný software

  OS je suita programů prováděných procesorem
- ✓ Programy soustředěné v jádru OS se organizačně neřeší formou procesů, jsou spíše nástrojem pro poskytování služeb procesům

OS je vývojeschopný organismus

- V průběhu života konkrétního konceptu OS dochází
  - ✓ k doplňování a k inovacím hardware
  - ✓ k doplňování nově požadovaných služeb
  - ✓ k opravám chyb

Neexistuje univerzálně akceptovatelná definice ,,co to je OS"

- OS je program, který funguje jako spojka mezi uživatelem počítače a hardware počítače –
  OS = cokoliv co je činné ve výpočetním systému a není to hardware nebo aplikace
- neexistuje univerzálně akceptovatelná definice ,,co to je OS"
  - ✓ *Everything a vendor ships when you order an operating system*
- *The one program running at all times on the computer is the kernel. Everything else is either a system program (ships with the operating system) or an application program*

Požadavky na OS jsou různorodé

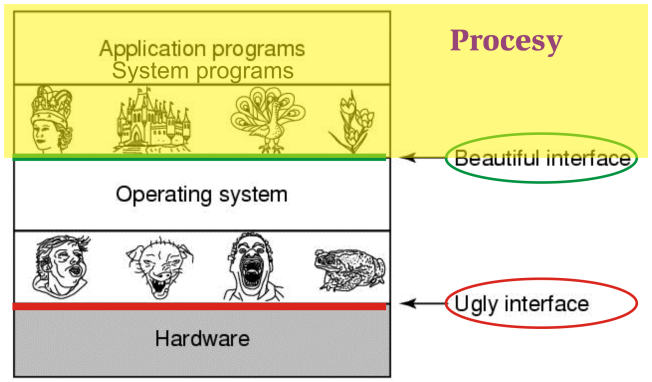
- Cíle (povinnosti) OS
  - ✓ řídit řešení uživatelských (aplikačních) programů
  - ✓ poskytnout nástroje pro řešení problémů uživatelů (aplikací)
  - ✓ učinit počítač snadněji použitelný
  - ✓ vytvářet podmínky umožňující efektivně používat hardware počítače
- Cíle (přání) uživatele
  - ✓ služby poskytované OS lze pohodlně používat, snadno zvládnout
  - ✓ OS je spolehlivý, bezpečný
  - ✓ požadované služby poskytuje OS pohotově
- Cíle (přání) provozovatele OS
  - ✓ OS je snadno navrhnutelný, implementovatelný a udržovatelný,,
  - ✓ OS je přizpůsobitelný, spolehlivý a bezchybný

Co to je operační systém?

- OS je poskytovatel problémově orientované abstrakce bázových fyzických prostředků
- bázové fyzické prostředky:
  - procesory, operační paměť, komunikační nástroje, vnější paměti, . . .
- OS nabízí programátorovi bázové fyzické prostředky k použití formou služeb poskytovaných na rozhraní volání systému
- programátor chce vidět spíše
  - soubory a záznamy než diskové bloky a vystavovací mechanismus disku
  - spíše schránky (*sockets*) než přímý přístup k síti
  - spíše procesy a vlákna než procesory, paměťový prostor

Cílem použití OS je abstrakce, extended machine

- Operační systém mění ošklivý hardware na atraktivní abstrakci



Co to tedy je operační systém?

- OS je správce prostředků
  - ✓ spravuje a řídí využívání všech zdrojů systému, eviduje jejich využívání,. . .
  - ✓ rozhoduje mezi konfliktními požadavky tak, aby používání zdrojů systému bylo efektivní a spravedlivé (odpovídající zvolené politice)
- OS je řídicí program,
  - ✓ řídí provádění ostatních programů tak, aby zabraňoval chybnému a nepatřičnému použití počítače
  - ✓ řídí bezpečné provádění uživatelských programů a operací I/O zařízení

Problémy budování OS

- OS jsou ,,obrovské"systémy
  - ✓ v současnosti představují až stovky miliónů řádků kódu,
  - ✓ pracnost řádově tisíce člověko-roků
- OS jsou složité systémy
  - ✓ požadavky různých uživatelů se často podstatně liší
  - ✓ nelze jednorázově odstranit všechny chyby, verifikace z důvodů složitosti selhává
- Chování OS se obtížně předpovídá, ,,seřizování"/ ladění se dělá vesměs odhadem

OS se staly složité – funkčně

✓ system calls: *open, read, write, close, wait, exec, fork, exit, kill ...*

OS	rok	počet
Unix	1971	33
Unix	1979	47
Windows 1.0	1985	450
SunOS 4.1	1989	171
4.3 BSD	1991	136
SunOS 4.5	1992	219
SunOS 5.6	1997	190
Linux 2.0	1998	319
FreeBSD	1998	330
Windows	1999 a dál	$x \times 10^3,\ x > 3$

OS se stávají složité – funkčně

- Enormě narůstá složitost vnitřních algoritmů (jádra) OS
- počty cyklů procesoru spotřebovaných ve Windows NT při
  - ✓ Zaslání zprávy mezi procesy: 6K – 120 K podle použité metody
  - ✓ Vyvtoření procesu: 3M
  - ✓ Vytvoření vlákna: 100K
  - ✓ Vytvoření souboru: 60K
  - ✓ Vytvoření semaforu: 10K – 30K
  - ✓ Nahrání DLL knihovny: 3M
  - ✓ Obsluha přerušení/výjimky: 100K – 2M
  - ✓ Přístup do systémové databáze *Registry*: 20K
  - ✓ Windows NT (New Technology) – původní systém fy Microsoft pro procesory IA-32 a novější
  - ✓ Windows 7, 8 ... jsou pouze obchodní názvy různých verzí NT
  - ✓ V současné době podporované verze NT 6.2 a NT 6.3

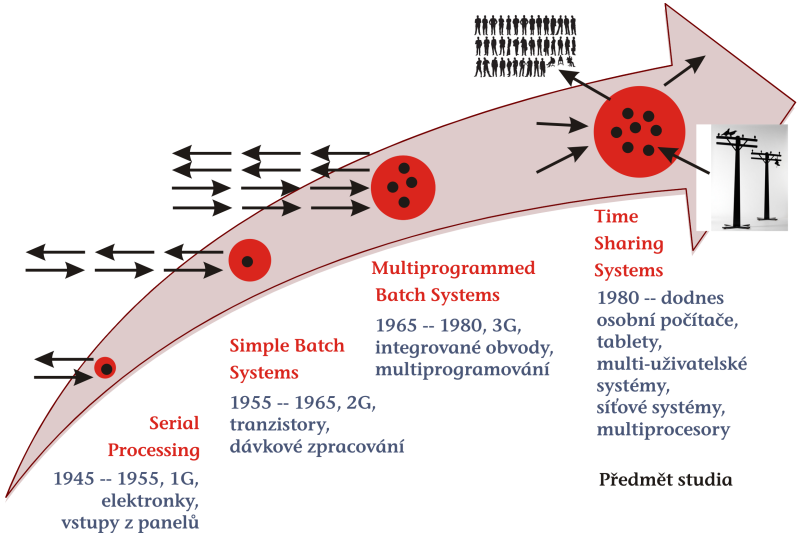
OS se stávají složité – rozsahem (LOC, Lines of Code, v 10^6)

OS	rok	LOC
3.1	1992	3
NT	1992	4
95	1995	15
NT 4.0	1996	16
98	1998	18
2000	2000	30
XP	2001	40
Vista	2007	64
7	2009	??

OS	rok	LOC
Solaris	1991	10
FreeBSD	1993-1998	9
Red Hat Linux 6.2	2000	17
Red Hat Linux 7.1	2001	30
Debian 2.2	2000	60
Debian 3.0	2002	104
Debian 3.1	2005	215
Debian 4.0	2007	283
Fedora 9	2008	205
Mac OS X 10	2010	86

- 2 dekády – zvýšení rozsahu OS v LOC na dvaceti až 30-násobek

Evoluce OS v bodech

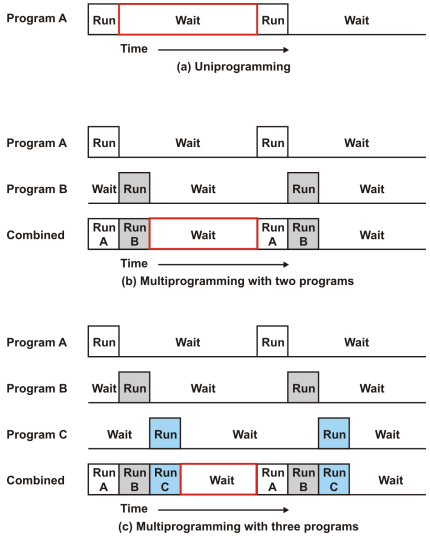


Klasifikace počítačů – klasické (střediskové) počítače

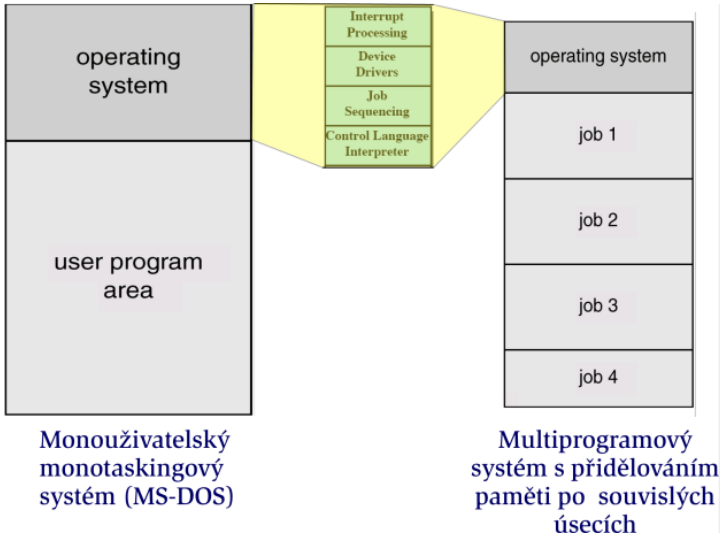
- střediskový počítač – dnes již historický pojem
- v současnosti vystupují v jejich rolích podnikové servery
- klasické hlavní rysy střediskových počítačů
  - ✓ redukce režijního času pro přípravu výpočtů se dosahovala řazením podobných prací / zakázek (*jobs*) do dávek – batch,
  - ✓ batch processing, automatizace řazení dávek vč. automaticky předávaného řízení mezi definovanými zakázkami (*jobs*)
  - ✓ rezidentní řídicí program – *monitor* – počátečně získává řízení, řízení předává mezi zakázkami, když zakázky končí – řízení se vrací monitoru
  - ✓ multiprogramový režim činnosti – souběžné řešení více programů

    jestliže čtení a zápis dat z/na disku trvá 15 μs, zpracování dat trvá 1μs, pak je procesor využitý pouze na 3,2 % (1/32) a jeho kapacitu v době čekání na IO leží ladem

Princip multiprogramování (multitasking)



Organizace hlavní paměti klasického počítače



Rysy OS potřebné pro implementaci multiprogramování

- ex. funkcionalita pro ovládání I/O a rysů nutných pro implementaci OS (viz níže)
  - ✓ vlastní řízení I/O operací provádí výhradně operační systém
  - ✓ 2-režimový provoz CPU (jádro operačního systému x uživatel),
  - ✓ privilegované / neprivilegované instrukce
- ex. funkcionalita pro správu paměti
  - ✓ Systém musí být schopný přidělovat paměť různým zakázkám a ze zakázek odvozeným procesům dynamicky
  - ✓ Dvojí vize paměti
    – z hlediska její fyzické konstrukce a šířky fyzických adresovacích registrů – fyzický adresový prostor, FAP
    – z hlediska konstrukce adresy ve strojovém jazyku logický adresový prostor, LAP
  - ✓ záruka ochrany oblastí paměti před neautorizovaným přístupem

Rysy OS potřebné pro implementaci multiprogramování

- ex. mechanismus přerušení
  - ✓ předávání řízení mezi aplikačním programem a jádrem OS – reakce na asynchronní události
- ex. funkcionalita pro plánování CPU
  - ✓ „srdcem" typicky bývá intervalový časovač
  - ✓ zabraňuje se, aby si uživatelský proces trvale obsadil CPU (záměrně, chybou, ...)
  - ✓ po uplynutí intervalu času se generuje přerušení
  - ✓ OS musí být schopný volit mezi různými procesy připravenými k provedení
- ex. funkcionalita pro přidělování zařízení (zdrojů)
  - ✓ Dynamicky, exklusivně / sdíleně

Základní definice související s OS

- jádro OS
  - ✓ logické rozšíření rysů hardware + poskytované služby
  - ✓ vše mimo jádro je řešeno formou procesů
  - ✓ jádro může využívat speciální rysy hardware nedostupné procesům
- mikrojádro OS
  - ✓ minimalistická varianta jádra OS použitá v některých OS
  - ✓ typicky zabezpečuje
    - správu přerušení,
    - správu paměti,
    - správu procesorů a
    - správu procesů a komunikaci mezi procesy předáváním zpráv – *Interprocess communication* (*IPC*)
  - ✓ ostatní funkce jádra se přesouvají do „procesové" oblasti (drivery, služby systému souborů, virtualizace paměti, ...)
  - ✓ mezi procesy se komunikuje předáváním zpráv

Hlavní přínosy z vývoje OS pro počítačové vědy

- Procesy
- Správa paměti
- Ochrana informací a bezpečnost
  - ✓ důvěrnost, *Confidentiality*
  - ✓ integrita, *Integrity*, integrita dat, integrita identity – *authenticity*
  - ✓ dostupnost, *Availability*
- Plánování a správa, řízení zdrojů
  - ✓ spravedlivost, diferenciovatelná přednost, efektivita
  - ✓ krátkodobé – bezprostřední plánování, dlouhodobé – strategické plánování
- Strukturování systémů

Proces

- Identifikovatelná jednotka činnosti charakterizovatelná sekvenčním vláknem (tokem) provádění operací a s ním souvisejících zdrojů
- Program je strukturovaný příkaz, proces je děj
- Program umístěný v paměti může řídit běh více procesů
- Souběžné řešení více procesů se nazývá multiprogramování, resp. multitasking
- Souběžnost řešení procesů může být příčinou časově závislých chyb
  - ✓ Nesprávná synchronizace
  - ✓ Chybné vzájemné vyloučení
  - ✓ Nedeterminismus operací
  - ✓ Uváznutí

Hlavní příčiny časově závislých chyb v procesech

- Nesprávná synchronizace
  - ✓ jeden proces očekává událost, kterou generuje jiný proces (naplnění / vyprázdnění vyrovnávací paměti, . . .)
  - ✓ proces který generuje událost signalizuje její vznik
  - ✓ signály se nesmí ani ztrácet ani duplikovat
- Chybné vzájemné vyloučení
  - ✓ editovat sdílenou datovou strukturu smí v jednom okamžiku pouze jeden proces
  - ✓ souběžné procesy se musí na při takové editaci vzájemně vylučovat, změna stavu sdílené struktury musí proběhnout nedělitelně, atomicky

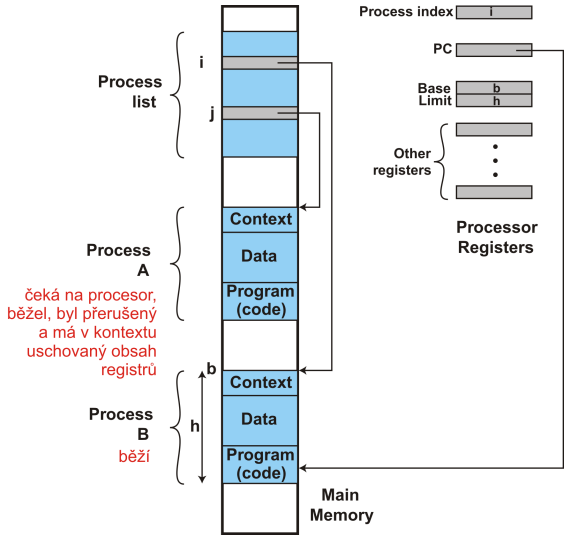
Hlavní příčiny časově závislých chyb v procesech

- Nedeterminismus operací
  - ✓ činnost procesu má záviset pouze na jeho vstupech, nikoli na činnosti jiných procesů
  - ✓ pokud procesy sdílejí paměť, pak při prokládání jejich běhu na procesoru nesmí výstupy ovlivňovat pořadí, ve kterém je plánovaný jejich běh
- Uváznutí
  - ✓ dva nebo více procesů na sebe vzájemně čekají
  - ✓ první proces výlučně drží zdroj A, druhý výlučně drží zdroj B a první požaduje přístup k B a druhý přístup k A

Komponenty procesu, správa procesů, vlákno

- program, který řídí tok operací
- zpracovávaná data
  - ✓ proměnné, pracovní prostory, vyrovnávací paměti, . . .
- kontext provádění – stav procesu
  - ✓ interní data v OS pro dohled na procesem a pro řízení procesu
  - ✓ obrazy registrů, které proces sdílí s ostatními procesy
  - ✓ priorita procesu, zda proces čeká na jistou událost, . . .
- Správa procesů
  - ✓ úplný stav je trvale obsažený v jeho kontextu
- Vlákno, *thread*
  - ✓ proces, který drží zdroje nutné pro jeho realizaci, lze vnitřně rozdělit do souběžně řešitelných dílčích toků – vláken, vlákno běží v kontextu jeho procesu

Typická implementace procesu (ve FAP získává proces souvislý úsek)



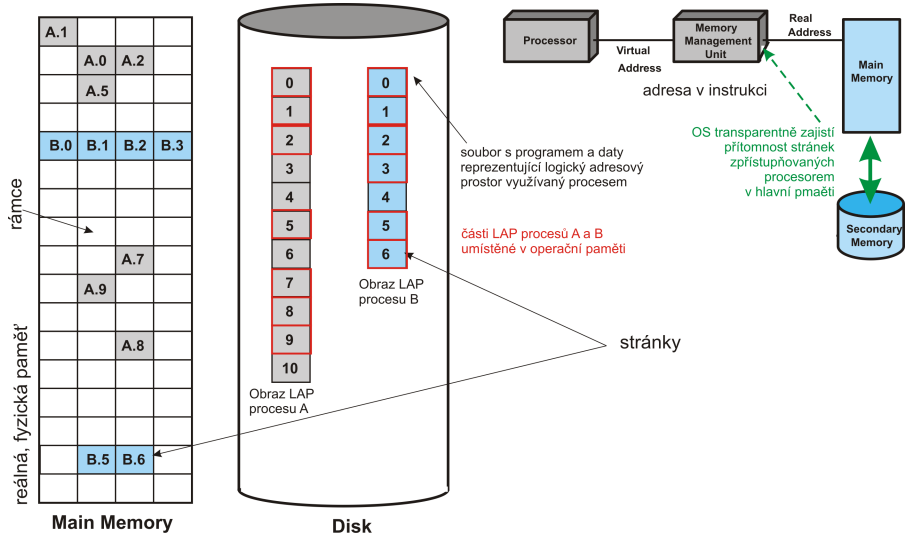
Správa paměti, pět nejdůležitějších požadavků

- Izolace procesů
  - ✓ data i programy různých procesů nesmí v paměti interferovat
- Podpora modulárního programování
- Automatické přidělování a správa
  - ✓ OS procesům přiděluje paměť v celé hierarchii pamětí podle jejich potřeby, z hlediska programátora transparentně,
- Ochrana a řízený přístup k paměti
- Dlouhodobé uchovávání dat
  - ✓ Data se mnohdy musí uchovat i po ukončení procesu, který je vytvořil a nezávisle na dostupnosti energie (na discích, . . .)
  - ✓ Plní společně se součástí OS systém souborů, *File System*

Virtuální paměť

- Procesy mohou adresovat paměť z logického hlediska, bez ohledu dostupnou fyzickou kapacitu hlavní paměti
- Nutná vlastnost pro splnění požadavku souběžného používání hlavní paměti více procesy
- Základni technika – stránkování, *paging*
  - ✓ proces se skládá z bloků pevné délky – stránek, *pages*
  - ✓ program adresuje operandy udáním čísla stránky a adresy operandu ve stránce
  - ✓ každá stránka může být umístěna v hlavní paměti kdekoliv – v rámci, *frame*
  - ✓ stránka zpřístupňovaná procesorem musí být v rámci v hlavní paměti
  - ✓ správa paměti poskytuje mechanismus dynamické transformace logických (virtuálních) adres na reálné (fyzické) adresy v hlavní paměti

Virtuální paměť

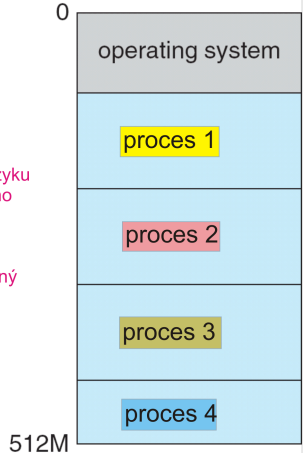


Studujeme OS využívající koncept multiprogramování

- jeden uživatel s jedním řešeným programem nemůže udržet CPU a I/O zařízení trvale v chodu
- co se nepoužívá, to je mrtvá investice
- multiprogramování organizuje řešení více úloh (zadání) formou procesů takovým způsobem, aby CPU byla pokud možno stále činná
  - ✓ proces – kód programu, data, prostor v paměti, udržování identity . . .
  - ✓ prostory v paměti OS přiděluje vybrané podmnožině procesů
  - ✓ každému jednomu procesu OS postupně přiděluje CPU pro jeho běh (řešení)
  - ✓ když běžící proces se rozhodne čekat (např. na dokončení I/O), OS přidělí CPU jinému procesu
  - ✓ když běžící proces usurpuje CPU neúměrně dlouho , OS přidělí CPU jinému procesu

Uspořádání hlavní paměti v multiprogramových systémech

Klasické (dnes již spíše historické) uspořádání hlavní paměti



Problém:

Program řídidící proces 2 byl připravený ve zdrojovém jazyku.

Program ve zdrojovém jazyku byl přeložený do strojového kódu překladačem.

Musí překladač vědělt kde bude program umístěný v paměti počítače ?

Odpověď dá poznání principů činnosti OS a souvisejícího hardware

Řešení problému:

Překladač zná pouze LAP.

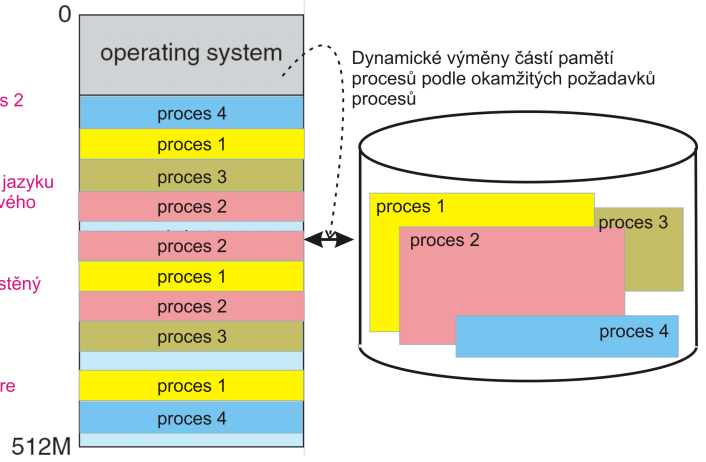
Překladač překládá do LAP.

Program zaváděný do FAP je umístěný ve FAP ve formě připravené pro LAP.

Transformaci LAP do FAP v textu proramu řeší hardware ve spolupráci s OS až při interpretaci instrukce

Uspořádání hlavní paměti v multiprogramových systémech

Řešení ,,problému"se stane složitější při virtualizaci paměti



Problém:

Program řídidící proces 2 byl připravený ve zdrojovém jazyku.

Program ve zdrojovém jazyku byl přeložený do strojového kódu překladačem.

Musí překladač vědělt kde bude program umístěný v paměti počítače ?

Odpověď dá poznání principů činnosti OS a souvisejícího hardware

Multiprogramování = Multitasking

- multitasking je synonymum konceptu multiprogramování
- CPU se předává mezi procesy tak rychle, že uživatelé mohou být se svými souběžně řešenými procesy v on-line interakci, jedná se tak o koncept interaktivního počítání (*interactive computing*)
  - ✓ doba reakce procesu vůči uživateli bývá < 1 sekunda
  - ✓ každý uživatel má v hlavní paměti přidělený prostor alespoň pro jeden prováděný program a nutná data – proces
  - ✓ poněvadž bývá připraveno k běhu více procesů, OS musí řešit plánování činnosti CPU, aby postup procesů splňoval zadaná kritéria
  - ✓ jestliže nelze všechny potřebné procesy umístit do hlavní paměti, lze některé z nich přesunout do vnější paměti a jejich běhy se odložit – technikou nazývanou swapping
  - ✓ virtualizace paměti umožňuje provádění procesů, aniž by byly v hlavní paměti umístěné celé jejich programy a všechna jejich data

Ochrana informací, informační bezpečnost

- Význam roste přechodem na multiuživatelské, multiprogramové, síťově orientované, . . . počítačové systémy
- Hlavní problémy informační bezpečnosti zajišťované OS
  - ✓ Dostupnost (*availability*) – ochrana systému proti přerušení činnosti
  - ✓ Důvěrnost (*confidentiality*) – dostupnost zdrojů je podřízená definovatelné autorizaci aktérů
  - ✓ Integrita – modifikovatelnost zdrojů je podřízená definovatelné autorizaci aktérů
  - ✓ Autenticita – ověřitelnost identity (uživatelů. procesů, dat)

Plánování a správa zdrojů

- Význam roste přechodem na multiuživatelské, multiprogramové, síťově orientované, ... počítačové systémy
- Hlavní cíle plánování a správa zdrojů zajišťované OS
  - ✓ Spravedlivost (*Fairness*) – všichni aktéři mají rovný, spravedlivý přístup ke zdrojům
  - ✓ Diferencovatelná vnímavost (*Differential responsiveness*) – komplement ke spravedlivosti respektující stanovená pravidla
  - ✓ Efektivita (*Efficiency*) – maximalizace propustnosti, minimalizace dob reakcí a přizpůsobivost co nejvíce uživatelům jsou konfliktní kritéria, jejichž naplnění musí OS balancovat