

Přepínání kontextu

pátek 11. října 2024 12:05

Přepínání kontextu, plánovače OS a plánování CPU

Přepínání kontextu

- Operace přepnutí z jednoho běžícího procesu na druhý
 - Opakuje se mnohokrát za sekundu
 - Vytváří iluzi paralelního běhu
- Podporují všechny multitaskingové OS
- Dochází k němu i při obsluze přerušení nebo změně režimu (privilegovaný vs. uživatelský)

PCB - Process Control block

- Změnu provádí dispatcher na základě scheduleru
 - Plánování CPU -> plánovací algoritmy -> fronty
- HW varianta
 - Rychlejší
 - Problém se správným uložením všech registrů
- SW varianta
 - Nejčastější
 - Pomalejší -> vyšší režie
- Řídicí blok procesu
 - Task Control Block / Task Structure
- Dynamická datová struktura obsahující nejnutnější informace pro správu a běh procesu
- Každý proces má svou PCB
- Umístěna v privilegovaném režimu
 - Chráněna před přístupem uživatele a dalšími procesy
- Kopie registrů pro jejich pozdější načtení (EIP, ESP)
- Priorita
- Účtovací informace
 - Doba běhu, poslední spuštění
- Alokovaná IO zařízení

Fronty procesů

- Fronta ukazatelů na první a poslední PCB
- Procesy mezi fronty migrují
- Synchronizace a urychlení práce

Krátkodobý plánovač

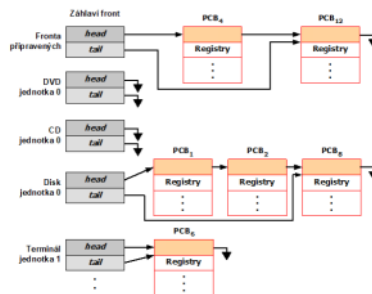
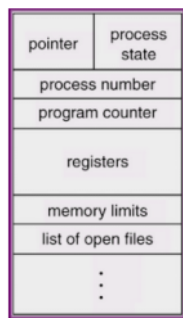
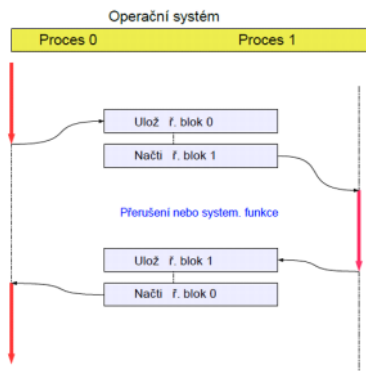
- Short Term, operační plánovač, plánování procesoru, dispatcher
- Vybírá, který proces bude využívat uvolněný CPU -> extrémně rychlý
- 2 základní fronty
 - Ready
 - IO Event

Dlouhodobý plánovač

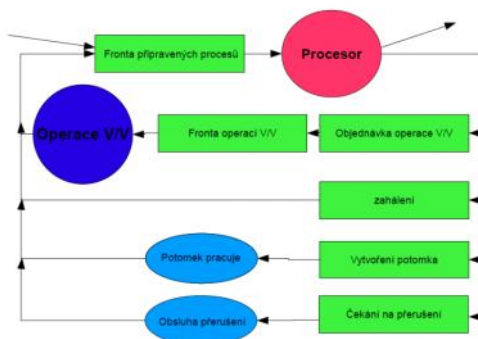
- Long Term, plánovač úloh (job scheduler), strategický plánovač
- Aktivován zřídka
 - Sekund až minuty
 - Nemusí být rychlý
- Vhodná kombinace několika úloh náročných na IO operace a CPU
 - Dávkové zpracování
- V interaktivních systémech se prakticky nepoužívá a degraduje na přímé předání práce krátkodobému plánovači

Střednědobý plánovač

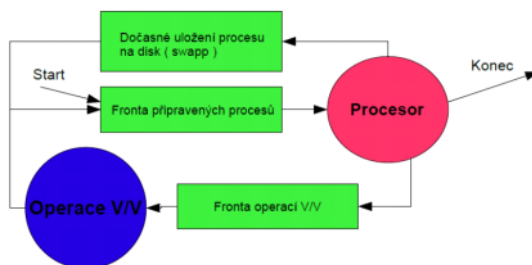
- Mid term, taktický plánovač
- Vybírá, který proces odloží z OP na swap oddíl a naopak
 - Swap out/ swap in
 - Spolupracuje se správcem hlavní paměti
- Dva nové stavy
 - Swap waiting
 - Swap ready



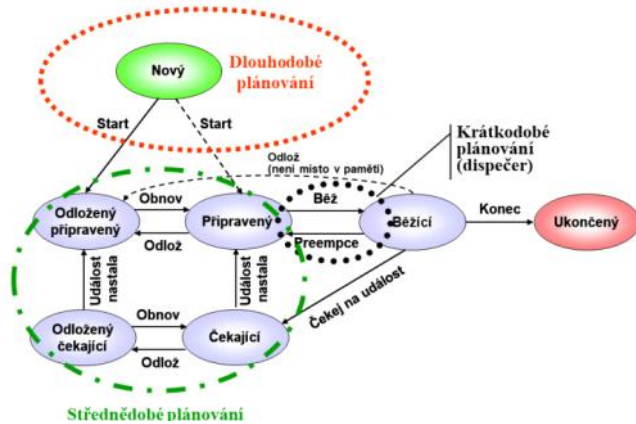
Dispatcher a dlouhodobý plánovač



Střednědobý plánovač



Životní fáze procesu – 7 stavů



Plánování CPU

- Scheduler se rozhoduje, kterému procesu přidělí procesor na základě:
 - Ukončení procesu
 - Změně stavu procesu z Run do Ready
 - Změně stavu procesu z Run do Wait
 - Změně stavu procesu z Wait do Ready

- Preemptivní
 - a, b, d
- Ne preemptivní
 - a, c, d

Cíle plánování CPU

- Využití CPU
 - Maximalizace kontinuální činnosti CPU
- Propustnost
 - Maximalizace přirozeně ukončených procesů za jednotku času
- Doba čekání
 - Minimalizace doby čekání procesu na CPU
- Doba obrátky
 - Minimalizace potřebné doby pro provedení konkrétního procesu
- Doba odpovědi
 - Minimalizace doby, která uběhne od vyvolání požadavku na spuštění procesu po jeho první reakci (např. výpis na terminál)

Preemptivní VS Ne preemptivní

TQ - Time Quantum

Preemptivní

- OS má plnou kontrolu nad procesem
 - Kdykoliv může odebrat CPU
 - Kontrola také nad všemi přidělenými prostředky
 - Algoritmy: SRTF, PS, RR, MFQS
- Většinou po uplynutí přidělené doby (TQ)
 - Interní časovač -> přerušení
- Složitější implementace OS
- Nutnost HW podpory v CPU
- Windows NT (32bit), Linux, Mac OS X, Unix

Ne preemptivní

- OS nemá plnou kontrolu nad procesem
 - Proces využívá CPU, jak dlouho potřebuje
 - OS nemůže odebrat CPU procesu
 - Problém s chybným programem -> nevrátí řízení OS -> zamrznutí
 - Algoritmy: FCFS, SJF
- Využití v tzv. Uzavřených systémech
 - Všechny procesy jsou předem známe, včetně jejich vlastností
- Windows 3.x/95/98 (16bit), Mac OS

FCFS - first come first served - FIFO - fronta

AT - arrival time - doba, kdy se proces objevil
 BT - burst time - Kolik času potřebuje proces
 CT - completion time - čas, kdy proces se dokončil
 TAT - turn around time - doba obrátky
 CT - AT
 WT - Waiting time
 TAT - BT
 VT - Visiting time - první návštěva procesu na procesor
 RT - Reading Time
 VT - AT

FCFS - First Come First Served

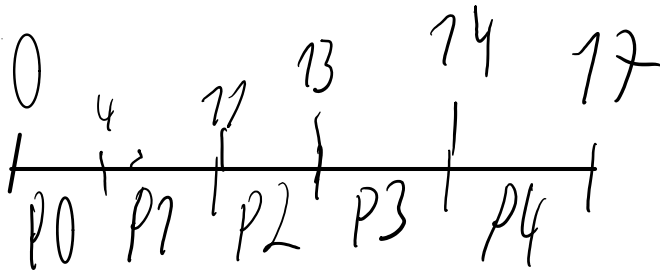
- Ne preemptivní
- Procesy jsou obsluhovány v pořadí, v jakém přišli do fronty Ready
- Nejjednodušší plánovací algoritmus
- Nevhodné pro OS se sdílením času
 - Vyžadována odpověď v přiměřené době
- Jeden proces může zablokovat ostatní na delší dobu
 - Efekt kolony
- Velká průměrná čekací doba
- Samostatně se nepoužívá
 - V kombinaci s RR -> MFQS

Proces	AT	BT	CT	VT	TAT	WT
P0	0	4	4	0	4	0
P1	2	7	11	4	9	2
P2	5	2	13	11	8	6
P3	6	1	14	13	8	7

P4	8	3	17	14	9	6
----	---	---	----	----	---	---

LIFO - last in first out - zásobník

4,2

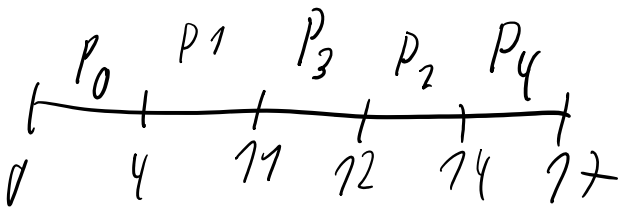


SJF - Shortest Job First

- Nepreemptivní
- Přednost mají ty úlohy u nichž je předpoklad nejkratšího běhu
 - o V případě stejného BT je vybrán dřívější proces
- Hrozí hladovění
- Mnohem menší doba čekání než u FCFS

Proces	AT	BT	CT	VT	TAT	WT
P0	0	4	4	0	4	0
P1	2	7	11	4	9	2
P2	5	2	14	12	9	7
P3	6	1	12	11	6	5
P4	8	3	17	14	9	6

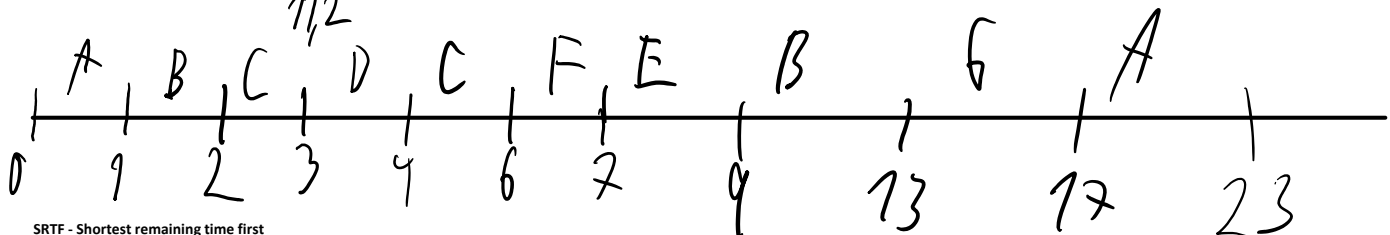
8,2 | 7,4 | 4



SRTF

Proces	AT	BT	BTR	CT	TAT	
A	0	7	6,0	23	23	
B	1	5	4,0	13	12	
C	2	3	2,0	6	4	
D	3	1	0	4	1	
E	4	2	0	9	5	
F	5	1	0	7	2	
G	6	4	0	17	11	

11,2

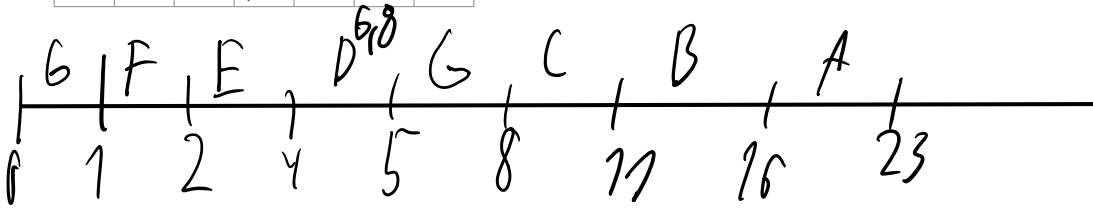


SRTF - Shortest remaining time first

- Preemptivní SJF -> SRTF
- Pokud má nově příchozí úloha nejkratší BT, přeruší a nahradí se aktuálně probíhající proces
- V případě dvou stejných BT, vybráno dle FCFS

Proces	AT	BT	BTR	CT	TAT	WT
A	6	7	0	23	17	
B	5	5	0	16	11	
C	4	3	0	11	7	
D	3	1	0	5	2	
E	2	2	0	4	2	
F	1	1	0	2	1	

G	0	4	3,0	8	8	
---	---	---	-----	---	---	--



PS - Priority scheduling

- Možnost, jak preemptivního, tak nepreemptivního řešení
- Plánování podle priority

Proces	Priority	AT	BT	BTR	CT	TAT	VT	WT
P0	2	0	4	3	25	25	0	21
P1	4	1	2	1	22	21	1	19
P2	6	2	3	2	21	19	2	16
P3	10	3	5	3,0	12	9	3	4
P4	8	4	1	0	19	15	18	14
P5	12	5	4	0	9	4	5	0
P6	9	6	6	0	18	12	12	6
						15		

P0 0-1

P1 1-2

P2 2-3

P3 3-5

P5 5-9

P3 9-12

P6 12-18

P4 18-19

P2 19-21

P1 21-22

P0 22-25

Proces	Priority	AT	BT	BTR	CT	TAT	VT	WT
A	10	0	6	4,0	22	22	0	18
B	12	1	2	0	25	24	23	22
C	9	2	3	2,0	18	16	2	13
D	4	3	5	3,0	12	9	3	4
E	11	4	1	0	23	19	22	18
F	2	5	4	0	9	4	5	0
G	6	6	4	0	17	11	12	7

A 0-2

C 2-3

D 3-5

F 5-9

D 9-12

G 12-16

C 16-18

A 18-22

E 22-23

B 23-25

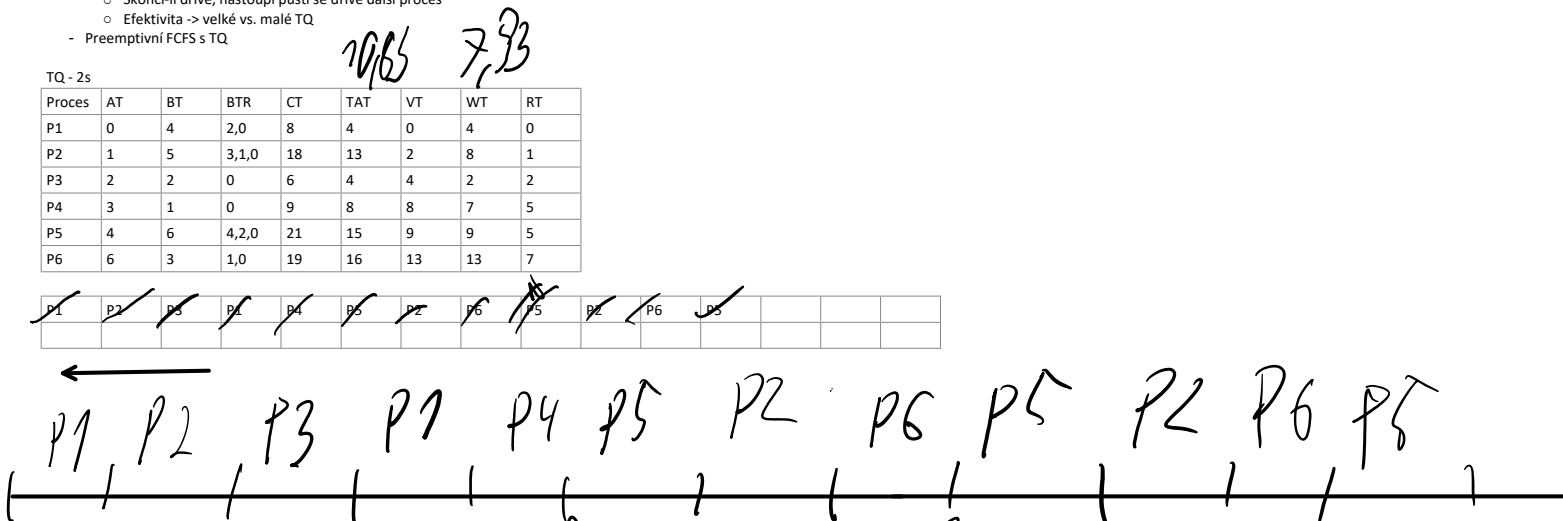
RR - Round Robin

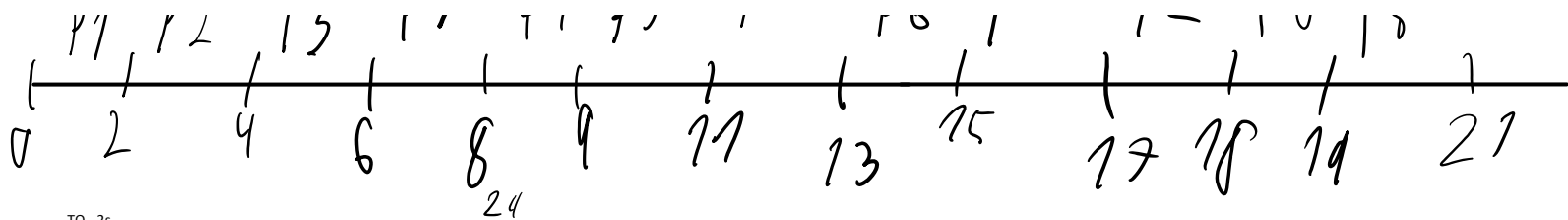
- Plánování cyklickou obsluhou
 - o Rotační plánování
- Vhodný pro OS se sdílením času
- Základ tvoří TQ - Time Quantum
 - o Maximálně po tuto dobu může proces využívat CPU
 - Poté zařazen na konec fronty -> fair play planning
 - o Skončí-li dříve, nastoupí pustí se dříve další proces
 - o Efektivita -> velké vs. malé TQ
- Preemptivní FCFS s TQ

TQ - 2s

Proces	AT	BT	BTR	CT	TAT	VT	WT	RT
P1	0	4	2,0	8	4	0	4	0
P2	1	5	3,1,0	18	13	2	8	1
P3	2	2	0	6	4	4	2	2
P4	3	1	0	9	8	8	7	5
P5	4	6	4,2,0	21	15	9	9	5
P6	6	3	1,0	19	16	13	13	7

P1	P2	P3	P4	P5	P6	P1	P2	P3	P4	P5	P6	P1	P2	P3				
---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------	--	--	--	--

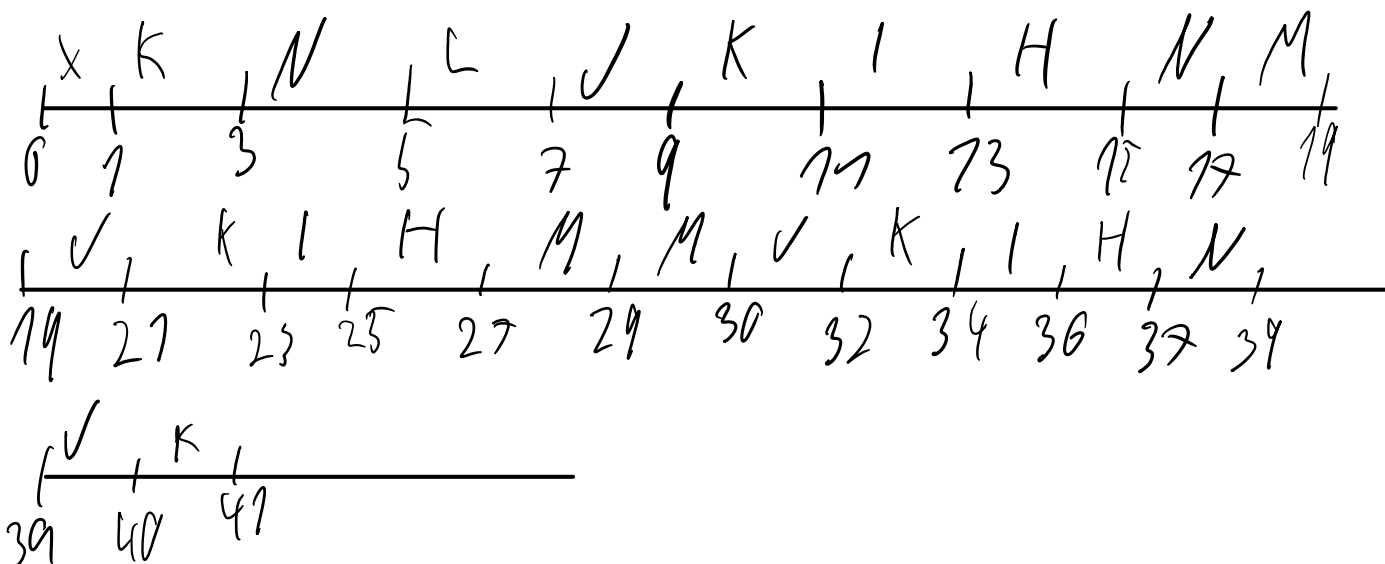




TQ - 2s

Proces	AT	BT	BTR	CT	TAT	VT	WT	RT
H	5	5	3,1,0	37	32	13	27	
I	4	6	4,2,0	36	32	11	26	
J	3	7	5,3,1,0	40	37	7	30	
K	1	9	7,5,3,1,0	41	41	1	32	
L	2	2	0	7	5	5	3	
M	6	3	1,0	30	24	17	21	
N	1	8	6,4,2,0	39	38	3	30	

K	N	L	J	K	H	N	M	J	K	I	H	N	M	J	K	I	H	N	J	K			
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	--



Procvičování

- Priority Scheduling
 - o Stejně zadání (tabulka), ale obrátit prioritu
 - Čím nižší číslo u priorit, tím má proces větší přednost
- Round Robin
 - o Stejně zadání (tabulka), ale TQ = 4ms

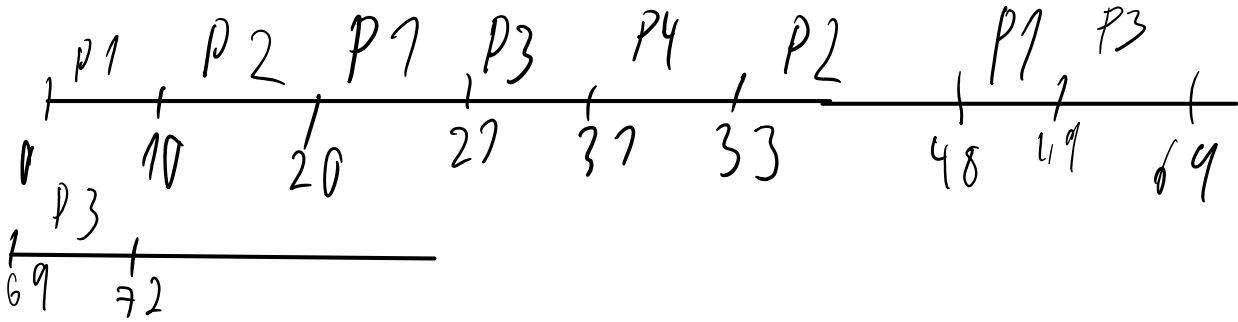
MQS - Multilevel Queue Scheduling

- Plánování pomocí více front
 - o Procesy se mezi frontami nemohou přesouvat
- Rozdělení procesů do skupin
 - o Interaktivní procesy -> vyšší priorita
 - o Procesy na pozadí -> nižší priorita
 - o Každá skupina má různé nároky na dobu odezvy
 - Plánování v rámci front
- Nutnost existence plánování mezi frontami
 - Podle priority s využitím preempece
 - Pokud není předchozí fronta prázdná, nedostane se proces z nižší fronty k CPU
 - Časové intervaly pro každou frontu
- Nejvyšší prioritu mají systémové procesy, pak interaktivní, dávkové a nakonec uživatelské

MFQS - Multilevel Feedback Queue Scheduling

- Plánování pomocí více front se zpětnou vazbou
 - o Zpětnovazební plánování
 - o Možnost přesunu procesů mezi frontami
- Jeden z nejvýznamnějších algoritmů plánování CPU
- Většinou využívá kombinace RR-RR-FCFS
 - o Procesy v první frontě mají nejkratší TQ
 - Do této fronty vstoupí každý proces
 - V následující frontě pak mají procesy delší TQ
 - o Procesy v nižší frontě mohou být vykonány pokud je předchozí fronta prázdná
- Velmi komplexní
- Hrozí stárnutí, tzv. Aging
 - o Řešením je umělé zvýšení priority
 - Přesun procesu do vyšší fronty
- Proces z vyšší fronty může preemptivně ukončit proces z nižší fronty
 - o Platí i pro frontu FCFS

Proces	AT	BT	BTR	CT	TAT	VT	WT	RT
P1	0	12	2,1,0	49	49	0		
P2	8	25	15,0	48	40	10		
P3	21	33	23,3,0	72	51	21		
P4	30	2	0	33	3	31		

[illegible]

Proces	AT	BT	BTR	CT	TAT	VT	WT	RT
D	0	53						
E	17	17						
F	34	68						
G	51	24						
H	28	31						
I	5	18						

[illegible]

- Guaranteed Scheduling
- Také znám jako „Fair Share“
- Při n uživateli je čas využití CPU rozdělen jako $1/n$
- Lottery Scheduling
- Každému procesu je přidělen tiket
- Periodické losování
- CPU je přiděleno tomu procesu, jež má výherní tiket
- Důležité procesy mohou mít více tiketů
- Kooperativní procesy si mohou předávat tikety

$$Tq = 2 \text{ ms}$$
$$P^T \quad A^T \quad B^T \quad B^T R$$

p_1	0	4
p_2	1	5
p_3	2	2
p_4	3	7
p_5	4	6
p_6	6	3