

Obsah

Články

Proces (program)	1
Plánování procesů	3

Reference

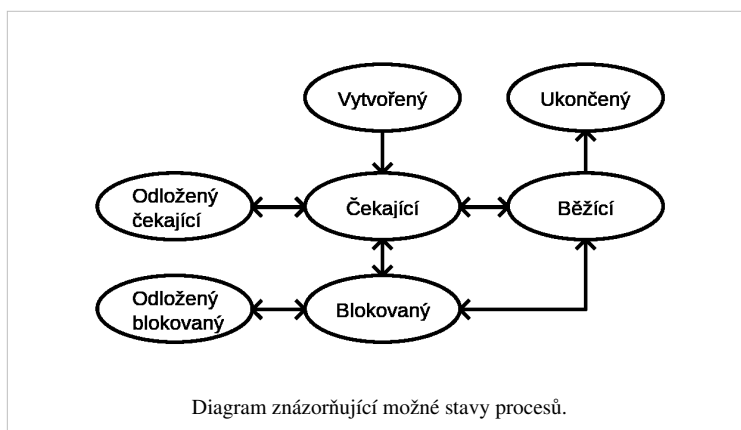
Zdroje článků a přispěvatelé	7
Zdroje obrázků, licence a přispěvatelé	8

Licence článků

Licence	9
---------	---

Proces (program)

Proces (anglicky **process**) je v informatice název pro spuštěný počítačový program. Proces je umístěn v operační paměti počítače v podobě sledu strojových instrukcí vykonávaných procesorem. Obsahuje nejen kód vykonávaného programu, ale i dynamicky měnící se data, která proces zpracovává. Jeden program může v počítači běžet jako více procesů s různými daty (například vícekrát spuštěný webový prohlížeč zobrazující různé stránky). Správu procesů vykonává operační systém, který zajišťuje jejich oddělený běh, přiděluje jim systémové prostředky počítače a umožňuje uživateli procesy spravovat (spouštět, ukončovat atp.).



Správa procesů

Moderní operační systémy umožňují spustit zároveň více procesů (textový editor, tabulkový procesor, e-mailový klient apod.), což nazýváme multitasking - multiprocessing. Pokud je v počítači méně procesorů, než je běžících procesů, musejí se procesy na procesorech střídát, což označujeme jako změnu kontextu.

Související informace naleznete také v článku Změna kontextu.

Protože programátor nepředpokládá, že by byla změněna data v průběhu vykonávání programu, je při přepínání procesů důležité uchovat všechny informace potřebné pro opětovné obnovení procesu přesně v místě a ve stavu, kde bylo jeho vykonávání přerušeno tak, aby běžící proces tuto změnu nepoznal (až na časové zpoždění mezi dvěma strojovými instrukcemi, mezi kterými došlo ke změně kontextu). Tyto informace se pro každý proces ukládají do tabulky PCB (anglicky *Process control block*), která je uložena uvnitř jádra operačního systému.

Související informace naleznete také v článku Process control block.

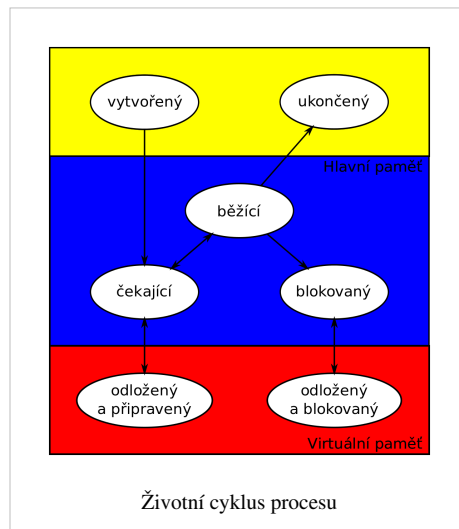
Změna kontextu je poměrně náročná operace, a proto byla zavedena vlákna (anglicky *thread*), která mají režii na přepnutí nižší. Uspoří se zejména na tom, že thready sdílejí jeden paměťový prostor. To jim též umožňuje velmi rychlou a efektivní vzájemnou komunikaci.

Související informace naleznete také v článku Vlákno (program).

Životní cyklus procesu

Životní cyklus procesu probíhá podle diagramu stavových přechodů. U několika soupeřících procesů je zařazení k běhu řízeno pravidly: časová kvanta, priorita, či bez možnosti přerušení. Přepínání procesů je značně časově náročné a může vyhovovat jen pro toleranci událostí s dlouhou latencí (například přístup na disk).

- proces je vytvořen buď příkazem uživatele (u terminálu), nebo na žádost operačního systému o provedení služby, či na žádost jiného procesu (rodiče)
- „vytvořený“ proces je ve stavu „připravený“ – připravený k vykonání a čeká pouze na přidělení procesoru
- spuštěním procesu, na základě plánovacího algoritmu přechází proces do stavu „běžící“
 - „běžící“ proces může být ukončen normálně, tj. byl celý proveden, nebo násilně ukončen uživatelem, provedením chybné strojové instrukce, chybou vstupně–výstupní zařízení, porušením ochrany paměti, nebo na žádost rodiče apod.
 - „běžící“ proces může být po vypršení časového limitu pro jeho běh (uplynutí časového kvanta) převeden do stavu „připravený“
 - „běžící“ proces může být jen jeden, máme-li jen jeden procesor, kdežto ve stavu „připravený“ může být více procesů zařazených do fronty nebo jiné datové struktury, kterou využívá plánovací algoritmus



Základní stavy procesů

Následující stavy procesů se vyskytují ve všech víceúlohových systémech:

- **vytvořený (created)** – proces je vytvořen buď příkazem uživatele (u terminálu), nebo na žádost operačního systému o provedení služby, či na žádost jiného procesu (rodiče)
- **připravený (ready)** nebo **čekající (waiting)** – připravený pro vstup do stavu běžící, čeká pouze na přidělení procesoru
- **běžící (running)** – procesu je přidělen procesor a právě se provádí příslušné programy
- **blokováný (blocked)** – proces je převeden do tohoto stavu v případě, kdy čeká na dokončení nějaké vstupně–výstupní operace, případně na skončení jiného procesu, uvolnění zdroje, synchronizační primitivum a podobně
- **ukončený (terminated)** – proces skončil

Rozšířené stavy procesů

Dva přidávané stavy jsou dostupné pro procesy v systémech, které podporují střednědobé plánování procesů (anglicky *mid-term scheduling*) s podporou virtuální paměti. V obou stavech je celý proces odsunut z vnitřní paměti na pevný disk, je-li vnitřní paměti nedostatek (anglicky *swapping out* a *swapping in*). Je chybou považovat stránkování paměti za střednědobé plánování, protože v tomto případě se odkládá celý proces. Běžné současné operační systémy rozšířené stavy procesů nepoužívají (například řada Windows NT, Mac OS X nebo Linux).

- **odložený a blokováný (suspend blocked)** – procesy, které jsou blokovány, mohou být také odloženy. V tom případě je proces jak odsunut, tak i blokován; v případě, že se fronta připravených vyprázdnila (nebo alespoň téměř vyprázdnila), pak přechází proces ze stavu „odložený připravený“ do stavu „připravený“
- **odložený a čekající (suspend waiting)** – v systémech, které podporují virtuální paměť, mohou být procesy odsunuty z hlavní paměti a umístěny do virtuální; odtud pak může být proces přesunut zpátky do stavu čekající

Související články

- Nepřerušitelný spánek
- Deadlock
- Operační systém
- Plánování procesů
- Synchronizační primitivum

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, které využívají pseudoparalelismus. 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.

Druhy plánování procesů

Operační systém může podporovat až 3 druhy plánování procesů [KOLÁŘ, 20]: 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.

Související informace naleznete také v článku Proces (program).

Dlouhodobé plánování

Dlouhodobé plánování (anglicky *long-term*) 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 spuštění úloh tak, aby byl počítač maximálně využit, například vhodného mixu úloh, které jsou náročné na I/O 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. [Stallings, 399]

Střednědobé plánování

Střednědobé plánování (anglicky *mid-term*) používají systémy s virtuální pamětí. Jde o výběr, který blokovaný nebo připravený proces bude odsunut z vnitřní paměti na pevný disk, je-li vnitřní paměti nedostatek (anglicky *swapping out* a *swapping in*). Je chybou považovat stránkování paměti za střednědobé plánování, protože v tomto případě se odkládá celý proces. Důvodem pro odložení procesu může být absence aktivity procesu, nízká priorita, časté výpadky stránek, odblokovaný proces, který již nečeká na systémové prostředky nebo alokace příliš velké části paměti, když je potřeba paměť pro jiné procesy. [Stallings, 396] [Stallings, 370]

Krátkodobé plánování

Krátkodobé plánování (anglicky *short-term*) se označuje též jako **plánování procesoru** (anglicky *CPU scheduling* – viz další odstavec), při němž se vybírá, kterému z připravených procesů bude přidělen procesor. Používá se ve všech víceúlohových systémech.

Plánování procesoru

Při plánování procesoru se v operačním systému plánovač (anglicky *scheduler*) rozhoduje, kterému procesu bude přidělen procesor, a tedy který proces v následujícím časovém úseku bude procesor počítače využívat pro svůj běh. K plánování procesoru dochází v následujících situacích (podrobnosti o stavech procesu viz článek o procesech):

1. pokud některý běžící proces přejde do stavu blokovaný
2. pokud některý proces skončí
3. pokud je běžící proces převeden do stavu připravený
4. pokud je některý proces převeden ze stavu blokovaný do stavu připravený

Preemptivnost

Pokud k plánování procesoru dochází jen v prvních dvou výše uvedených případech, označujeme takový operační systém (resp. plánovač procesů) jako nepreemptivní. Jinak se jedná o preemptivní plánování procesoru. Čtvrtý případ přeplánování používá operační systém reálného času.

Nepreemptivní

Dochází-li k plánování ve výše uvedených případech 1 až 2, jedná se o nepreemptivní plánování procesoru, u kterého musí operační systém vyčkat, až mu proces nabídne přeplánování procesoru speciálním voláním služby operačního systému. Tj. nemůže násilně odebrat procesu přidělené systémové prostředky a nedrží tedy absolutní kontrolu nad počítačem [Stallings, 396]. Důvodem může být absence příslušných hardwarových možností procesoru (16bitové procesory x86) nebo nedostatečné využití schopností pokročilejšího procesoru operačním systémem (tj. systémy Microsoft Windows 3.x, avšak například OS/2 dokázal pokročilých schopností procesorů IA-32 využít).

Preemptivní

Dochází-li k plánování ve výše uvedených případech 1 až 3, jedná se o preemptivní plánování procesoru, u kterého je operační systém schopen kdykoliv procesu odebrat CPU, tj. drží si absolutní kontrolu nad počítačem a všemi prostředky, které procesům přiděluje (CPU, operační paměť, I/O zařízení, ...). Dochází k němu zpravidla po uplynutí časového kvanta určeného pro běh procesu a je vyvoláno přerušením od časovače. Do této kategorie patří například všechny 32bitové systémy pro IBM PC kompatibilní počítače (Windows NT, Linux, Mac OS X, ...).

RTOS – operační systém reálného času

Čtvrtý případ plánování procesoru se používá zřídka, zejména u speciálních operačních systémů, jako je operační systém reálného času (RTOS).

Související informace naleznete také v článku Preempce (informatika).

Přepnutí kontextu

Ve víceúlohových systémech je obvykle k dispozici méně procesorů, 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 a postupně je jim na určitou dobu (tzv. časové kvantum) přidělován procesor. Je-li přepínání dostatečně rychlé, vzniká dojem, že procesy běží zároveň. K přepínání 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 procesoru 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*).

Související informace naleznete také v článku Změna kontextu.

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. Každý proces má svoji PCB, která je využívána při změně kontextu (anglicky *context switch*) k uložení stavu procesu.

Související informace naleznete také v článku Process control block.

Strategie plánování procesoru

Strategie použitá pro výběr, kterému z připravených procesů bude přidělen procesor, může zohledňovat různá kritéria [KOLÁŘ, 21]:

- 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

Příklady strategií plánování procesoru

Podle toho, které z výše uvedených kritérií bere tvůrce strategie plánování procesoru v úvahu a jakou jim přiřkládá váhu, se používají (a nadále vznikají) různé strategie. Používají se nejen v operačních systémech pro rozdělení času procesoru mezi jednotlivé procesy a thready, ale i pro prioritní směřování síťového provozu v routerech. Hlavním úkolem strategie je zamezit vyhladovění procesů (anglicky *starvation*) a zajistit spravedlivé rozdělení času procesoru.

- First In, First Out (FIFO), též známá jako First Come First Served (FCFS)
- Borrowed-Virtual-Time Scheduling (BVT)
- Completely Fair Scheduler (CFS)
- Critical Path Method of Scheduling
- Deadline-monotonic scheduling (DMS)
- Deficit round robin (DRR)
- Dominant Sequence Clustering (DSC)
- Earliest deadline first (EDF)
- Elastic Round Robin
- Fair-share scheduling
- Gang scheduling
- Genetic Anticipatory
- Highest response ratio next (HRRN)
- Interval scheduling
- Last In, First Out (LIFO)

- Job Shop Scheduling (see Job shops)
- Least-connection scheduling
- Least slack time scheduling (LST)
- List scheduling
- Lottery Scheduling
- Multilevel queue
- Multilevel Feedback Queue
- Never queue scheduling
- O(1) scheduler
- Priority scheduling
- Proportional Share Scheduling
- Rate-monotonic scheduling (RMS)
- Round-robin scheduling (RR)
- Shortest expected delay scheduling
- Shortest job next (SJN)
- Shortest time remaining (STR)
- Staircase Deadline scheduler (SD) – [http://lwn.net/Articles/224865/\(anglicky\)](http://lwn.net/Articles/224865/(anglicky))
- "Take" Scheduling
- Two-level scheduling
- Weighted fair queuing (WFQ)
- Weighted least-connection scheduling
- Weighted round robin (WRR)
- Group Ratio Round-Robin: O(1) – http://www.usenix.org/events/usenix05/tech/general/full_papers/caprita/caprita_html/index.html (anglicky)

Literatura

- STALLINGS, William. *Operating systems : internals and design principles*. 5.. vyd. Upper Saddle River, N.J. : Pearson/Prentice Hall, 2005. 818 s. ISBN 0-13-147954-7.
- KOLÁŘ, Petr. *Operační systémy* [online]. Liberec: 2005-02-01, [cit. 2008-08-09]. Dostupné online. ^[1]
- Understanding the Linux 2.6.8.1 CPU Scheduler ^[2] (anglicky)

Reference

[1] <http://www.nti.tul.cz/~kolar/os/>

[2] http://labe.felk.cvut.cz/~stepan/X33OSA/linux_cpu_scheduler.pdf

Zdroje článků a přispěvatelé

Proces (program) *Zdroj:* <https://cs.wikipedia.org/w/index.php?oldid=11227313> *Přispěvatelé:* Alois Mahdal, Beta M, Bezdek9, Finalman, Hkmaly, JAn Dudík, Kolarp, Ludek, Milan Keršláger, Miltim, Miraceti, Petr Kopač, Reaperman, Slady, Twardowski, Vita.Jindra, 8 anonymní úpravy

Plánování procesů *Zdroj:* <https://cs.wikipedia.org/w/index.php?oldid=11226809> *Přispěvatelé:* Alesh69, Anouarattn, DaBler, Doktory, Jj14, Jvs, Milan Keršláger, Miraceti, Smili, Tchoř, Twardowski, Zaantar, Zacatecnik, 3 anonymní úpravy

Zdroje obrázků, licence a přispěvatelé

File:Stavy procesu.svg Zdroj: https://cs.wikipedia.org/w/index.php?title=Soubor:Stavy_procesu.svg Licence: Creative Commons Attribution-Sharealike 3.0 Přispěvatel: User:Bezdek9

File:Životní cyklus procesu.svg Zdroj: https://cs.wikipedia.org/w/index.php?title=Soubor:Životní_cyklus_procesu.svg Licence: Creative Commons Attribution-Sharealike 3.0 Přispěvatel: User:Alois Mahdal

Licence

Creative Commons Attribution-Share Alike 3.0
[//creativecommons.org/licenses/by-sa/3.0/](https://creativecommons.org/licenses/by-sa/3.0/)
