ÚVOD DO OPERAČNÍCH SYSTÉMŮ

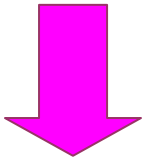
Ver.1.00

Procesy a vlákna

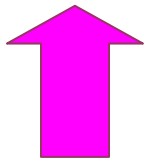
Plánování procesů (*Process Scheduling*)

České vysoké učení technické Fakulta elektrotechnická

Studijní materiály a informace o předmětu



- http://measure.feld.cvut.cz/vyuka/predmety/bakalarske/navody
- http://aldebaran.feld.cvut.cz/vyuka/uvod_do_os



Použitá literatura

[1] Stallings, W.: Operating Systems. Internals and Design Principles. 4th Edition. Prentice Hall, New Jersey, 2001.

[2] Silberschatz, A. – Galvin, P. B. - Gagne, G. : Operating System Concepts. 6th Edition. John Wiley & Sons, 2003.

[3] Tanenbaum, A.: Modern Operating Systems. Modern Operating Systems. Prentice Hall, New Jersey, 2008.

Stavový model procesu/vlákna

Základní pětistavový model:

Stav „blokován“ (také „čekající“ nebo “spící“) významně zlepšuje výsledné využití procesoru.

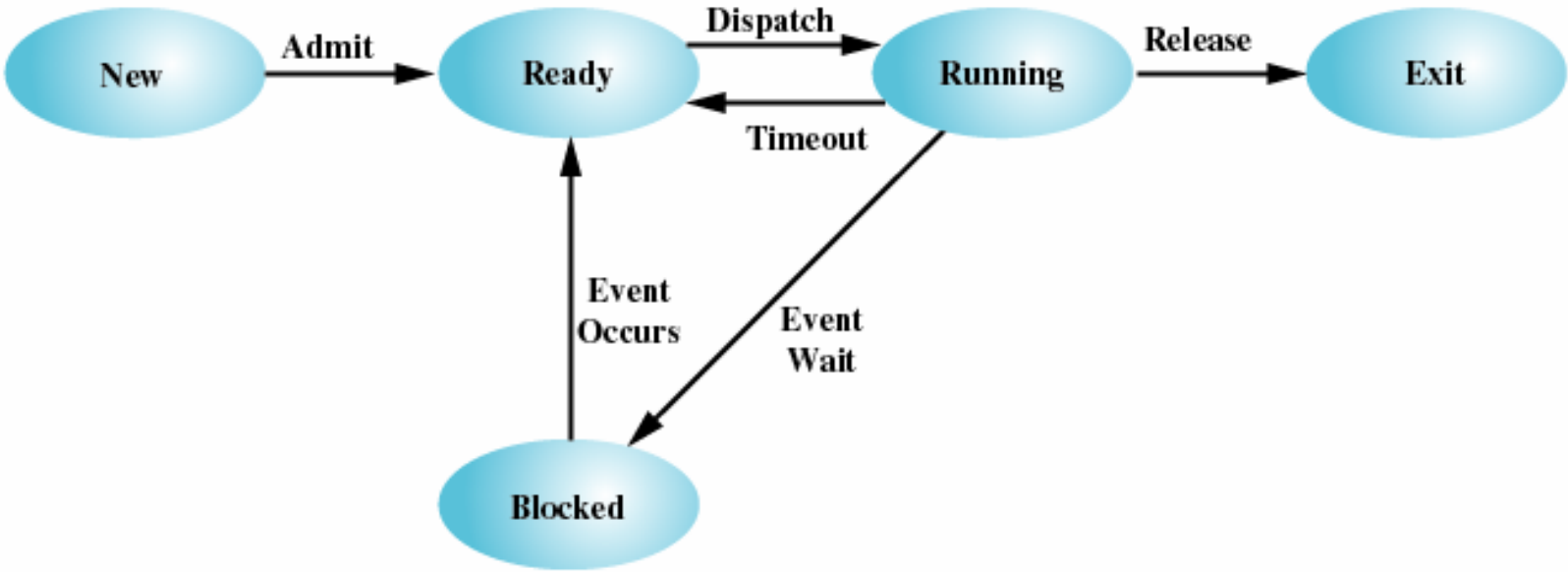


Figure 3.6 Five-State Process Model

Viz [1]

Procesy a vlákna (*Threads*)

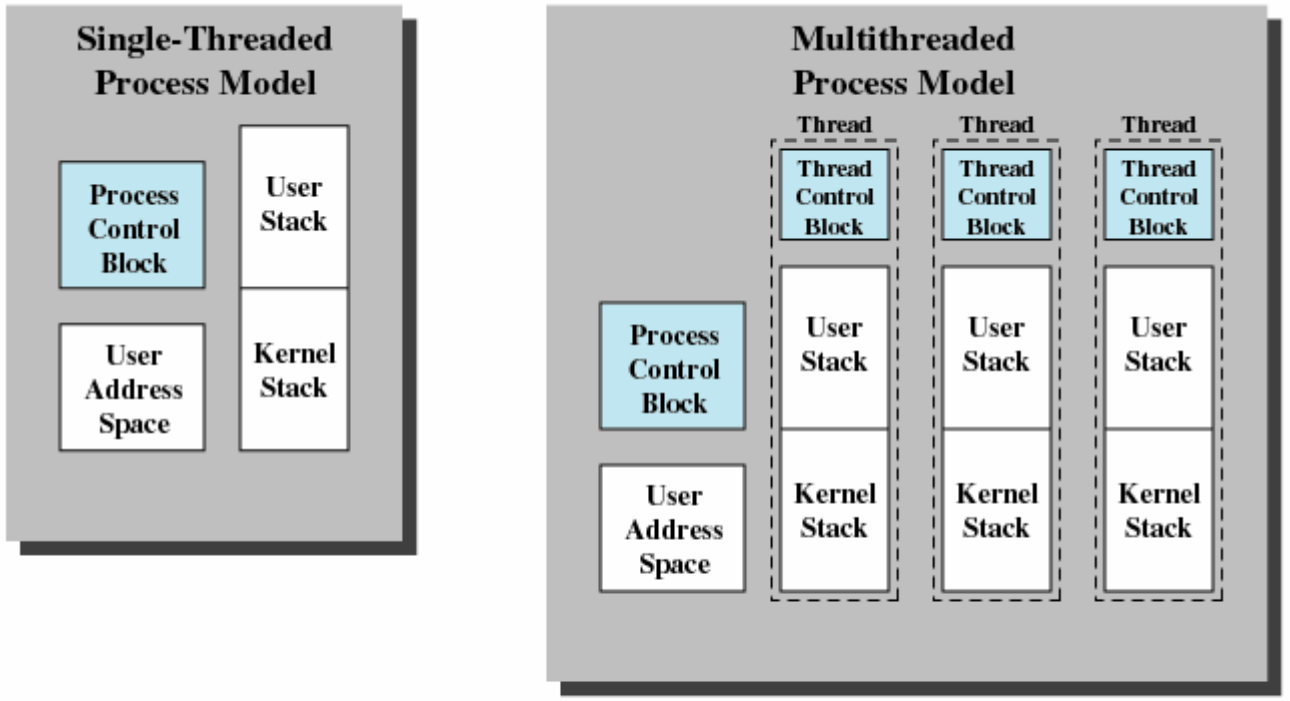


Figure 4.2 Single Threaded and Multithreaded Process Models

Viz [1]

Plánování procesů a vláken (*Schedulling*)

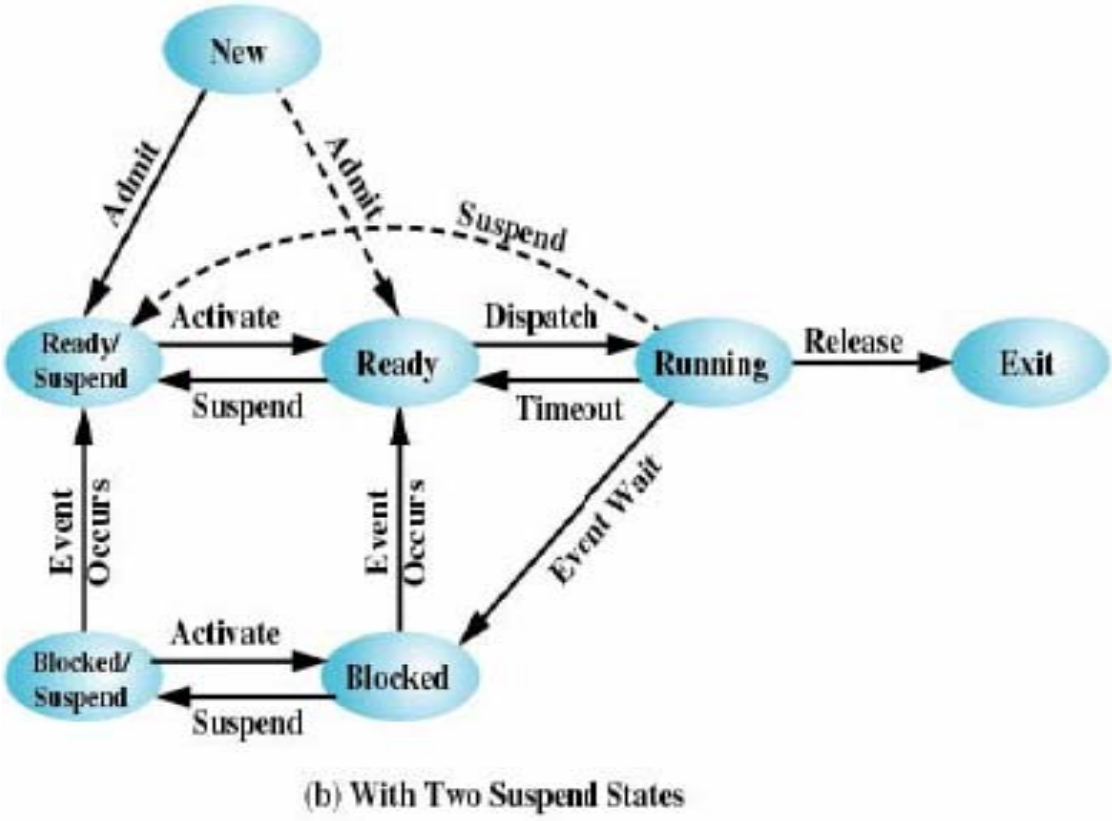
Plánování (v OS, kde je spuštěno více procesů (vláken)

- rozhodnout, který proces (vlákno) poběží
- rozhodnutí typicky optimalizuje vybraný parametr OS

Typické parametry:

- **doba odezvy** (*response time) - čas* do první odezvy
- **doba zpracování** (*turnaround time)* – celkový čas od spuštění do ukončení procesu/vlákna
- **doba čekání** (*waiting time)* – doba čekání ve frontě READY
- **propustnost** (*throughput*) - počet dokončených procesů/vláken za jednotku času
- **využití procesoru** (*CPU utilization*) – udává kolik % času CPU pracuje
- **spravedlnost** (*fairness*) – každý proces dostane „spravedlivý“ díl času

Rozšířený stavový model



(b) With Two Suspend States

Figure 3.9 Process State Transition Diagram with Suspend States

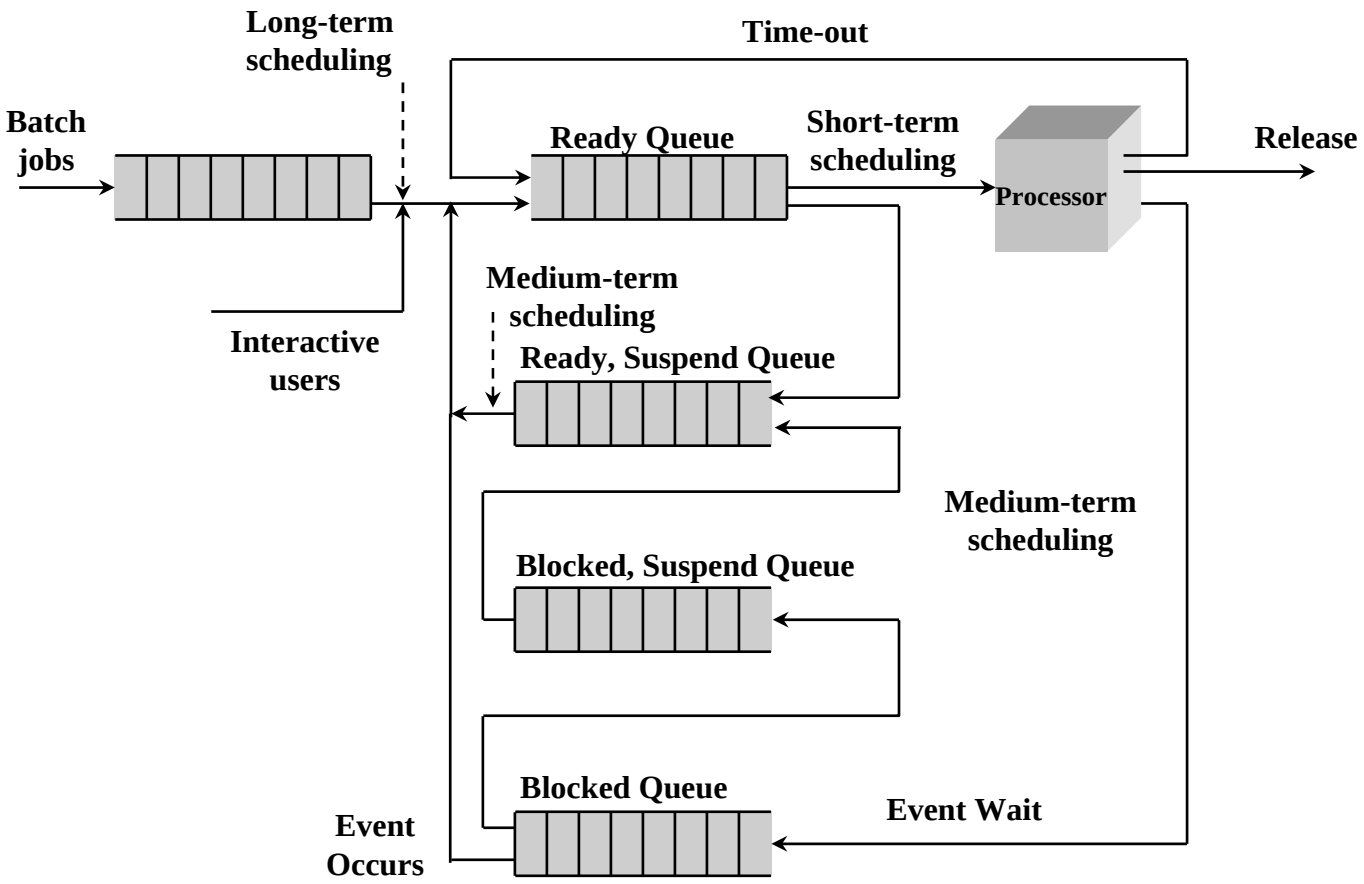
Viz [1]

Plánování procesů a vláken

Typy plánování:

- Dlouhodobé (*long-term*) – při vzniku nového procesu. Důležité u dávkového zpracování a RTOS.
- Střednědobé (*medium-term*) – přesunutí procesu ve stavu READY nebo BLOCKED na disk (*swapping)* nebo použití virtuální paměti
- Krátkodobé (*short-term*) – přepínání READY-RUNNING, BLOCKED-RUNNING, reakce na signály, přerušení, systémová volání

Plánování procesů a vláken – model front



Viz [1]

Plánování procesů a vláken

Off-line plánování:

předpoklady

- všechny procesy jsou k dispozici od začátku, jejich počet se nemění
- o všech procesech je známo jak dlouho poběží

Výsledek:

- dávkové zpracování s ohledem na požadované parametry
- běh procesů je optimální (není nutné procesy přerušovat)

Problém:

- předpoklady jsou nereálné, protože požadované informace zpravidla nejsou k dispozici

Off-line plánování– základní algoritmy

FCFS – First Come First Served:

základní algoritmus dávkového zpracování

- procesy plánovány v pořadí, v jakém přicházejí
- procesy běží dokud neskončí

SJF - Shortest Job First:

algoritmus minimalizuje průměrnou dobu odezvy

- kratší úlohy plánovány přednostně

On-line plánování

On-line plánování:

Předpoklady

- procesy nejsou k dispozici od začátku, jejich počet se nemění, objevují libovolně a neočekávaně
- doba běhu jednotlivých procesů není známá

Kritéria plánování:

- vázanost na CPU nebo I/O
- charakter procesů (interaktivní/dávkový)
- chování procesu v minulosti
- priorita

On-line plánování – **preemptivní/nepreemptivní**

K plánování procesoru dochází v následujících situacích:

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

Pokud 1. a 2. => nepreemptivní plánování

Pokud 1. až 4. => preemptivní plánování:

- potřebuje podporu HW (časovač)
- možnost měnit plán na základě nových informací
- ***context switch*** - přepnutí kontextu = přepnutí na jiný proces/vlákno
- běžící proces je přepnut do stavu READY

Preemptivní plánování umožňuje přerušení procesu/vlákna ve stavu RUNNING, přesun do stavu READY a přidělení procesoru novému procesu!

On-line plánování– základní algoritmy 1

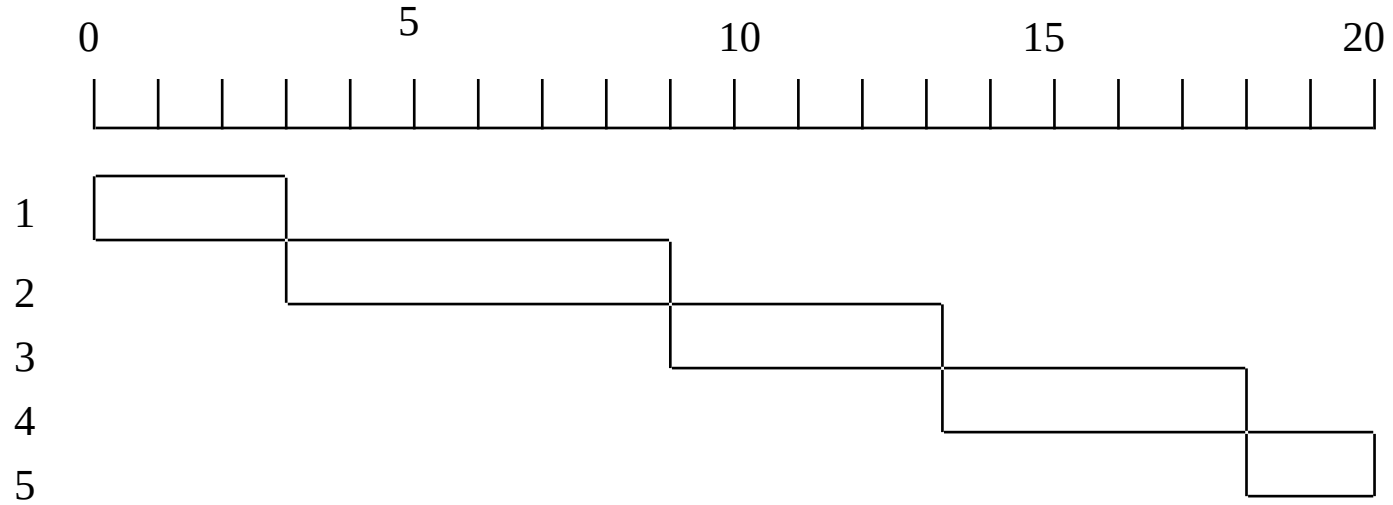
FCFS – First Come First Served:

základní algoritmus dávkového zpracování

- procesy plánovány v pořadí, v jakém přicházejí
- Nepreemptivní

Process	Arrival	Service
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

Zpracování pěti procesů (viz [1]):



Viz [1]

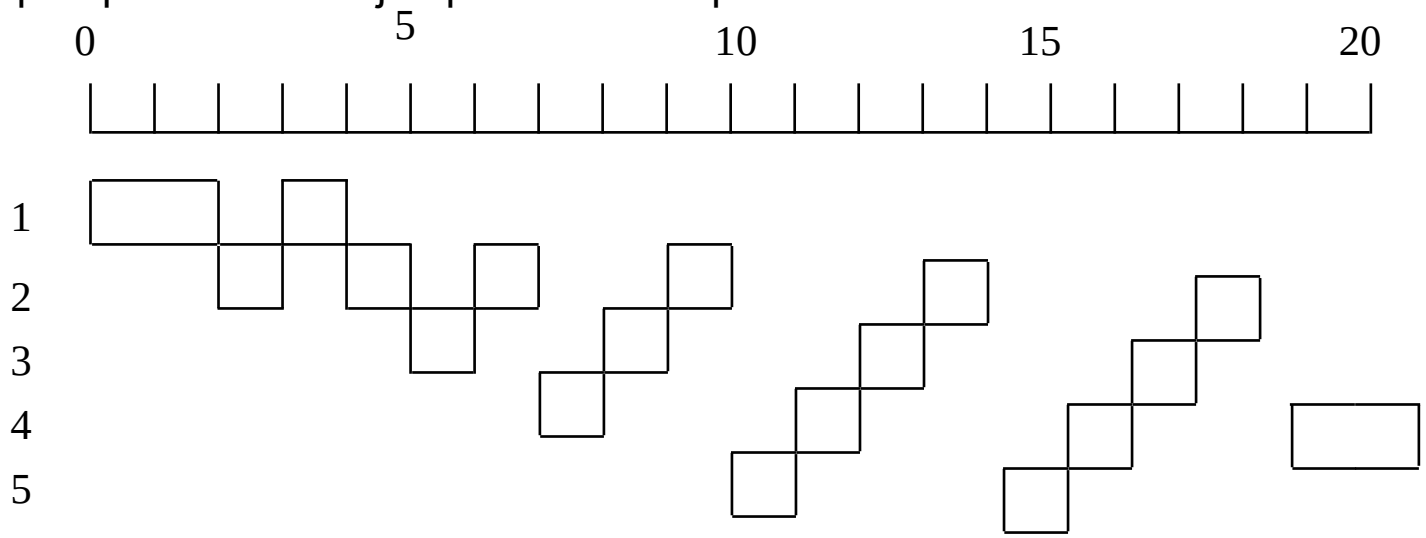
On-line plánování – základní algoritmy 2

RR – Round Robin:

algoritmus je založen na cyklickém přepínání procesů/vláken. Po dobu časového kvanta (*time slice, quantum*) je proces zpracováván procesorem. Pak dojde k přepnutí kontextu (na jiný proces).

- preemptivní plánování
- fairness
- předpokládá se stejná priorita všech procesů/vláken

Process	Arrival	Service
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2



Viz [1]

On-line plánování – základní algoritmy 3

PP – prioritní plánování (*Priority Schedulling*):

- algoritmus je založen na přepínání procesů/vláken na základě priorit jednotlivých procesů/vláken
- CPU zpracovává proces/vlákno s nejvyšší prioritou (preemtivně/nepreemptivně)

Priorita: číslo typu integer přidělené každému procesu/vláknu

- Statická – konstantní během provádění procesu
- Dynamická - může se měnit během provádění procesu (změnu provádí jádro, tedy plánovač, v závislosti na systémových cílech)
- Kombinovaná - proces má obě priority, základní plánování podle statické priority, z více procesů se stejnou statickou prioritu, dostane procesor proces s vyšší dynamickou prioritou.

On-line plánování – základní algoritmy 4

Problémy při prioritním plánování:

- Vyhladovění (*starvation*) – procesy s nejnižší prioritou se neprovedou

  Řešení:

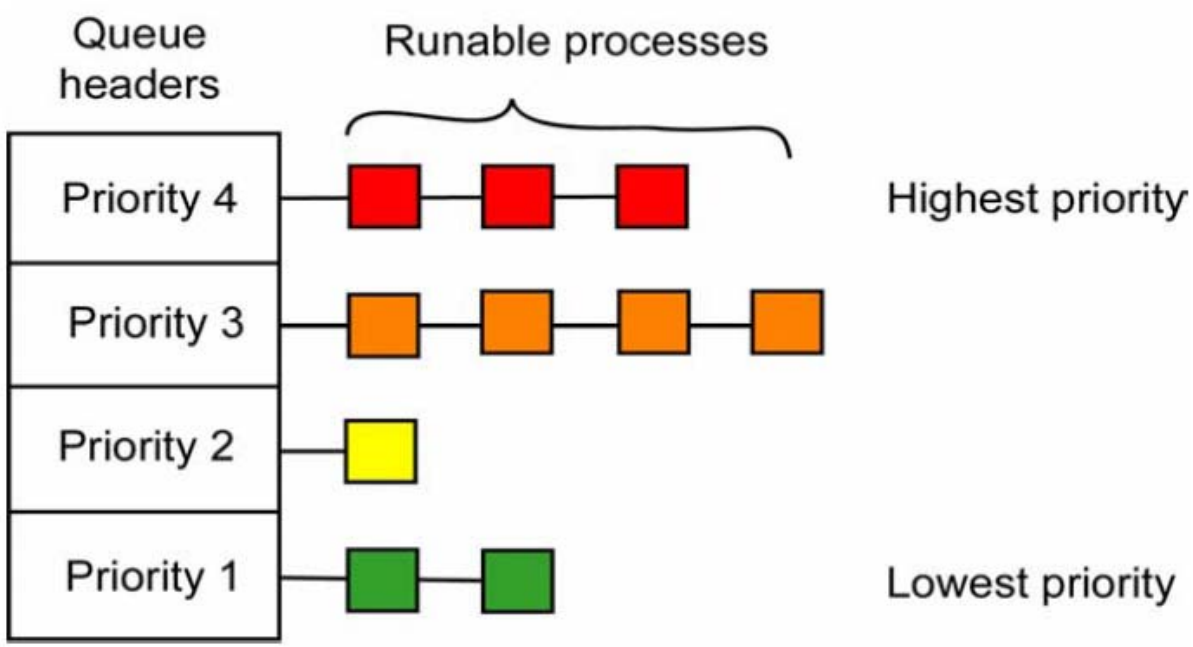
  obsluha procesů s nižší prioritou je zajištěna tak, že procesům s vyšší prioritou, kterým je přidělován procesor, se priorita postupně snižuje o 1. Jakmile priority procesů klesnou na úroveň procesů s nižší prioritou, bude procesor přidělen i těmto procesům.
- Inverze priorit – situace kdy proces s nízkou prioritou zablokuje proces s vysokou prioritou. Nepřípustné u RTOS!

  Např. proces s nízkou prioritou vstoupí do kritické sekce, pak je plánovačem odstaven od CPU. Následně je spuštěn proces s vysokou prioritou, který potřebuje vstoupit do stejné kritické sekce, ale nemůže, protože je blokována. Po určité době je znovu zpracován proces s nízkou prioritou a kritická sekce je zpřístupněna ostatním procesům, nicméně proces s vysokou prioritou bude ukončen později než proces s nízkou.

On-line plánování – základní algoritmy 5

Plánování pomocí prioritních tříd:

- procesy se stejnou prioritou seskupeny do tříd
- plánovač používá prioritní plánování mezi prioritními třídami
- v rámci každé třídy se používá RR plánování.



On-line plánování – další možnosti

Existuje řada dalších plánovacích algoritmů a strategií.

Plánování pro více procesorů

- každý procesor má vlastní „*ready queue*“
- vyvažování zátěže, aj.

Plánování pro real-time systémy (RTOS)

- aplikace řízené událostmi
- běh omezen reálným časem dokončení = *deadline*
- hard real-time / soft real-time

Tyto problémy budou zmíněny v dalších přednáškách.

Příklad - Windows XP

Prioritní preemptivní multitasking, plánování pomocí prioritních tříd

Úroveň priority vlákna je dána:

- **Třídou priority procesu**: idle (úroveň priority=4), normal (8), high (13), real-time (24)
- **Relativní prioritou vlákna** (viz 1. sloupec tabulky a následující strana)

	real-time	high	above normal	normal	below normal	idle priority
time-critical	31	15	15	15	15	15
highest	26	15	12	10	8	6
above normal	25	14	11	9	7	5
normal	24	13	10	8	6	4
below normal	23	12	9	7	5	3
lowest	22	11	8	6	4	2
idle	16	1	1	1	1	1

Viz [2]

Windows XP

Ke změně priority se používá funkce (z Win32 API):

BOOL SetThreadPriority (HANDLE hThread, int fdWPriority);

fdwPriority	úroveň priority vlákna
THREAD_PRIORITY_LOWEST	TPP - 2
THREAD_PRIORITY_BELOW_NORMAL	TPP - 1
THREAD_PRIORITY_ NORMAL	TPP
THREAD_PRIORITY_ABOVE_NORMAL	TPP + 1
THREAD_PRIORITY_HIGHEST	TPP + 2
THREAD_PRIORITY_ IDLE	1 (mimo *realtime*)
	16 (*realtime*)
THREAD_PRIORITY_TIME_CRITICAL	15 (mimo *realtime*)
	31 (*realtime*)

TPP je úroveň priority odpovídající třídě priority procesu

Windows XP

- dynamickým zvyšováním priority vlákna

- úroveň priority je dána kombinací třídy priority procesu a relativní priority, to je tzv**. bázová úroveň priority**;
- systém může tuto úroveň priority v určitých případech dynamicky zvýšit (takto může reagovat na některé I/O události – vstup z klávesnice, čtení z disku, apod. – čekající vlákno je aktivováno a ještě je mu zvýšena priorita např. o 2);
- po proběhnutí prvního časového kvanta je priorita snížena o 1, v dalším běhu opět o 1;
- to se opakuje, dokud nedojde ke snížení priority na bázovou úroveň.
- dynamické zvyšování úrovně se provádí u vláken s nižší bázovou prioritou, ale nikdy se nepřekračuje úroveň nad hodnotu 15.
- dynamicky se zvyšuje úroveň také procesům na popředí. Ve Windows NT 4.0 je možné dynamické zvyšování priority vyřadit pomocí funkcí SetThreadPriorityBoost() a SetProcessPriorityBoost().

ÚVOD DO OPERAČNÍCH SYSTÉMŮ

KONEC 4. přednášky

České vysoké učení technické Fakulta elektrotechnická