

Přepnutí kontextu

Ve víceúlohových systémech je obvykle k dispozici méně procesorů (méně jader CPU), než je procesů, které by měly zároveň běžet. Proto se u takových systémů využívá pseudoparalelismus, který umožňuje mít zdánlivě spuštěno zároveň více procesů. Procesy čekají ve frontě (ve stavu „připraven“) a postupně je jim na určitou dobu (tzv. časové kvantum) přidělován procesor. Jak už víme, k přepínání procesů dochází zhruba 100 až 1000 krát za sekundu (podle výkonu počítače, nebo podle výše režie přepínání, kterou chceme obětovat ve prospěch plynulosti).

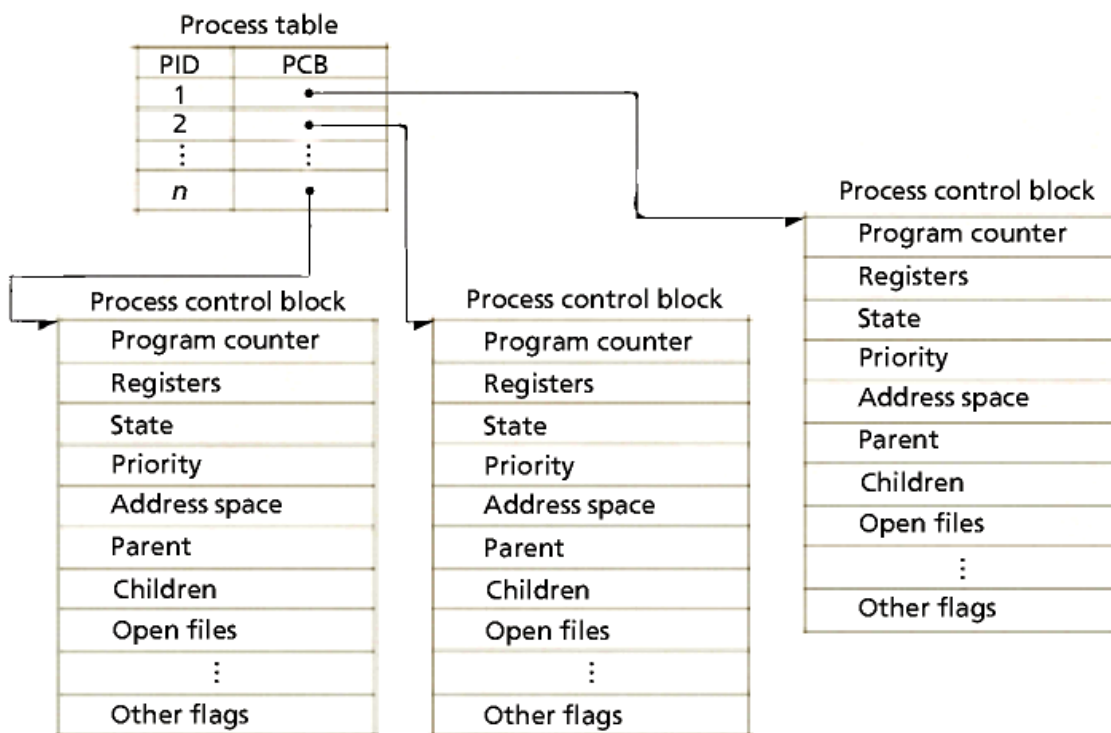
Při přepínání procesů je nutné, aby proces po opětovném spuštění pokračoval od stejného místa, ve kterém byl přerušen a aby v procesu až na časové zpoždění nebylo poznat, že k přerušení došlo. Z tohoto důvodu je nutné při přerušení vykonávání procesu (mezi dvěma libovolnými strojovými instrukcemi) **uschovat kompletní stav procesu** a při opětovném přidělení procesoru stejnému procesu tento **stav kompletně obnovit**. Kompletní uložení stavu procesu označujeme jako **uložení kontextu** (anglicky *context save*) a **obnovení kontextu** (anglicky *context restore*). **Při výměně procesů na procesoru dochází k uložení kontextu jednoho a obnovení kontextu druhého.** Tuto činnost označujeme souhrnně jako **změna kontextu** (anglicky *context switch*).

Tabulka popisu procesů (PCB)

Tabulka popisu procesů (anglicky *process control block*, zkratka PCB) **je datová struktura v jádře operačního systému, která uchovává data potřebná k běhu procesu.** (Např: Čítač, registry- (vstupní/výstupní hodnoty), stav procesu, priorita, adresový prostor, podproces/nadproces, otevřené soubory ...) Každý proces má svoji PCB, která je využívána při změně kontextu k uložení stavu procesu. Obsah PCB je v různých OS různý.

Současně s „**PCB**“ je v operačním systému vedena „**Tabulka procesů**“ která udržuje ukazatele na PCB každého procesu. (Umožňuje tak přístup do PCB) (viz obrázek na další straně)

Po ukončení procesu, OS odstraní záznam z tabulky a uvolní zdroje (paměť, CPU...)



(ukázka „tabulky procesů“ a PCB jednotlivých procesů)

Plánování procesů

Plánování procesů (anglicky **scheduling**) je v informatice úkol jádra operačního systému, ve kterém je spuštěno více procesů najednou. Týká se tedy víceúlohových systémů podporujících multitasking. **Plánování procesů řeší výběr, kterému následujícímu procesu bude přidělen procesor a proces tak poběží**, přičemž výběr je závislý na **prioritách** jednotlivých procesů a **algoritmu**, kterým výběr proběhne.

Běžné operační systémy (pro desktopové počítače) vyžadují, aby byla při přidělování procesoru jednotlivým procesům **zachována jistá míra spravedlnosti**. Operační systém reálného času (anglicky *real-time operating system*) vyžadují, aby navíc byly splněny dodatečné podmínky, například aby výběr byl preciznější, deterministický* a pracoval se zárukami. (Jak už víme operační systémy reálného času se používají například v robotice, automatizaci, telekomunikacích, vestavěných „jednoúčelových“ systémech.)

* **Deterministický algoritmus** je v informatice označení pro algoritmus, který vždy ze stejných výchozích (vstupních) podmínek svým během vytvoří stejné výsledky (je tedy předvídatelný). Každý aktuální i následující krok vykonávání algoritmu je vždy jednoznačně definován.

Druhy plánování procesů

Operační systém může podporovat až 3 druhy plánování procesů **krátkodobé** (anglicky *short-term*), **střednědobé** (anglicky *mid-term*) a **dlouhodobé** (anglicky *long-term*). Název typu plánování vyjadřuje frekvenci, s jakou k ní dochází. **Typy plánování též rozlišují, v jakém stavu procesu k němu dochází.**

- **Dlouhodobé plánování** (anglicky *long-term*)

Dlouhodobé plánování se označuje též jako **plánování úloh** (anglicky *job scheduling*) je výběr, která úloha bude spuštěna. Má význam zejména u dávkového zpracování. **Jeho účelem je naplánovat spouštění úloh tak, aby byl počítač maximálně využit**, například vhodného mixu úloh, které jsou náročné na I/O (vstupy/výstupy) nebo CPU. **V současné době není obvykle u desktopových systémů implementován, avšak je velmi důležitý u operačních systémů reálného času**, protože systém by v případě spuštění více procesů, než může bezpečně zvládnout, nemohl plnit garantované limity.

- **Střednědobé plánování** (anglicky *mid-term*)

Střednědobé plánování používají systémy s virtuální pamětí. **Jde o výběr, který blokový nebo připravený proces bude odsunut z vnitřní paměti na pevný disk, je-li vnitřní (operační) paměti nedostatek** (anglicky *swapping out* a *swapping in*). Důvodem pro odložení procesu může být **absence aktivity procesu, nízká priorita procesu**, nebo proces, který **alokuje příliš velké části paměti**, když je potřeba paměť pro jiné procesy. (na hodině si ukážeme, jak v OS vypnout / zapnout nebo nastavit velikost „swapovacího souboru“, který se pro toto odkládání procesů používá)

- **Krátkodobé plánování** (anglicky *short-term*)

Krátkodobé plánování se označuje též jako plánování procesoru (anglicky *CPU scheduling*), při němž se vybírá, kterému z připravených procesů bude přidělen procesor (časové kvantum). Používá se ve všech víceúlohových systémech.

Rozlišujeme **Preemptivní** a **Nonpreemptivní** (Kooprativní) plánování.

-**Nonpreemptivní** – OS musí počkat, až mu sám proces předá procesor pro **přepnutí kontextu**, tj. OS nemůže násilně odebrat procesu procesor

-**Preemptivní** – OS je schopen kdykoliv procesu odebrat CPU, tj. drží si absolutní kontrolu nad počítačem a všemi prostředky, které procesům přiděluje (CPU, operační paměť, I/O zařízení, ...). K **přepnutí kontextu** dochází zpravidla po uplynutí **časového kvanta** určeného pro běh procesu a je vyvoláno **přerušením** od časovače.

Strategie (kritéria) pro plánování procesů

Strategie použitá pro výběr, kterému z připravených procesů bude přidělen procesor, může zohledňovat různá kritéria:

- **priorita procesu** – možno jí uživatelsky ovlivnit (správce úloh)
 - **spravedlnost** – každý proces dostane spravedlivý díl času procesoru
 - **efektivita** – udržovat maximální vytížení procesoru, případně 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
-

V dnešní době existuje celá řada strategií (naprogramovaných algoritmů), které tyto kritéria zohledňují.

Například strategie:

FIFO – First In, First Out

CFS – Completely Fair Scheduler

EDF – Earliest deadline first