OSTRAVSKÁ UNIVERZITA
PŘÍRODOVĚDECKÁ FAKULTA

Procesy a jejich plánování

Ing. Pavel Smolka, Ph.D.

Program x proces

Program

- Zápis algoritmu v nějakém programovacím jazyce (například ve strojovém kódu). Je statický, neměnný (neuvažujeme-li vývoj nových verzí programů).
- Algoritmizovaný plán činnosti.

Proces (process, task)

- Běžící program. Proces je tvořen neměnným kódem programu a konstantami a proměnnými daty jako je stav procesoru, data na zásobníku, globální proměnné, halda, soubory atd.
- Samotná činnost.

Definice procesu

- Činnosti probíhající v systému mají podobu procesů.
- Proces se provádí vykonáváním posloupností svých instrukcí.
- Každému spuštěnému programu odpovídá jistý proces.
- Způsob přidělování procesů procesoru se nazývá **plánovací strategie**. (= obecněji způsob přidělování prostředků (zdrojů) systému - paměti, procesoru, periférií)
- Uživatel: rychlé střídání procesů vede k **iluzi** jednouživatelského systému.
- **Přepínaní procesů** (střídání procesů na procesoru) = přepínání kontextu (preemptivní x nepreemptivní)

Typy plánování

- **Nepreemptivní plánování** (plánování bez předbíhání, někdy také kooperativní plánování), kdy procesu schopnému dalšího běhu procesor není odnímán.
- **Kooperativní plánování** se používá v „uzavřených systémech“, kde jsou předem známy všechny procesy a jejich vlastnosti. Navíc jsou naprogramovány tak, aby samy uvolňovaly procesor ve prospěch procesů ostatních.
- **Preemptivní plánování** (plánování s předbíháním), kdy procesu schopnému dalšího běhu může být procesor odňat i „bez jeho souhlasu“.

Nepreemptivní plánování

- V případě nepreemptivního plánování se proces musí procesoru sám vzdát.
- Pokud má být doba, po kterou je proces ve stavu běžící, omezená, je nutné, aby proces kontroloval časovač a po překročení stanovené doby se dobrovolně vzdal procesoru vyvoláním služby OS, která je k tomuto účelu určena.
- Výhodou je, že proces nemůže být přerušen, pokud nechce (například v kritické sekci). Nevýhodou je, že špatně chovající se proces může zablokovat celý OS. Takto fungují například MS-Windows.

Preemptivní plánování

- V případě preemptivního plánování OS může odebrat procesu procesor. Zpravidla se tak děje při uplynutí časového kvanta určeného pro běh procesu a celá akce je vyvolána přerušením od časovače. Příkladem OS, který používá preemptivní plánování je OS Unix.

Zdroje systému nutné pro běh procesu

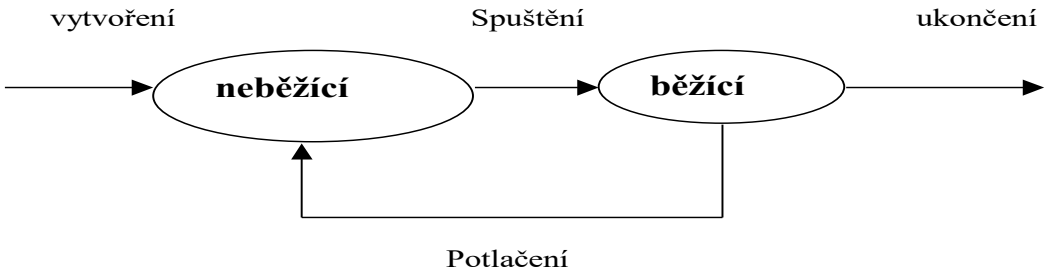
- procesor - čas procesorů
- vnitřní paměť
- další prostředky
  - vnější paměť
  - I/O zařízení
  - datové struktury - soubory apod

Kontext procesů

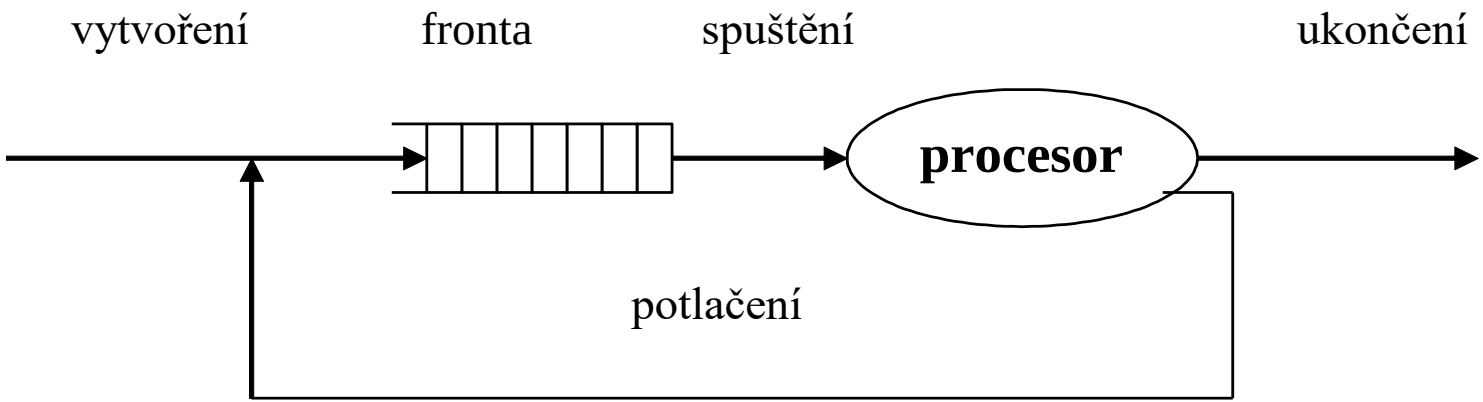
- Proces je definován:
  - textovým a datovým regionem (segmentem),
  - zásobníkovým regionem,
  - datovými strukturami spojenými s procesem (registry, systémové tabulky a záznamy v nich).
- Přechod ke zpracování jiného procesu je označován jako přepnutí kontextu.
- Pro zaručení možnosti návratu k „rozpracovanému“ procesu musí být před přepnutím kontextu uloženo určité kvantum informací - kontext procesu.

Stavy procesu

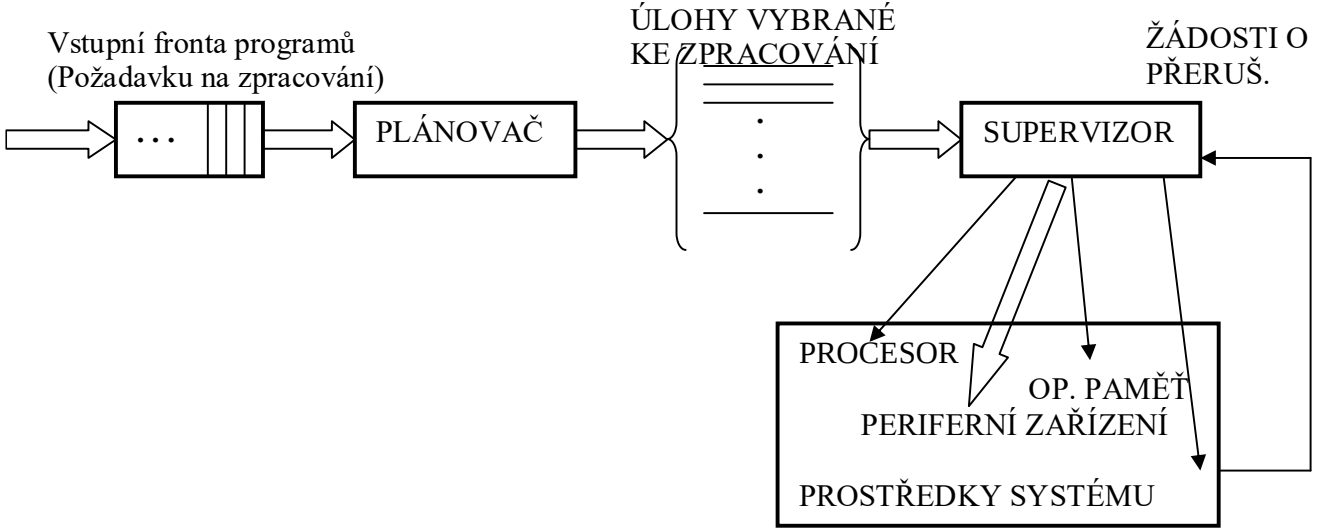
- Proces je dynamická entita – může se nacházet v různých **stavech** a během své existence **přechází** z jednoho stavu do druhého stavu, stavy jsou chápány jako diskrétní stavy.
- Přechod mezi jednotlivými stavy mohou způsobit různé **události**. Přechod je **řízen** jednotlivými částmi operačního systému.



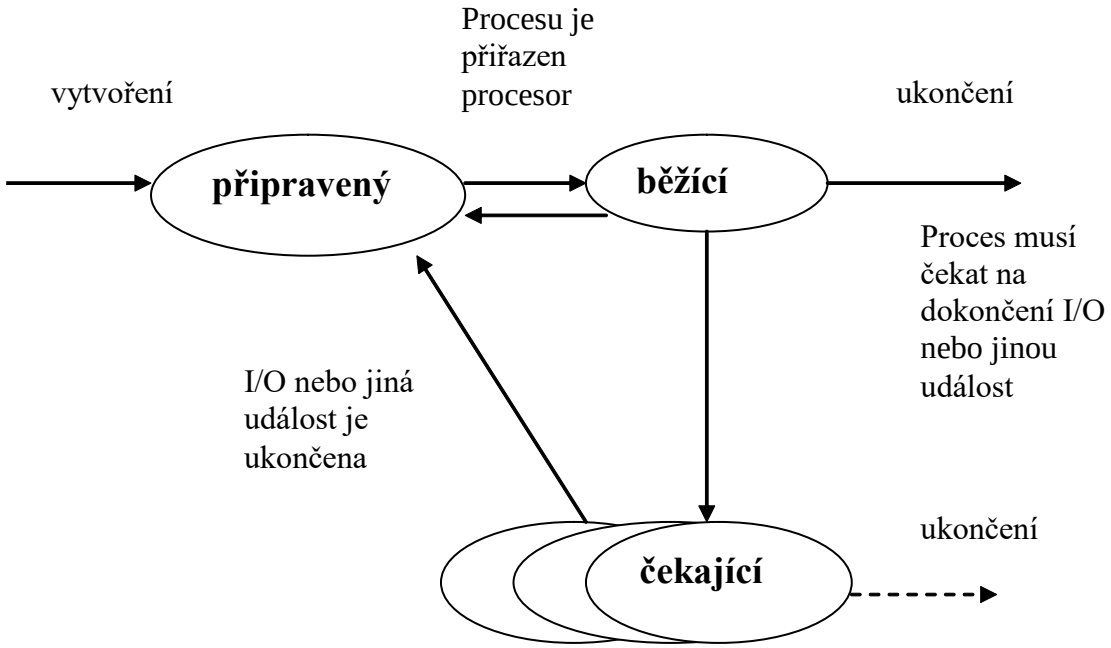
Detail dvou-stavového modelu



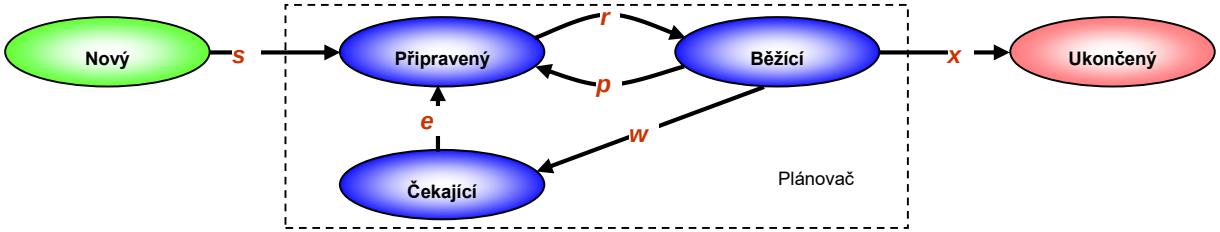
Řízení multiprogramového zpracování



3 stavový model



Význam přechodů



Přechod	Význam
s	Proces vzniká – *start*
r	Procesu je přidělen procesor (může pracovat) – *run*
w	Proces žádá o službu, na jejíž dokončení musí čekat – *wait*
e	Vznikla událost, která způsobila, že se proces „dočkal“ – *event*
x	Proces ukončil svoji existenci (sám nebo „násilně“) – *exit*
p	Procesu byl odňat procesor, přestože je proces dále schopen běhu, tzv. **preempce** (např. vyčerpání přiděleného času) – *preemption*. *Tento přechod v nepreemptivním plánování (➡) neexistuje!*

Implementace procesů.

- Datová struktura spravovaná OS – řídící blok procesu - Process Control Block (PCB), tabulka procesů, obsahuje položky které:
  - zachycují současný stav procesu,
  - umožňují jedinečnou identifikaci procesu (číslo procesu),
  - obsahují ukazatele na rodičovské procesy a na potomky,
  - obsahují parametry procesu důležité pro plánování procesů (priorita procesu)

Implementace procesů.

- obsahují informace důležité pro správu paměti (ukazatele na rozsah přidělené paměti, tabulky stránek nebo segmentů),
- uchovávají obsah registrů procesoru (střadače, indexové registry, ukazatel zásobníku a další registry),
- uchovávají obsah čítače programu (adresu příští instrukce, která má být vykonána),
- obsahují seznam otevřených souborů, vstupně/výstupních zařízení,
- definují číslo procesoru, na kterém proces běží (u víceprocesorových systémů),
- obsahují informace pro účtování (celkový čas využití procesoru, číslo úlohy, číslo účtu apod.)

Implementace procesů.

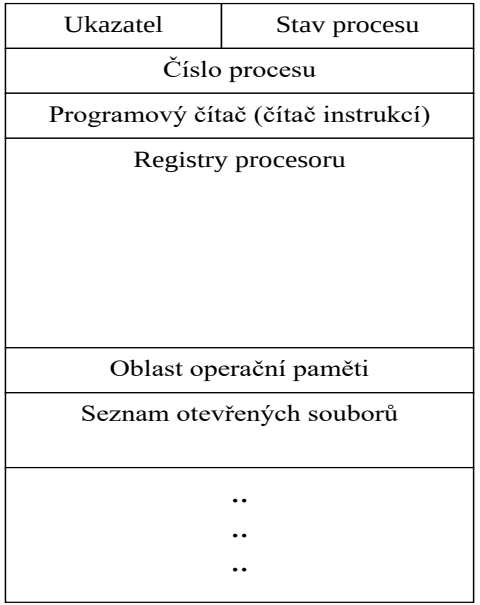
- Význam při přepnutí kontextu procesu (context switch)
- **Režie**
- Ukládání údajů z PCB do speciálních registrů (sady HW registrů)
- Uchování adresového prostoru procesu při přepnutí kontextu

Deskriptor procesu – Process Control Block (PCB)

- Identifikátor procesu (pid)
- Globální stav (*process state*)
- Čítač instrukcí (PC)
- Registry procesoru
- Informace potřebné pro plánování procesoru(ů) - priorita, využití CPU, …
- Informace potřebné pro správu paměti - odkazy do paměti (*memory pointers*), registry MMU
- Účtovací informace (*accounting*)
- Stavové informace o V/V (*I/O status*)
- Kontextová data (*context data*)
  - Otevřené soubory
  - Proměnné prostředí (*environment variables*)
  - Spojka pro řazení PCB do front a seznamů

PCB – strukturovaný záznam podobný tabulce

- Process Control Block – tabulka obsahující informace potřebné pro definici a správu procesu
  - stav procesu (běžící, připravený, …)
  - čítač instrukcí
  - registry procesoru
  - informace potřebné pro správu paměti
  - informace potřebné pro správu I/O
  - účtovací informace

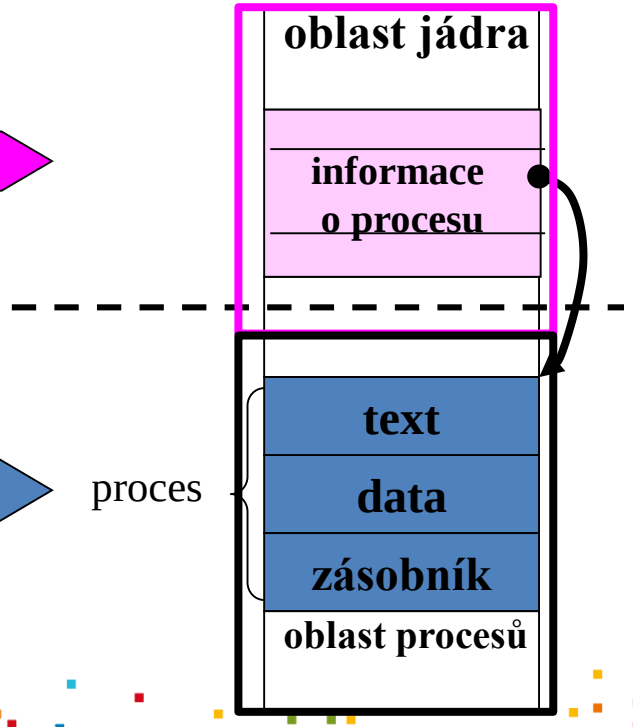


Přepnutí kontextu procesu

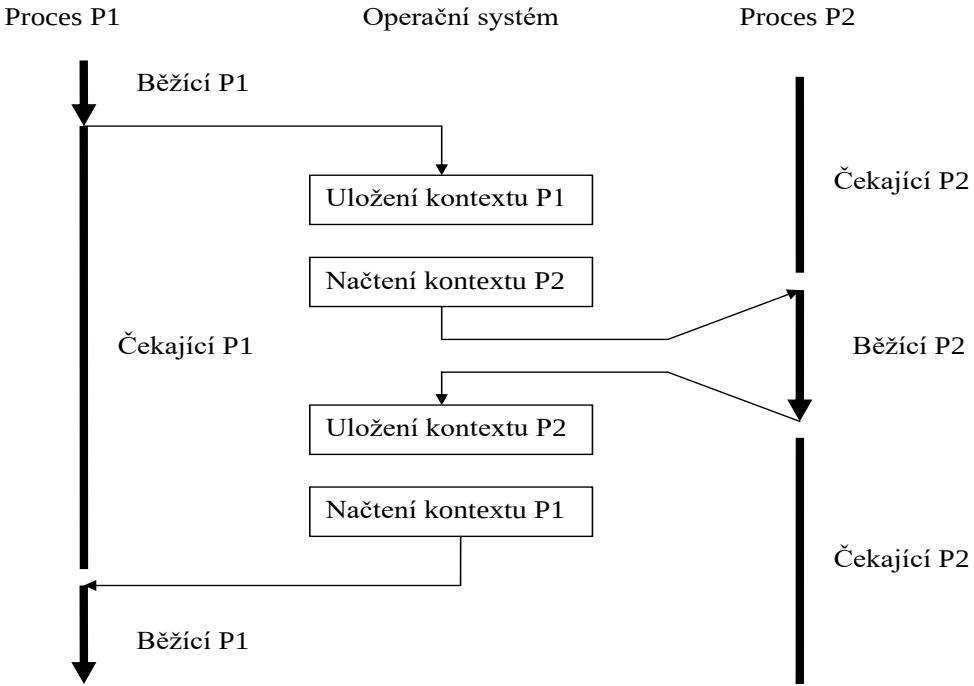
- Přechod od procesu A k B zahrnuje tzv. přepnutí kontextu
  - Přepnutí od jednoho procesu k jinému výhradně v důsledku nějakého přerušení
  - Proces A → operační systém/přepnutí kontextu → proces B
  - Nejprve OS uchová (zapamatuje v PCBA) stav původně běžícího procesu A
  - Provede se potřebná akce v jádru OS
  - Nastaví stav nově běžícího procesu B (z PCBB)
- Přepnutí kontextu představuje režijní ztrátu (zátěž)
  - během přepínání systém nedělá nic efektivního
  - časově nejnáročnější je správa paměti dotčených procesů
- Doba přepnutí závisí na hardwarové podpoře v procesoru
  - minimální hardwarová podpora při přerušení:
    - uchování čítače instrukcí
    - naplnění čítače instrukcí hodnotou z vektoru přerušení
  - lepší podpora:
    - ukládání a obnova více registrů procesoru jedinou instrukcí

Operační systémy - procesy

- **Proces** – spuštěný a zpracovávaný program.
- Součásti procesu
  - V paměti
    - Systémová tabulka procesů
    - Oblast OP, kde je program uložen
      - strojové instrukce – „text“
      - data
      - zásobník
  - Registry v procesoru
    - pracovní
    - systémové
    - stavové



Graf přepnutí procesu



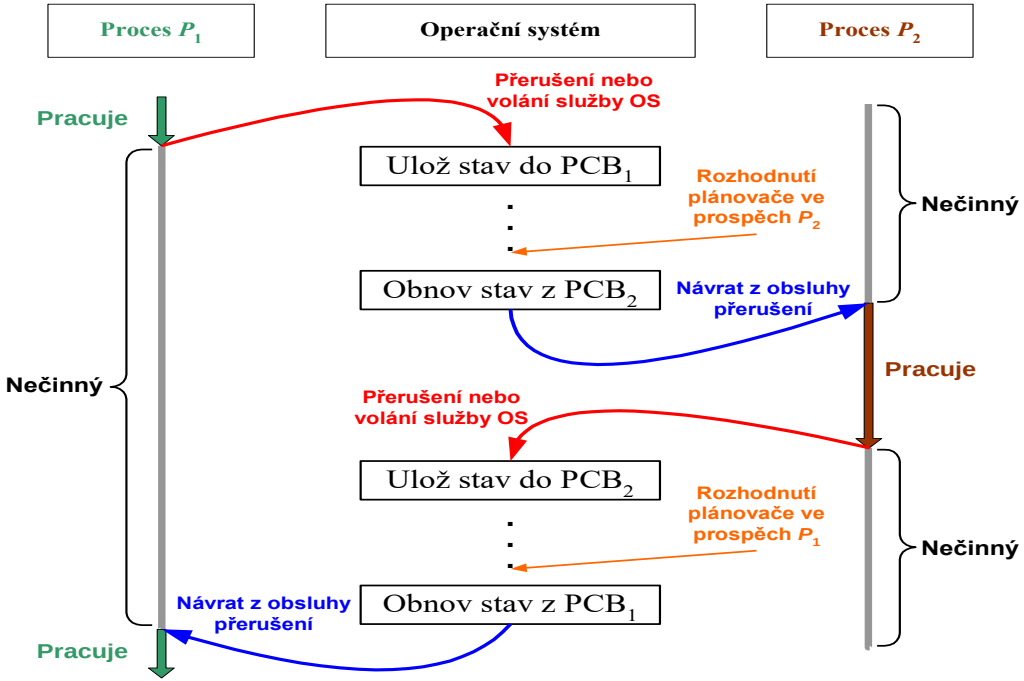
Popis přepnutí kontextu

- Vyžádá se služba, akceptuje se některé asynchronní přerušení, obslouží se a nově se vybere jako běžící proces
- Když OS přepojuje CPU z procesu X na proces Y, musí:
  - uchovat (uložit v PCB procesu X) stav původně běžícího procesu
  - zavést stav nově běžícího procesu (z PCB procesu Y)
- Přepnutí kontextu představuje režijní ztrátu (zátěž)
  - během přepínání systém nedělá nic efektivního
- Doba přepnutí závisí na konkrétní HW platformě
  - Počet registrů procesoru, speciální instrukce pro uložení/načtení všech registrů procesoru apod.
- Při přerušení musí procesor
    - uchovat čítač instrukcí
    - zavést do čítače instrukcí hodnotou adresy vstupního bodu ovladače přerušení z vektoru přerušení

Změna kontextu - Context switch

- U operačních systémů s preemtivním plánováním procesoru může být proces přerušen mezi libovolnými dvěma strojovými instrukcemi v programu a řízení může být předáno jinému procesu.
- Programy jsou však psány tak, že nepředpokládají, že by došlo ke změně obsahu registrů procesoru případně některých dalších oblastí mezi dvěma instrukcemi. Při přepnutí na jiný proces musí být také změněny další registry (ukazatel na tabulku stránek, klíč pro ochranu paměti, registr udávající, zda je procesor v privilegovaném stavu apod.).
- Proto se při přepínání mezi procesy provádí tzv. uložení kontextu (context save) původně běžícího procesu a obnovení kontextu (context restore) procesu, kterému se přiděluje procesor.
- Pod pojmem context je myšlen stav procesoru (obsah registrů), stav případného koprocesoru, případně i stav dalších zařízení. Tento context se ukládá buď na zásobník procesu, nebo do předem připravené oblasti dat v adresním prostoru procesu.

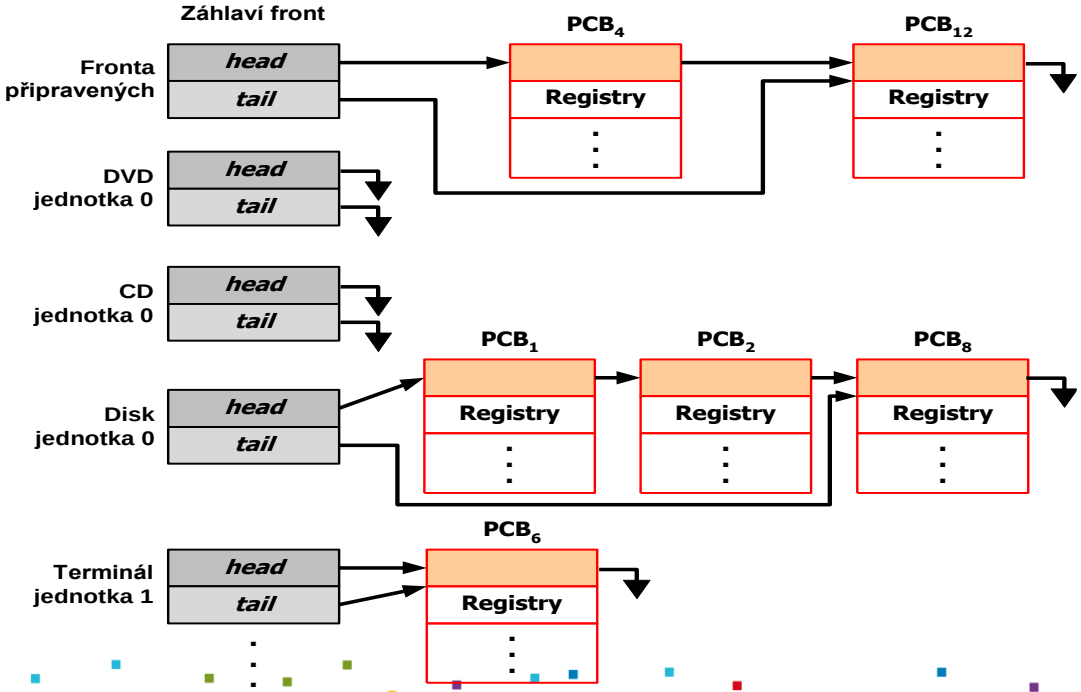
Režie přepínání mezi procesy



Fronty a seznamy procesů pro plánování

- Fronta připravených procesů
  - množina procesů připravených k běhu čekajících pouze na přidělení procesoru
- Fronta na přidělení zařízení
  - Samostatná fronta pro každé zařízení
- Seznam odložených procesů
  - množina procesů čekajících na přidělení místa v hlavní paměti, FAP
- Fronta na semafor
  - množina procesů čekajících synchronizační událost
- Fronta na přidělení prostoru v paměti
  - množina procesů potřebujících zvětšit svůj adresní prostor
- Procesy mezi různými frontami migrují

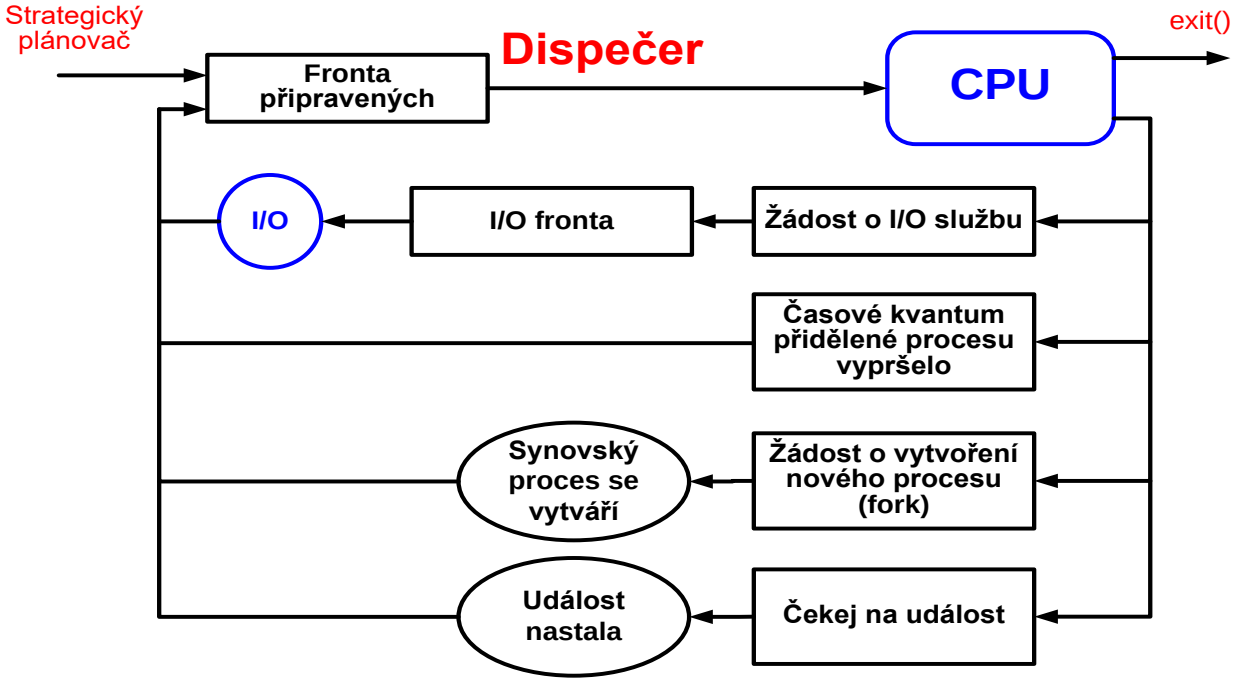
Fronty a seznamy procesů – příklad



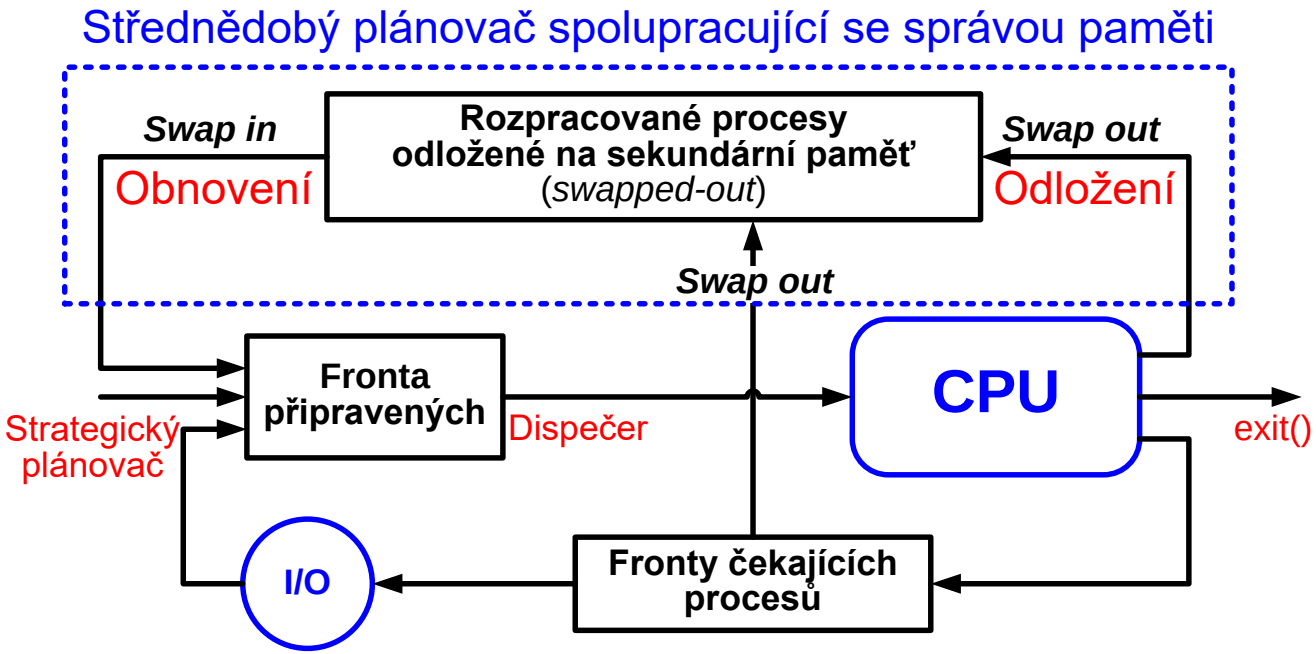
Plánovače procesů v OS

- **Dlouhodobý plánovač** (strategický plánovač, *job scheduler*)
  - Vybírá který požadavek na výpočet lze zařadit mezi procesy, a definuje tak stupeň multiprogramování
  - Je vyvoláván zřídka (sekundy až minuty), nemusí být rychlý
- **Krátkodobý plánovač** (operační plánovač, dispečer, *dispatcher*):
  - Základní optimalizace využití procesoru(ů)
  - Vybírá proces, který poběží na uvolněném procesoru přiděluje procesu procesor (CPU)
  - vyvoláván velmi často, desítky – stovky milisekund, musí být rychlý
- **Střednědobý plánovač** (taktický plánovač)
  - Logicky patří částečně do správy hlavní paměti
  - Taktika využívání omezené kapacity FAP při multitaskingu
  - Vybírá který proces je možno zařadit mezi odložené procesy ➡ (uvolní tím prostor zabíraný procesem ve FAP)
  - Vybírá, kterému odloženému proces lze opět přidělit prostor ve FAP

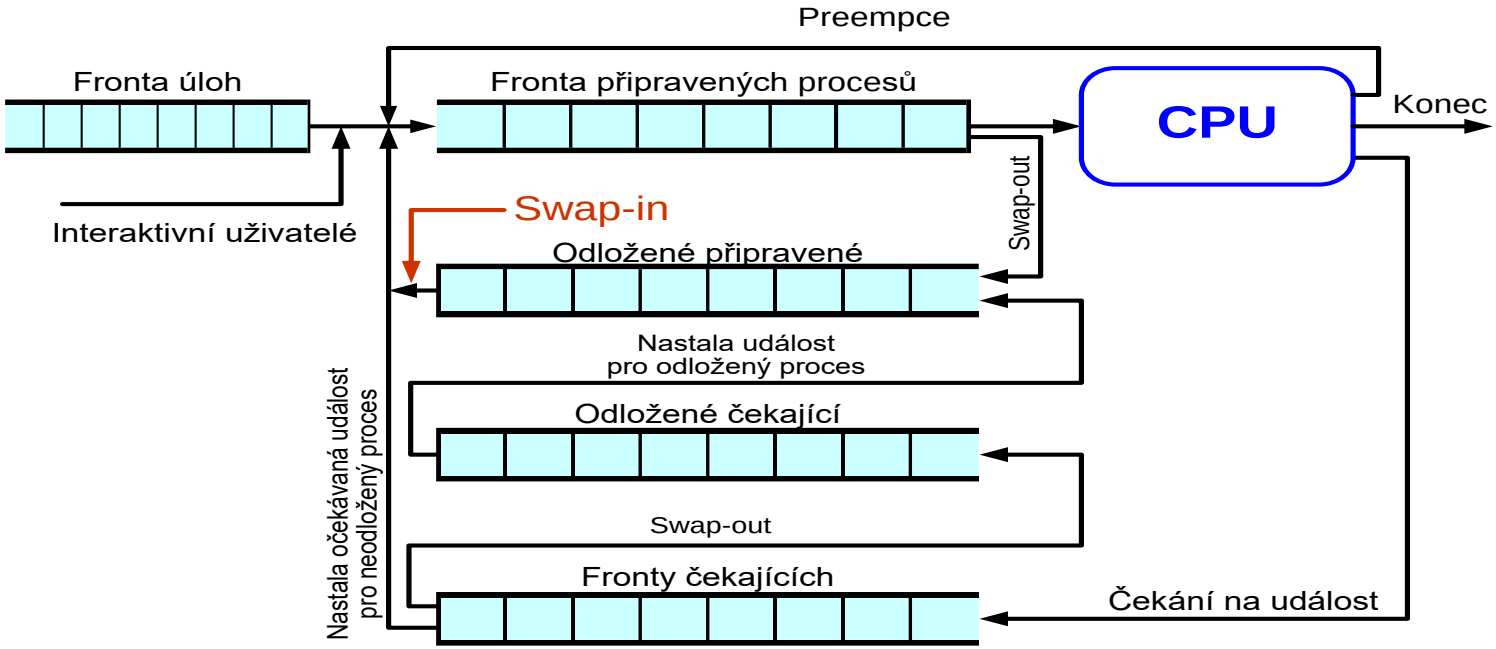
Strategický plánovač a dispečer



Odkládání procesů a střednědobé plánování



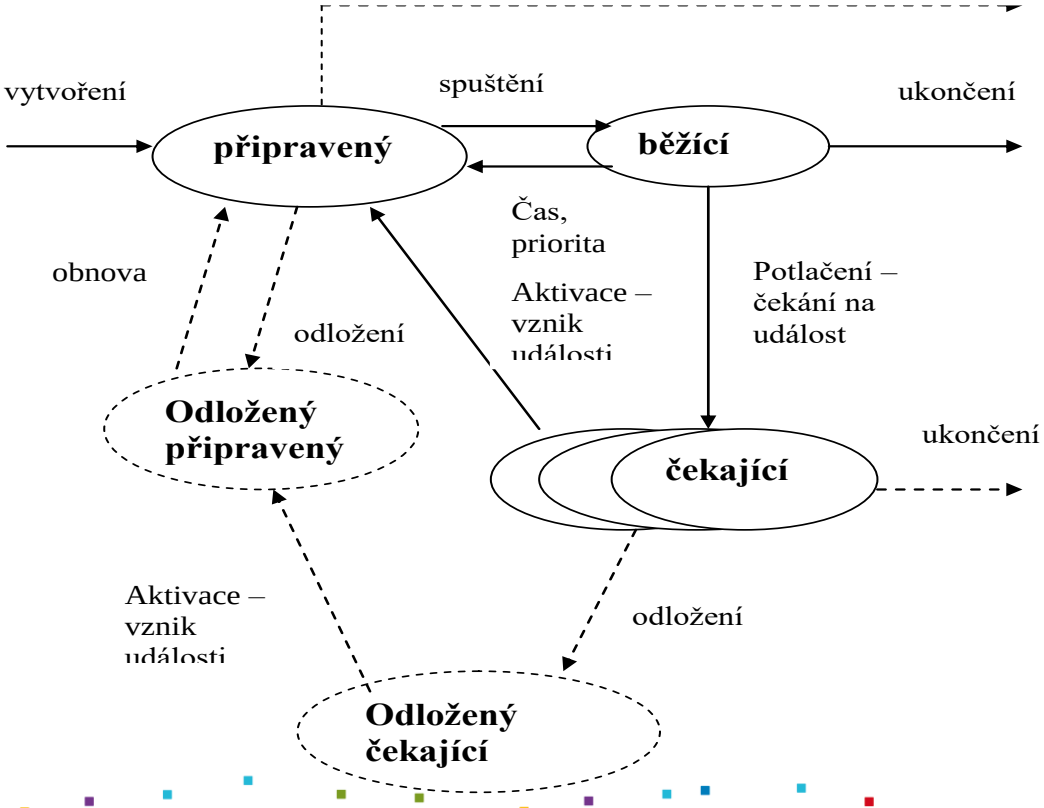
Frontový model plánování CPU



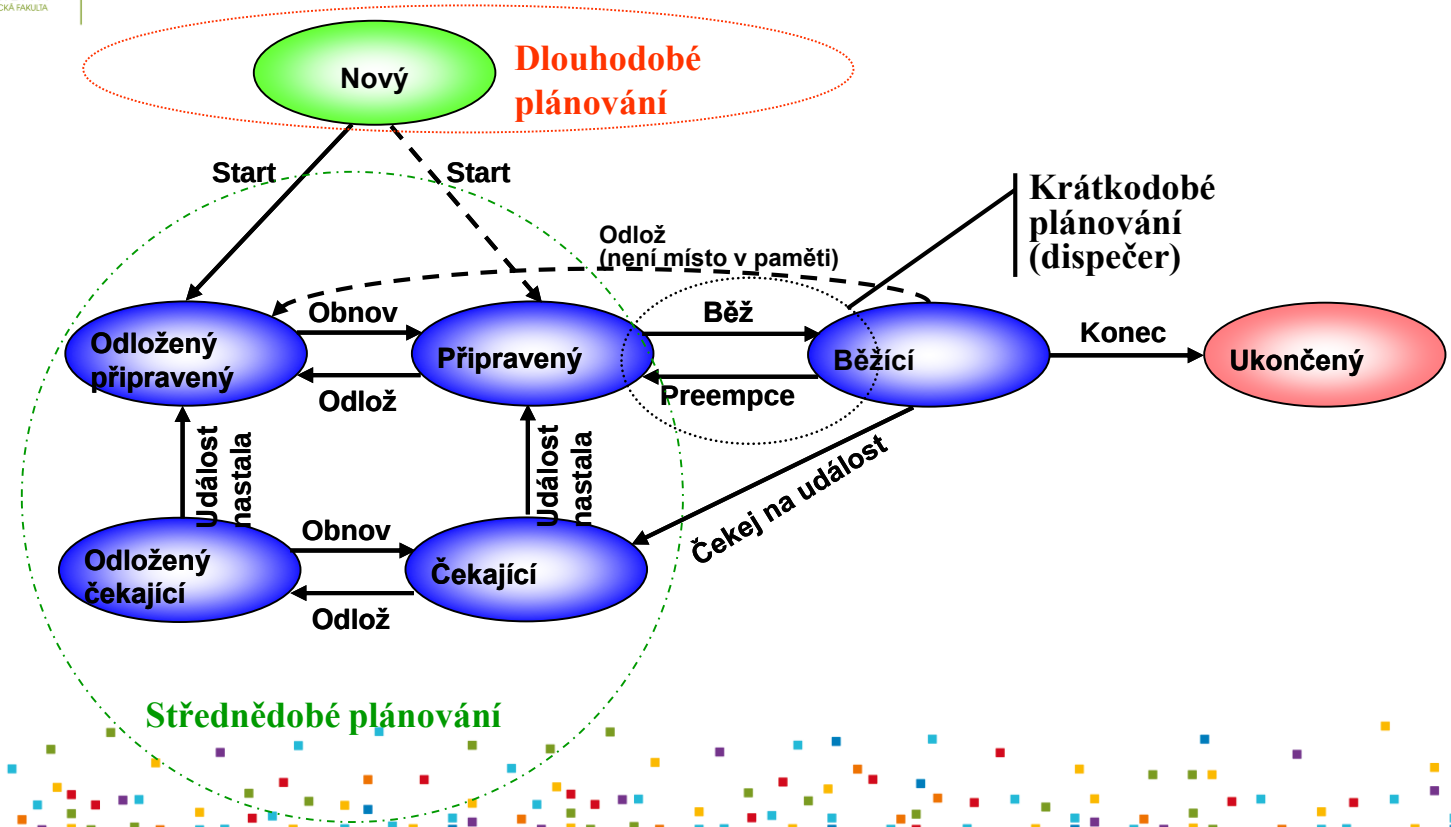
Odkládání, *swapping*

- Běžící proces musí mít alespoň pro aktuální části LAP přidělen FAP
  - jinak by nemohl pracovat
- I když se používá princip virtuální paměti, příliš mnoho procesů ve FAP snižuje výkonnost (alespoň částečně)
  - jednotlivé procesy obdrží malý prostor ve FAP a po částech se jim vyměňuje aktuální úsek LAP ve FAP enormně často
- OS musí provádění některých procesů odložit
  - **odložení** – *swap-out*, okopírování na disk
  - **obnova** – *swap-in*, zavedení do FAP
- Přibývají tak další dva stavy procesů
  - **odložený připravený** – nechybí mu nic kromě místa paměti
  - **odložený čekající** – čeká na nějakou událost a i kdyby byl v paměti, stejně by nebyl schopen běhu

5-stavový model



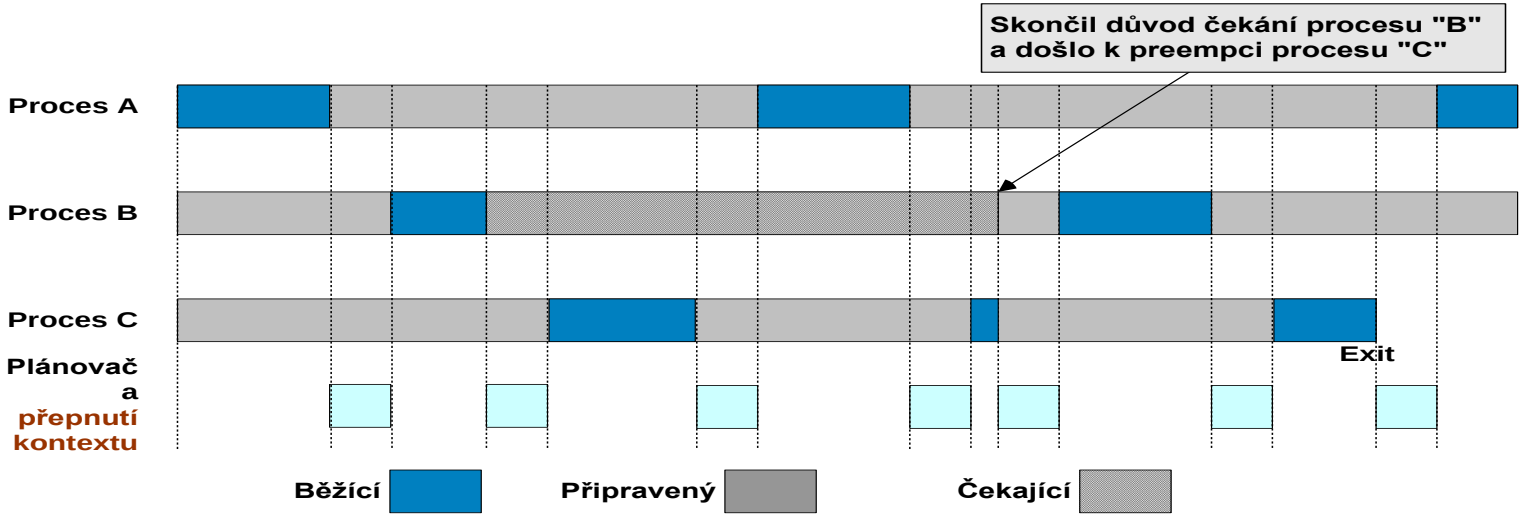
Plánovače procesů



Plánovač CPU a typy plánování

- Plánovač CPU vybírá z procesů, které jsou v hlavní paměti, ty procesy, které jsou připravené (*ready*)
- Plánovač rozhoduje v okamžiku, kdy proces:
  1. přechází ze stavu běžící do stavu čekající
  2. končí
  3. přechází ze stavu čekající do stavu připravený
  4. přechází ze stavu běžící do stavu připravený
- První tři případy se vyskytují v obou typech plánování
- Poslední je charakteristický pro plánování preemptivní

Stavy procesů v čase – preemptivní případ



Přechody procesů mezi stavy

- Čekající – odložený čekající
  - Všechny procesy jsou čekající a OS dělá prostor pro přidělení běžícímu procesu
- Odložený čekající – odložený připravený
  - Stala se očekávaná událost (stavovou info má OS přístupnou)
- Odložený připravený – připravený
  - Fronta připravených se vyprázdnila (skoro)
- Připravený – odložený připravený
  - Nepravděpodobné
  - Nejsou čekající procesy a je potřeba uvolnit RAM

Stavy procesu

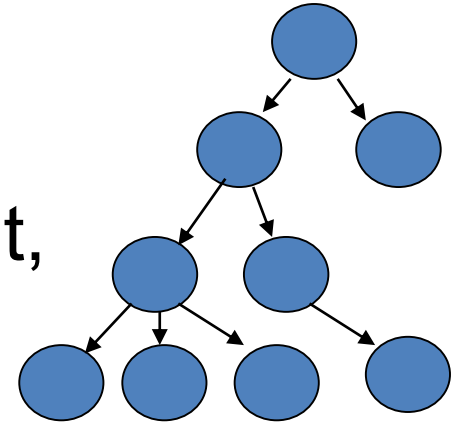
- Úplná množina stavů procesů:
  - **odložený a blokovaný** (na vnější paměťové zařízení pro vytvoření prostoru v hlavní paměti pro další procesy),
  - **odložený a připravený** (před zařazením do fronty připravených procesů musí být uložen do hlavní paměti),
  - **mátoha (zombie)** - proces vykonal systémové volání pro ukončení procesu (volání jádra exit), neexistuje, ale zůstaly po něm ještě záznamy v příslušných datových strukturách OS (tabulce procesů), závěrečná stav před definitivním ukončením.

Operace s procesy

- Vytvoření procesu (create),
- Ukončení procesu (terminate),
- Odložení procesu (suspend),
- Obnovení procesu (resume),
- Změna priority procesu,
- Zablokování procesu (/block)
- Odblokování procesu (wakeup)
- Přidělení procesoru, plánování běhu (dispatch)
- Meziprocesová komunikace

Vytvoření procesů

- pojmenování,
- zavedení do seznamu procesů,
- přiřazení počátečních hodnot priorit,
- přiřazení počátečních prostředků,
- vytvoření datové struktury PCB,
- Systémové volání pro vytvoření procesu (child process – parent process) - hierarchická struktura procesů v systému (strom procesů)



Operace s procesy

- Přidělování prostředků nově vytvořenému procesu.
- Stav rodičovského procesu po vytvoření procesu-potomka.
- Stav procesu-potomka po zániku rodičovského procesu

Přerušení

- HW čítač
- Signál „page fault“
- Vykonání I/O operace v důsledku realizace instrukce programu
- Přerušení → událost, která vede ke změně pořadí, ve kterém procesor vykonává instrukce (odezva na asynchronní nebo vyjímečnou událost)

Odezva OS na přerušení

- OS převezme kontrolu (HW),
- OS zajistí uchování přerušeného procesu,
- OS provede analýzu přerušení (modul obsluhy přerušení – interrupt handler),
- modul obsluhy přerušení obslouží přerušení,
- je obnoven stav přerušeného procesu,
- a pokračuje se v jeho vykonávání.

Přerušení

- Operační systémy – systémy řízené přerušením,
- Možnost přetížení
  - Zákaz přerušení,
  - Rozdělení přerušení do tříd přerušení s prioritami,
- Koncepce přerušení silně ovlivní architekturu výpočetního systému

Procesy jádra operačního systému

- Operace s procesy řídí část OS nazývaná jadro systému (kernel, nucleus, core)
  - obsluhu přerušení,
  - vytvoření a ukončení procesu,
  - přechod mezi dalšími stavy procesu,
  - přidělování a odebírání procesoru ,
  - odložení procesu, obnovení procesu,
  - synchronizaci procesů,
  - meziprocesovou komunikaci,
  - manipulaci s PCB,

Jádro operačního systému

- podporu činnosti vstupně/výstupních zařízení,
- podporu přidělování a uvolňování paměti,
- podporu systému souborů,
- podporu mechanismu volání procedury a návratu z volání procedury,
- podporu účtování v systému

Plánování procesů (process scheduling)

- krátkodobé (short-term), CPU scheduling (plánování procesoru): výběr kterému z připravených procesů bude přidělen procesor, ve všech víceúlohových systémech
- střednědobé (medium-term): výběr který blokovaný nebo připravený proces bude odsunut z vnitřní paměti na disk, je-li vnitřní paměti nedostatek (swap out, roll out)
- dlouhodobé (long-term), job scheduling (plánování prací, úloh): výběr, která úloha bude spuštěna (má význam zejména při dávkovém zpracování). Účelem je namixovat úlohy tak, aby byl počítač co nejvíce vytížen (třídy úloh dle náročnosti).

Strategie plánování procesoru - podle těchto kritérií:

- spravedlnost: každý proces dostane spravedlivý díl času procesoru
- efektivita: udržovat maximální vytížení procesoru, příp. jiné části systému
- čas odezvy: minimalizovat dobu odezvy pro interaktivní uživatele
- doba obrátky: minimalizovat dobu zpracování každé dávkové úlohy
- průchodnost: maximalizovat množství úloh zpracovaných za jednotku času

Cíle plánování a kritéria kvality plánů

- Využití CPU - *maximalizace* kontinuální užitečné činnosti CPU
- Propustnost - *maximalizace* počtu procesů, které dokončí svůj běh za jednotku času
- Doba obrátky - *minimalizace* doby potřebné pro provedení konkrétního procesu
- Doba čekání - *minimalizace* doby, po kterou proces čekal ve frontě připravených
- Doba odpovědi - minimalizace doby, která uplyne od okamžiku zadání požadavku na spuštění procesu do jeho první reakce, např. prvního výpisu na terminál, nikoli doba do poskytnutí úplného výstupu jakožto výsledku běhu celého procesu

OSTRAVSKÁ UNIVERZITA
PŘÍRODOVĚDECKÁ FAKULTA

Plánovací algoritmy

Typ	Využívá charakteristickou veličinu	Princip
FCFS, také FIFO (first come - first served / First in - first out)	t_{oi}	řádný frontový režim
SXFS (shortest execution - first served)	t_i	proces s nejkratší dobou provádění, je první obsloužen
LCFS (least completed - first served)	t_{bi}	přednostně se obsluhuje proces, který zatím běžel nejkratší dobu
EDFS (earliest - due-time fisrst served)	t_{di}	přednostně se obsluhuje proces, kterému zbývá nejméně času na dokončení, tj. do okamžiku, kdy musí být dokončen
HSFS (highest static priority first served)	P_i	přednostně se obsluhuje proces s nejvyšší statickou prioritou
RR (round-robin)	Δt_i	cyklická obsluha procesů po časových intervalech

OSTRAVSKÁ UNIVERZITA
PŘÍRODOVĚDECKÁ FAKULTA

Děkuji za pozornost