

Procesor se dodeljuje drugom procesu pod sledećim okolnostima:

- kada proces pređe u stanje čekanja na resurs (čeka na završetak I/O operacije koju je inicirao),
- kada proces roditelj čeka proces dete da završi aktivnosti,
- prilikom tranzicije RUN-STOP, odnosno kada tekući proces završi aktivnosti.
- prilikom tranzicije RUN-READY => PRE-EMPTIVE
- prilikom tranzicije WAIT-READY (u slučaju da je proces koji je napustio stanje WAIT i prešao u stanje ready višeg prioriteta od procesa koji se trenutno izvršava) => PRE-EMPTIVE

Nonpreemptive scheduling – procesor se može oduzeti samo od procesa koji je završio svoje aktivnosti ili čeka na neki resurs.

Preemptive scheduling – procesor se može oduzeti procesu koji nije završio svoje aktivnosti i nije blokiran.

Vreme potrebno za kompletiranje procesa (turnaround time) = ukupno vreme potrebno da se izvrši proces. Računa se od trenutka kreiranja do završetka procesa, a uključuje i vreme potrebno da proces uđe u red čekanja, vreme provedeno u redu čekanja, vreme korišćenja procesora i izvršenja ulazno-izlaznih operacija.

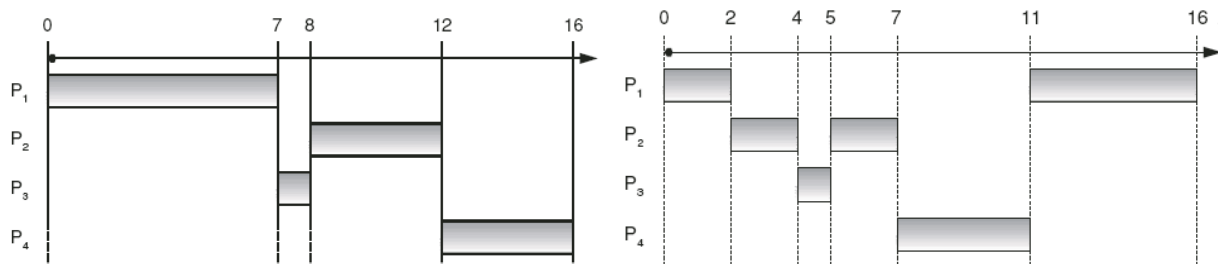
Vreme čekanja (waiting time) = vreme koje proces provede u redu čekanja na procesor (ready queue).

Vreme odziva (response time) = vreme potrebno da se nakon slanja zahteva pojave prvi rezultati izvršenja procesa. Ovo vreme je veoma bitno kod interaktivnih sistema - proces koji otpočne sa radom treba relativno brzo da prikaže neke početne rezultate na ekranu.

Scheduling algoritmi

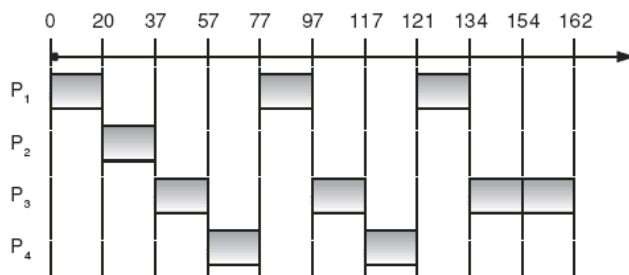
- **First Come, First Served (FCFS)**
- **Shortest Job First (SJF)**
 - preemptive – uvek će završiti tekući proces, bez obzira na to kakav se novi proces pojavio u redu čekanja.
 - non-preemptive = Shortest Remaining Time First (SRTF) - ukoliko je vreme potrebno za izvršenje novog procesa kraće od vremena potrebnog za završetak aktivnosti tekućeg procesa, procesor će biti dodeljen novom procesu.

Proces	Vreme nailaska u sistem	Vreme izvršavanja
P1	0	7
P2	2	4
P3	4	1
P4	5	4



- **SJF w/ idle time.** Vreme čekanja na procese (*idle time*) je vremenski interval u kom planer poslova niskog nivoa čeka da u sistem stigne još nekoliko procesa.

- vreme poveñati i broj procesa u redu, pa ñe se samim tim poboããati i performanse rasporeoeivaãa.
- **Priority Scheduling**
 - preemptive
 - nonpreemptive
- **Round Robin (RR)** = FCFS sa pretprañnjenjem. Primer: P1, P2, P3, i P4 (vremena izvršavanja 53, 17, 68 i 24 vremenske jedinice) došla u READY queue u istom trenutku; formirana kružna lista P1->P2->P3->P4.



Primeri

1. Dat je sledeći skup procesa čija su vremena izvršavanja na procesoru (*CPU-burst time*, izražena u milisekundama) i prioriteti dati u sledećoj tabeli:

Proces	Vreme izvršavanja	Prioritet
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

Procesi su u sistem naišli u poretku P1, P2, P3, P4, P5, svi približno u trenutku $t=0$.

a. Nacrtati *Gantt*-ove karte dodele procesora ukoliko se raspoređivanje vrši na osnovu sledećih algoritama: FCFS, SJF bez pretprañnjenja, raspoređivanje na osnovu prioriteta bez pretprañnjenja (manji broj znači veći prioritet) i RR sa kvantomom $Q=1$.

b. Odrediti vreme potrebno za kompletiranje procesa (*turnaround time*) za svaki proces (za sve gore pomenute algoritme).

c. Odrediti vreme čekanja za svaki proces i srednje vreme čekanja (za sve gore pomenute algoritme). Za koji je algoritam srednje vreme čekanja najmanje ?

Kašnjenje dispečera zanemariti.

2. Dat je sledeći skup procesa čija su vremena nailaska u sistem (*arrival time*) i vremena izvršavanja na procesoru, izražena u milisekundama data u sledećoj tabeli:

Process	Vreme nailaska	Vreme izvršavanja
P1	0	8
P2	0.4	4
P3	1	1

Nacrtati *Gantt*-ovu kartu i odrediti srednje vreme potrebno za kompletiranje procesa i srednje vreme čekanja ukoliko se raspoređivanje procesa obavlja po:

- a. FCFS algoritmu,
 - b. SJF algoritmu (bez pretpražnjenja),
 - c. SJF algoritmu sa vremenom čekanja na procese (*idle time*) $t_{idle}=1$,
 - d. SRTF (*shortest remaining time first*) algoritmu, odnosno SJF algoritmu sa pretpražnjenjem.
- Kašnjenje dispečera zanemariti.

3. Dat je sledeći skup procesa čija su vremena nailaska u sistem (*arrival time*) i vremena izvršavanja na procesoru, izražena u milisekundama data u sledećoj tabeli:

<u>Process</u>	<u>Vreme nailaska</u>	<u>Vreme izvršavanja</u>
P1	0	7
P2	2	4
P3	4	1
P4	5	4

Nacrtati *Gantt*-ovu kartu i odrediti srednje vreme potrebno za kompletiranje procesa i srednje vreme čekanja ukoliko se raspoređivanje procesa obavlja po:

- a. SJF algoritmu.
- b. SRTF algoritmu.

Kašnjenje dispečera zanemariti.

4. Četiri procesa su u trenutku $t=0$ ušli u red čekanja na procesor u sledećem redosledu: P1, P2, P3, P4. Vremena izvršavanja na procesoru za ova četiri procesa su 6, 3, 1 i 7 vremenskih jedinica, respektivno. Ukoliko se raspoređivanje procesa vrši prema Round Robin algoritmu sa kvantomom (a) $Q=1$, (b) $Q=2$, (c) $Q=3$, (d) $Q=4$, (e) $Q=5$, (f) $Q=6$, (g) $Q=7$
- nacrtati *Gantt*-ovu kartu i odrediti srednje vreme izvršavanja na procesoru i srednje vreme čekanja (kašnjenje dispečera zanemariti),
 - odrediti koliko puta je obavljena zamena konteksta i koliko je ukupno vremena potrebno da sva četiri procesa završe aktivnosti (kašnjenje dispečera je $dl=0.01$ vremenskih jedinica).

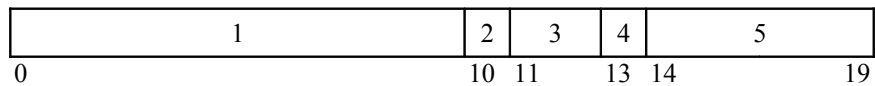
+ programerski zadaci:

1. U trenutku $t=0$ u sistem stižu tri procesa sa vremenima izvršavanja t_1 , t_2 i t_3 vremenskih jedinica koje zadaje korisnik na standardnom ulazu. Napišite C/C++ program koji će prema RR algoritmu da rasporedi ove procese za kvantum koji unosi korisnik i da raspored dodele procesora prikaže na standardnom izlazu.
2. Proširite prethodni program tako da odredi waiting i turnaround time za sve procese i da ih prikaže na glavnom izlazu. Program treba da odredi srednji waiting i turnaround time.
3. Proširite dalje program tako da za različite vremenske kvantume računa srednji waiting i turnaround time i da određuje kvantum za koji je najmanji waiting time.
4. Proširite program 1 tako da se korisniku nudi i mogućnost da za sva tri procesa zadaje i vreme kad je proces stigao u sistem.
5. Proširite program 4 tako da se rasporedi proizvoljan broj procesa (korisnik unosi broj procesa i za svaki proces arrival time i vreme izvršavanja).

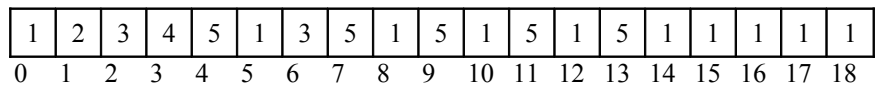
1.

(a) *Gantt*tove karte dodele procesora:

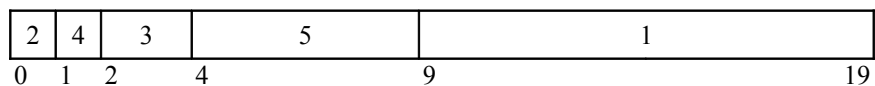
FCFS:



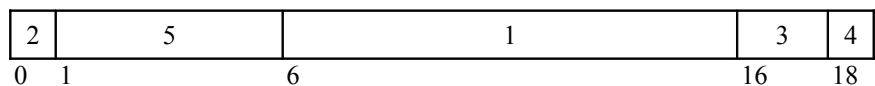
RR (Q = 1):



SJF (bez pretpražnjenja):



Rapoređivanje na osnovu prioriteta (bez pretpražnjenja):



(b) Vreme potrebno za kompletiranje procesa:

	FCFS	RR	SJF	Prioritetno
P1	10	19	19	16
P2	11	2	1	1
P3	13	7	4	18
P4	14	4	2	19
P5	19	14	9	6

(c) Vreme čekanja

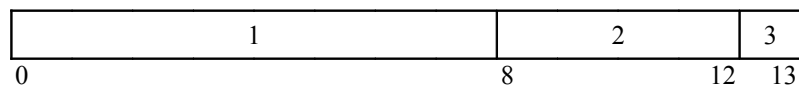
	FCFS	RR	SJF	Prioritetno
P1	0	9	9	6
P2	10	1	0	0
P3	11	5	2	16
P4	13	3	1	18
P5	14	9	4	1
sr.vreme:	9.6	5.4	3.2	8.2

Srednje vreme čekanja se računa kao srednja vrednost vremena čekanja za sve procese.

Srednje vreme čekanja je najmanje u slučaju SJF bez pretpražnjenja.

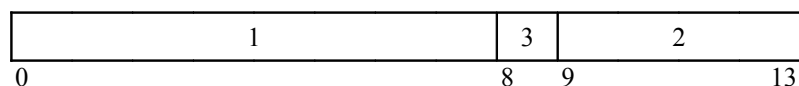
2.

(a) *Gantt*-ova karta dodele procesora - FCFS algoritam:



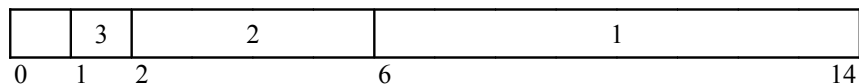
	vreme kompletiranja	vreme čekanja
P1	8	0
P2	$12 - 0.4 = 11.6$	$8 - 0.4 = 7.6$
P3	$13 - 1 = 12$	$12 - 1 = 11$
srednje vreme:	$\frac{8+11.6+12}{3} = 10.53$	$\frac{0+7.6+11}{3} = 6.2$

(b) *Gantt*-ova karta dodele procesora - SJF algoritam:



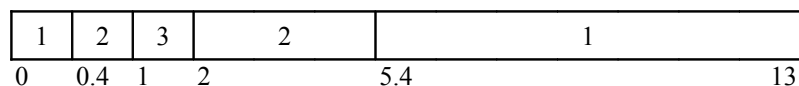
	vreme kompletiranja	vreme čekanja
P1	8	0
P2	$13 - 0.4 = 12.6$	$9 - 0.4 = 8.6$
P3	$9 - 1 = 8$	$8 - 1 = 7$
srednje vreme:	$\frac{8+12.6+8}{3} = 9.53$	$\frac{0+8.6+7}{3} = 5.2$

(c) *Gantt*-ova karta dodele procesora - SJF algoritam sa vremenom čekanja na procese (*idle time*) $t_{idle}=1$.



	vreme kompletiranja	vreme čekanja
P1	14	6
P2	$6 - 0.4 = 5.6$	$2 - 0.4 = 1.6$
P3	$2 - 1 = 1$	0
srednje vreme:	$\frac{14+5.6+1}{3} = 6.86$	$\frac{6+1.6+0}{3} = 2.53$

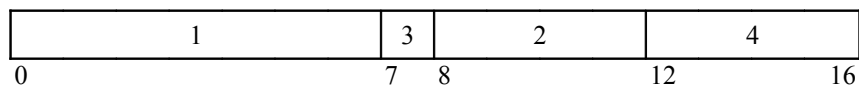
(d) *Gantt*-ova karta dodele procesora - SRTF algoritam



	vreme kompletiranja	vreme čekanja
P1	13	$5.4 - 0.4 = 5$
P2	$5.4 - 0.4 = 5$	$2 - 1 = 1$
P3	$2 - 1 = 1$	0
srednje vreme:	$\frac{13+5+1}{3} = 6.33$	$\frac{5+1+0}{3} = 2$

3.

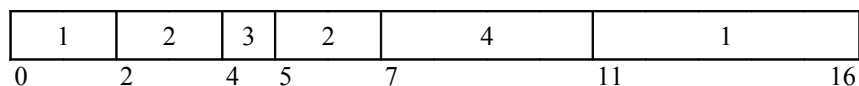
(a) *Gantt*-ova karta dodele procesora - SJF algoritam:



Vremena potrebna za kompletiranje procesa i vremena čekanja su:

	vreme kompletiranja	vreme čekanja
P1	7	0
P2	$12 - 2 = 10$	$8 - 2 = 6$
P3	$8 - 4 = 4$	$7 - 4 = 3$
P4	$16 - 5 = 7$	$12 - 5 = 7$
srednje vreme:	7	4

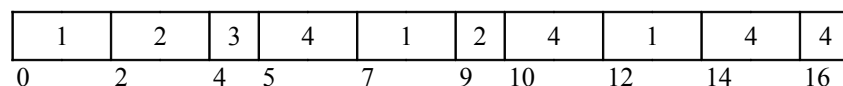
(b) *Gantt*-ova karta dodele procesora - SRTF algoritam:



Vremena potrebna za kompletiranje procesa i vremena čekanja su:

	vreme kompletiranja	vreme čekanja
P1	16	$11 - 2 = 9$
P2	$7 - 2 = 5$	$5 - 4 = 1$
P3	$5 - 4 = 1$	0
P4	$11 - 5 = 6$	$7 - 5 = 2$
srednje vreme:	7	3

4. *Gantt*-ova karta dodele procesora za slučaj RR, Q=2:



Vremena potrebna za kompletiranje procesa i vremena čekanja su:

	vreme kompletiranja	vreme čekanja
P1	14	$5 + 3 = 8$
P2	10	$2 + 5 = 7$
P3	5	4
P4	17	$5 + 3 + 2 = 10$
srednje vreme:	11.5	7.25

Obavljeno je 9 zamena konteksta, tako da je ukupno vreme potrebno da sva četiri procesa završe aktivnosti: $t = 17 + 9 \cdot 0.01 = 17.09$.