ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

Χρονοπρογραμματισμός
Εργαστηριακές Ασκήσεις

Υλικό από:
Κ Διαμαντάρας, Λειτουργικά Συστήματα, Τμήμα Πληροφορικής ΤΕΙΘ

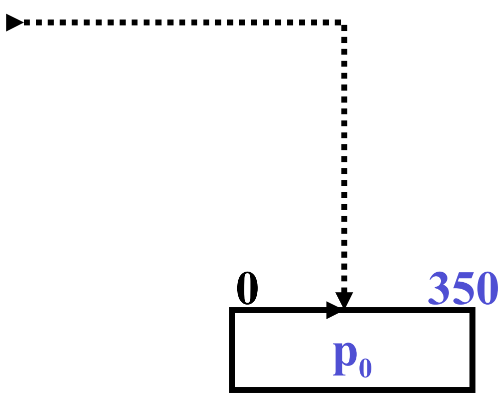
Σύνθεση
Κ.Γ. Μαργαρίτης, Τμήμα Εφαρμοσμένης Πληροφορικής, Πανεπιστήμιο Μακεδονίας

Παραδείγματα - Συμβολισμοί

- $P = \{p_i \mid 0 \le i < n\}$ = σύνολο διεργασιών
- $S(p_i) \in \{Running, Ready, Blocked\}$
- $\tau(p_i)$ = *Service Time*
- $W(p_i)$ = *Wait Time*
- $T_{TRnd}(pi)$ = *Turnaround Time*

First-Come-First-Served

i	$\tau(p_i)$
0	350
1	125
2	475
3	250
4	75

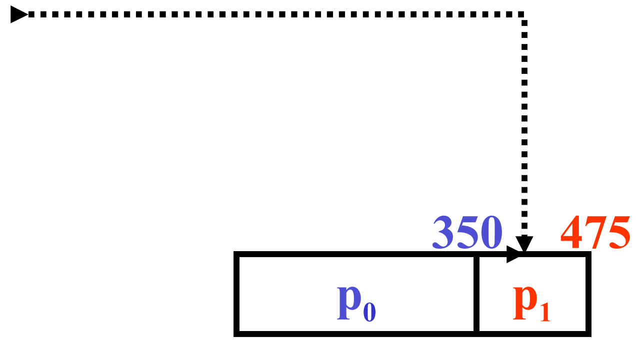


$T_{TRnd}(p_0) = \tau(p_0) = 350$

$W(p_0) = 0$

First-Come-First-Served

i	$\tau(p_i)$
0	350
1	125
2	475
3	250
4	75



$T_{TRnd}(p_0) = \tau(p_0) = 350$

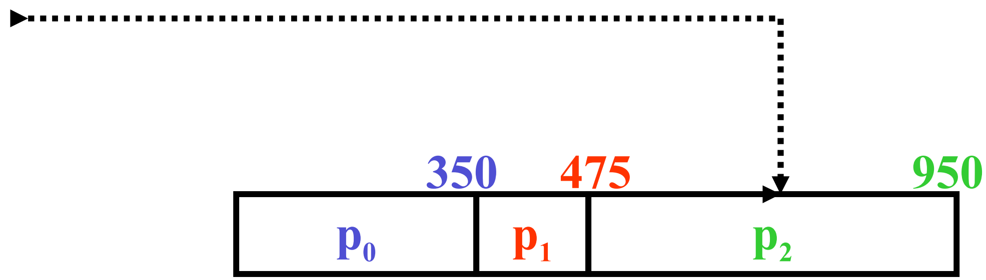
$T_{TRnd}(p_1) = (\tau(p_1) + T_{TRnd}(p_0)) = 125+350 = 475$

$W(p_0) = 0$

$W(p_1) = T_{TRnd}(p_0) = 350$

First-Come-First-Served

i	$\tau(p_i)$
0	350
1	125
2	475
3	250
4	75



$T_{TRnd}(p_0) = \tau(p_0) = 350$

$T_{TRnd}(p_1) = (\tau(p_1) + T_{TRnd}(p_0)) = 125+350 = 475$

$T_{TRnd}(p_2) = (\tau(p_2) + T_{TRnd}(p_1)) = 475+475 = 950$

$W(p_0) = 0$

$W(p_1) = T_{TRnd}(p_0) = 350$

$W(p_2) = T_{TRnd}(p_1) = 475$

First-Come-First-Served

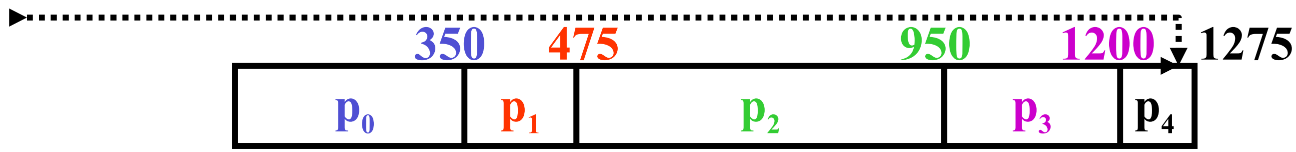
i	$\tau(p_i)$
0	350
1	125
2	475
3	250
4	75

$$T_{TRnd}(p_0) = \tau(p_0) = 350$$
$$T_{TRnd}(p_1) = (\tau(p_1) + T_{TRnd}(p_0)) = 125+350 = 475$$
$$T_{TRnd}(p_2) = (\tau(p_2) + T_{TRnd}(p_1)) = 475+475 = 950$$
$$T_{TRnd}(p_3) = (\tau(p_3) + T_{TRnd}(p_2)) = 250+950 = 1200$$

$$W(p_0) = 0$$
$$W(p_1) = T_{TRnd}(p_0) = 350$$
$$W(p_2) = T_{TRnd}(p_1) = 475$$
$$W(p_3) = T_{TRnd}(p_2) = 950$$

First-Come-First-Served

i	$\tau(p_i)$
0	350
1	125
2	475
3	250
4	75



$T_{TRnd}(p_0) = \tau(p_0) = 350$

$T_{TRnd}(p_1) = (\tau(p_1) + T_{TRnd}(p_0)) = 125+350 = 475$

$T_{TRnd}(p_2) = (\tau(p_2) + T_{TRnd}(p_1)) = 475+475 = 950$

$T_{TRnd}(p_3) = (\tau(p_3) + T_{TRnd}(p_2)) = 250+950 = 1200$

$T_{TRnd}(p_4) = (\tau(p_4) + T_{TRnd}(p_3)) = 75+1200 = 1275$

$W(p_0) = 0$

$W(p_1) = T_{TRnd}(p_0) = 350$

$W(p_2) = T_{TRnd}(p_1) = 475$

$W(p_3) = T_{TRnd}(p_2) = 950$

$W(p_4) = T_{TRnd}(p_3) = 1200$

FCFS Average Wait Time

i	$\tau(p_i)$
0	350
1	125
2	475
3	250
4	75

p_0	p_1	p_2	p_3	p_4
350	475	950	1200	1275

$T_{TRnd}(p_0) = \tau(p_0) = 350$

$T_{TRnd}(p_1) = (\tau(p_1) + T_{TRnd}(p_0)) = 125+350 = 475$

$T_{TRnd}(p_2) = (\tau(p_2) + T_{TRnd}(p_1)) = 475+475 = 950$

$T_{TRnd}(p_3) = (\tau(p_3) + T_{TRnd}(p_2)) = 250+950 = 1200$

$T_{TRnd}(p_4) = (\tau(p_4) + T_{TRnd}(p_3)) = 75+1200 = 1275$

$W(p_0) = 0$

$W(p_1) = T_{TRnd}(p_0) = 350$

$W(p_2) = T_{TRnd}(p_1) = 475$

$W(p_3) = T_{TRnd}(p_2) = 950$

$W(p_4) = T_{TRnd}(p_3) = 1200$

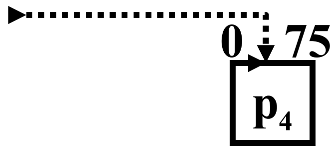
$T_{TRnd} = (350+475+950+1200+1275)/5 = 4250/5 = 850$

$W_{avg} = (0+350+475+950+1200)/5 = 2975/5 = 595$

- Ευκολία στην εφαρμογή
- Δεν έχει μεγάλη απόδοση

Shortest Job First

i	$\tau(p_i)$
0	350
1	125
2	475
3	250
4	75

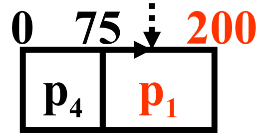


$T_{TRnd}(p_4) = \tau(p_4) = 75$

$W(p_4) = 0$

Shortest Job First Non-Preemptive

i	$\tau(p_i)$
0	350
1	125
2	475
3	250
4	75

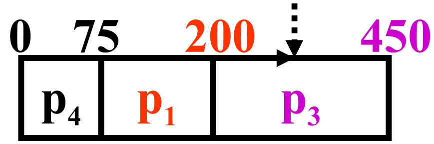


$T_{TRnd}(p_1) = \tau(p_1)+\tau(p_4) = 125+75 = 200$ $W(p_1) = 75$

$T_{TRnd}(p_4) = \tau(p_4) = 75$ $W(p_4) = 0$

Shortest Job First Non-Preemptive

i	$\tau(p_i)$
0	350
1	125
2	475
3	250
4	75



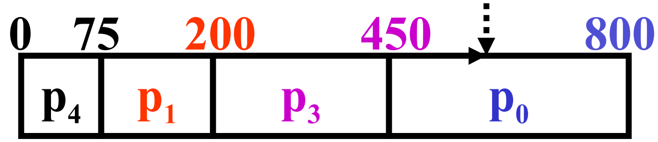
$T_{TRnd}(p_1) = \tau(p_1)+\tau(p_4) = 125+75 = 200$ $\quad W(p_1) = 75$

$T_{TRnd}(p_3) = \tau(p_3)+\tau(p_1)+\tau(p_4) = 250+125+75 = 450$ $\quad W(p_3) = 200$

$T_{TRnd}(p_4) = \tau(p_4) = 75$ $\quad W(p_4) = 0$

Shortest Job First Non-Preemptive

i	$\tau(p_i)$
0	350
1	125
2	475
3	250
4	75



$T_{TRnd}(p_0) = \tau(p_0)+\tau(p_3)+\tau(p_1)+\tau(p_4) = 350+250+125+75 = 800$ $W(p_0) = 450$

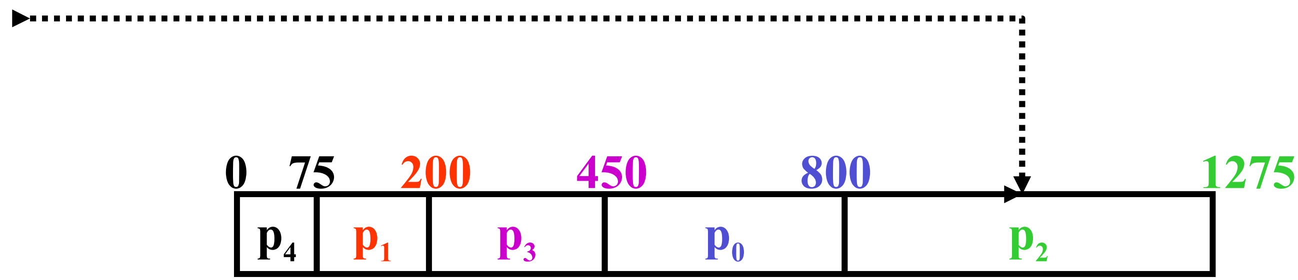
$T_{TRnd}(p_1) = \tau(p_1)+\tau(p_4) = 125+75 = 200$ $W(p_1) = 75$

$T_{TRnd}(p_3) = \tau(p_3)+\tau(p_1)+\tau(p_4) = 250+125+75 = 450$ $W(p_3) = 200$

$T_{TRnd}(p_4) = \tau(p_4) = 75$ $W(p_4) = 0$

Shortest Job First Non-Preemptive

i	$\tau(p_i)$
0	350
1	125
2	475
3	250
4	75



$T_{TRnd}(p_0) = \tau(p_0)+\tau(p_3)+\tau(p_1)+\tau(p_4) = 350+250+125+75 = 800$ $\quad W(p_0) = 450$

$T_{TRnd}(p_1) = \tau(p_1)+\tau(p_4) = 125+75 = 200$ $\quad W(p_1) = 75$

$T_{TRnd}(p_2) = \tau(p_2)+\tau(p_0)+\tau(p_3)+\tau(p_1)+\tau(p_4)$ $\quad W(p_2) = 800$

$= 475+350+250+125+75 = 1275$

$T_{TRnd}(p_3) = \tau(p_3)+\tau(p_1)+\tau(p_4) = 250+125+75 = 450$ $\quad W(p_3) = 200$

$T_{TRnd}(p_4) = \tau(p_4) = 75$ $\quad W(p_4) = 0$

Shortest Job First Non-Preemptive

- Ελαχιστοποιεί τον χρόνο αναμονής
- Γνώση εκ των προτέρων! των χρόνων εξυπηρέτησης

i	$\tau(p_i)$
0	350
1	125
2	475
3	250
4	75

0	75	200	450	800	1275
p_4	p_1	p_3	p_0	p_2	

$T_{TRnd}(p_0) = \tau(p_0)+\tau(p_3)+\tau(p_1)+\tau(p_4) = 350+250+125+75 = 800$

$T_{TRnd}(p_1) = \tau(p_1)+\tau(p_4) = 125+75 = 200$

$T_{TRnd}(p_2) = \tau(p_2)+\tau(p_0)+\tau(p_3)+\tau(p_1)+\tau(p_4)$
$= 475+350+250+125+75 = 1275$

$T_{TRnd}(p_3) = \tau(p_3)+\tau(p_1)+\tau(p_4) = 250+125+75 = 450$

$T_{TRnd}(p_4) = \tau(p_4) = 75$

$W(p_0) = 450$
$W(p_1) = 75$
$W(p_2) = 800$
$W(p_3) = 200$
$W(p_4) = 0$

$T_{TRnd} = (800+200+1275+450+75)/5$
$= 2800/5 = 560$

$W_{avg} = (450+75+800+200+0)/5$
$= 1525/5 = 305$

Σύγκριση δρομολογήσεων χωρίς προεκτόπιση

- First-Come-First-Served

$T_{TRnd} = (350+475+950+1200+1275)/5$ $= 4250/5 = 850$	$W_{avg} = (0+350+475+950+1200)/5$ $= 2975/5 = 595$

- Shortest Job First Non-preemtive

$T_{TRnd} = (800+200+1275+450+75)/5$ $= 2800/5 = 560$	$W_{avg} = (450+75+800+200+0)/5$ $= 1525/5 = 305$

Round Robin (TQ=50)

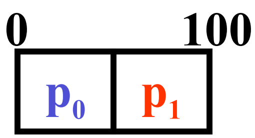
i	$\tau(p_i)$
0	350
1	125
2	475
3	250
4	75

0	50
p_0	

$W(p_0) = 0$

Round Robin (TQ=50)

i	$\tau(p_i)$
0	350
1	125
2	475
3	250
4	75

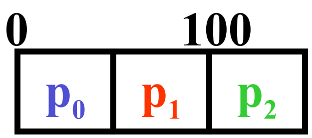


$W(p_0) = 0$

$W(p_1) = 50$

Round Robin (TQ=50)

i	$\tau(p_i)$
0	350
1	125
2	475
3	250
4	75



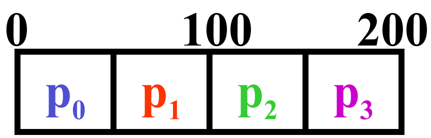
$W(p_0) = 0$

$W(p_1) = 50$

$W(p_2) = 100$

Round Robin (TQ=50)

i	$\tau(p_i)$
0	350
1	125
2	475
3	250
4	75



$W(p_0) = 0$

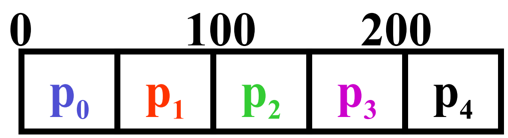
$W(p_1) = 50$

$W(p_2) = 100$

$W(p_3) = 150$

Round Robin (TQ=50)

i	$\tau(p_i)$
0	350
1	125
2	475
3	250
4	75



$W(p_0) = 0$

$W(p_1) = 50$

$W(p_2) = 100$

$W(p_3) = 150$

$W(p_4) = 200$

Round Robin (TQ=50)

i	$\tau(p_i)$
0	350
1	125
2	475
3	250
4	75

0 100 200 300

p_0	p_1	p_2	p_3	p_4	p_0

$W(p_0) = 0$

$W(p_1) = 50$

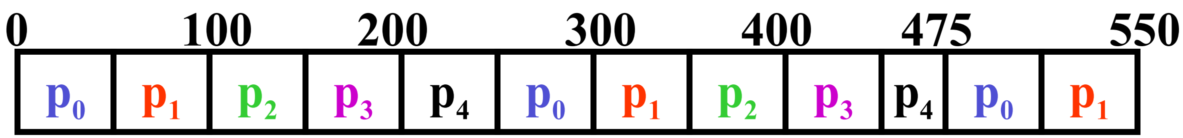
$W(p_2) = 100$

$W(p_3) = 150$

$W(p_4) = 200$

Round Robin (TQ=50)

i	$\tau(p_i)$
0	350
1	125
2	475
3	250
4	75



$T_{TRnd}(p_4) = 475$

$W(p_0) = 0$

$W(p_1) = 50$

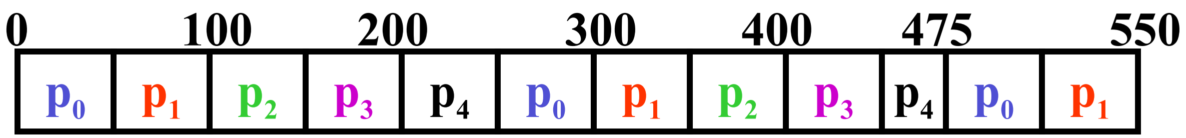
$W(p_2) = 100$

$W(p_3) = 150$

$W(p_4) = 200$

Round Robin (TQ=50)

i	$\tau(p_i)$
0	350
1	125
2	475
3	250
4	75



$T_{TRnd}(p_1) = 550$

$T_{TRnd}(p_4) = 475$

$W(p_0) = 0$

$W(p_1) = 50$

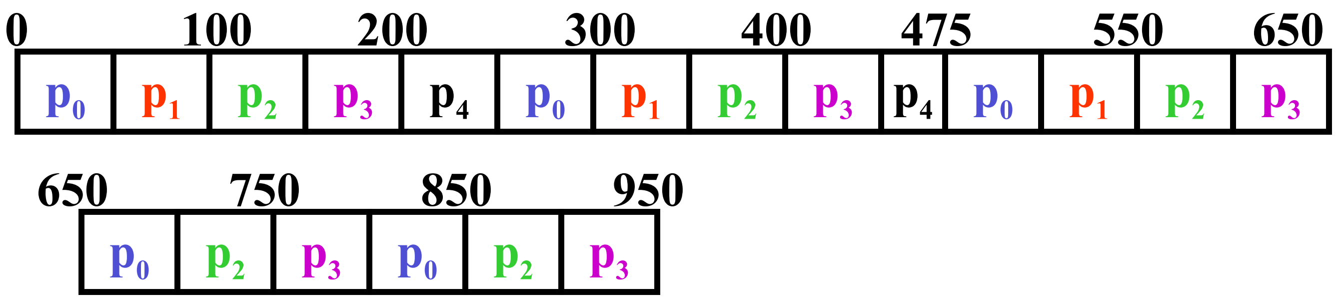
$W(p_2) = 100$

$W(p_3) = 150$

$W(p_4) = 200$

Round Robin (TQ=50)

i	$\tau(p_i)$
0	350
1	125
2	475
3	250
4	75



$T_{TRnd}(p_1) = 550$

$T_{TRnd}(p_3) = 950$

$T_{TRnd}(p_4) = 475$

$W(p_0) = 0$

$W(p_1) = 50$

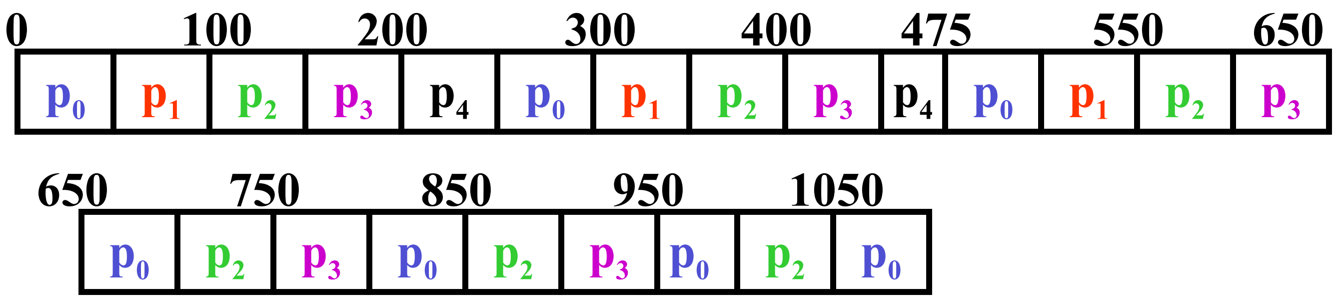
$W(p_2) = 100$

$W(p_3) = 150$

$W(p_4) = 200$

Round Robin (TQ=50)

i	$\tau(p_i)$
0	350
1	125
2	475
3	250
4	75



$T_{TRnd}(p_0) = 1100$

$T_{TRnd}(p_1) = 550$

$T_{TRnd}(p_3) = 950$

$T_{TRnd}(p_4) = 475$

$W(p_0) = 0$

$W(p_1) = 50$

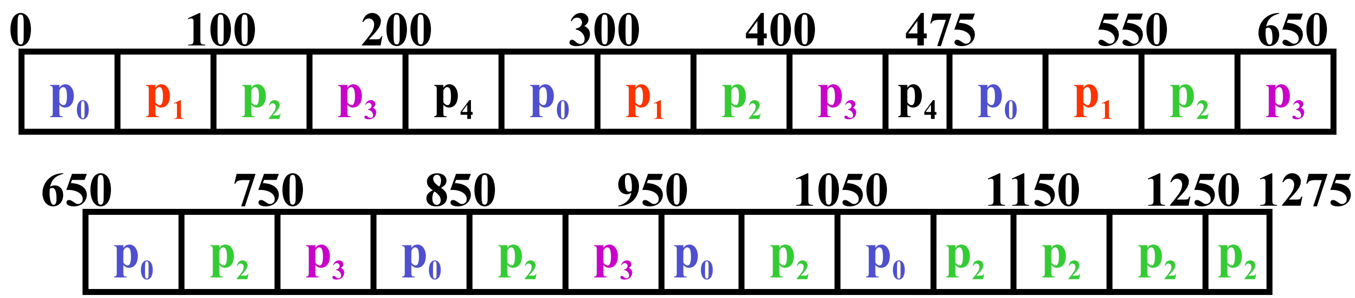
$W(p_2) = 100$

$W(p_3) = 150$

$W(p_4) = 200$

Round Robin (TQ=50)

i	$\tau(p_i)$
0	350
1	125
2	475
3	250
4	75



$T_{TRnd}(p_0) = 1100$

$T_{TRnd}(p_1) = 550$

$T_{TRnd}(p_2) = 1275$

$T_{TRnd}(p_3) = 950$

$T_{TRnd}(p_4) = 475$

$W(p_0) = 0$

$W(p_1) = 50$

$W(p_2) = 100$

$W(p_3) = 150$

$W(p_4) = 200$

Round Robin (TQ=50)

i	$\tau(p_i)$
0	350
1	125
2	475
3	250
4	75

- δίκαιη
- Χρησιμοποιείται ευρέως

0 100 200 300 400 475 550 650

p_0	p_1	p_2	p_3	p_4	p_0	p_1	p_2	p_3	p_4	p_0	p_1	p_2	p_3

650 750 850 950 1050 1150 1250 1275

p_0	p_2	p_3	p_0	p_2	p_3	p_0	p_2	p_0	p_2	p_2	p_2	p_2

$T_{TRnd}(p_0) = 1100$ $W(p_0) = 0$

$T_{TRnd}(p_1) = 550$ $W(p_1) = 50$

$T_{TRnd}(p_2) = 1275$ $W(p_2) = 100$

$T_{TRnd}(p_3) = 950$ $W(p_3) = 150$

$T_{TRnd}(p_4) = 475$ $W(p_4) = 200$

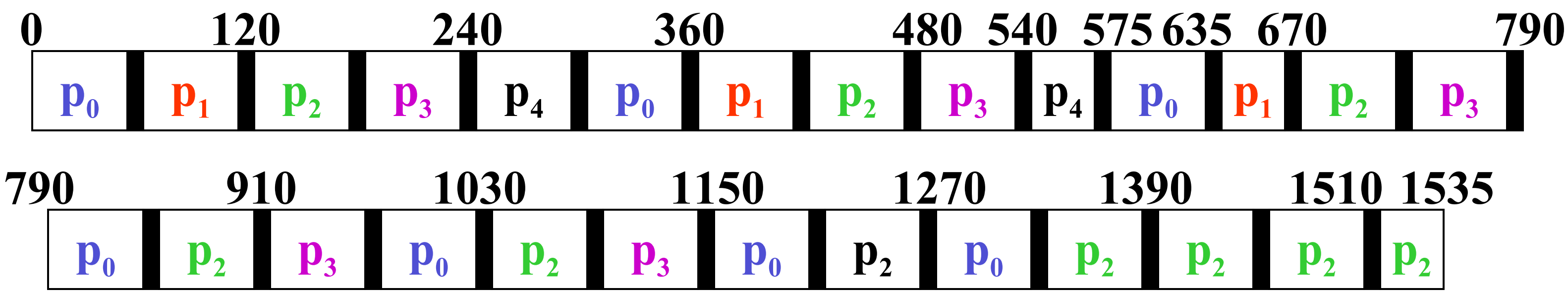
$$\boxed{T_{TRnd\_avg} = (1100+550+1275+950+475)/5 = 4350/5 = 870}$$

$$\boxed{W_{avg} = (0+50+100+150+200)/5 = 500/5 = 100}$$

RR with Overhead=10 (TQ=50)

i	$\tau(p_i)$	i	$\tau(p_i)$
0	350	2	475
1	125	3	250
		4	75

- Η επιβάρυνση είναι σημαντική
- Κάθε εναλλαγή κοστίζει κύκλους



$T_{TRnd}(p_0) = 1320$

$T_{TRnd}(p_1) = 660$

$T_{TRnd}(p_2) = 1535$

$T_{TRnd}(p_3) = 1140$

$T_{TRnd}(p_4) = 565$

$W(p_0) = 0$

$W(p_1) = 60$

$W(p_2) = 120$

$W(p_3) = 180$

$W(p_4) = 240$

$$T_{TRnd\_avg} = (1320+660+1535+1140+565)/5 = 5220/5 = 1044$$

$$W_{avg} = (0+60+120+180+240)/5 = 600/5 = 120$$

Άσκηση 1

Θεωρείστε το ακόλουθο σύνολο διεργασιών, στο οποίο το μήκος των CPU burst times είναι σε msec.

Υποθέστε ότι οι διεργασίες έχουν φθάσει με τη σειρά P1, P2, P3, P4, και P5, όλες τη χρονική στιγμή 0.

process	Burst time
P1	10
P2	1
P3	2
P4	1
P5	5

Σχεδιάστε το διάγραμμα εκτέλεσης για καθένα από τους αλγορίθμους : FCFS, SJF, RR.

Βρείτε για κάθε διεργασία τον turnaround time και τον waiting time καθώς και τις μέσες τιμές τους.

Λύση 1 (1)

Time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
FCFS:	P1	P1	P1	P1	P1	P1	P1	P1	P1	P1	P2	P3	P3	P4	P5	P5	P5	P5	P5
SJF:	P2	P4	P3	P3	P5	P5	P5	P5	P5	P1	P1	P1	P1	P1	P1	P1	P1	P1	P1
RR:	P1	P2	P3	P4	P5	P1	P3	P5	P1	P5	P1	P5	P1	P5	P1	P1	P1	P1	P1

Λύση 1 (2)

Turnaround time = waiting time + execution time

	FCFS	SJF	RR
P1	10	19	19
P2	11	1	2
P3	13	4	7
P4	14	2	4
P5	19	9	14
Average	13.4	7.0	9.2

Λύση 1 (3)

Waiting time = turnaround time - burst time

	FCFS	SJF	RR
P1	0	9	9
P2	10	0	1
P3	11	2	5
P4	13	1	3
P5	14	4	9
Average	9.6	3.2	5.4

Άσκηση 2

Θεωρείστε τις παρακάτω δύο διεργασίες. Κάθε διεργασία εκτελεί ένα CPU burst μετά ένα I/O burst, άλλο ένα CPU burst, άλλο ένα I/O burst και τερματίζει με ένα CPU burst. Τα μήκη των CPU burst και I/O burst χρόνων σε milliseconds δίνονται στον παρακάτω πίνακα:

Process	**CPU-burst1**	**I/O-burst1**	**CPU-burst2**	**I/O-burst2**	**CPU-burst3**	**Arrival**
P1	3	2	2	3	2	0
P2	2	2	3	3	2	1

Σχεδιάστε δύο διαγράμματα που θα δείχνουν την εκτέλεση των διεργασιών χρησιμοποιώντας τον αλγόριθμο Round Robin (RR) με quantum = 1 και quantum = 3. Αν ένα I/O burst λήξει τη στιγμη που συμβεί ένα CPU timeout, τότε εξυπηρετείται πρώτα η Ε/Ε.

Ποιος είναι ο waiting time κάθε διεργασίας σε κάθε περίπτωση; Ποια είναι η μέση τιμή; Ποιος είναι ο turn-around time κάθε διεργασίας σε κάθε περίπτωση; Ποια είναι η μέση τιμή;

Τι θα συμβεί αν αντιστοιχηθεί προτεραιότητα σε κάθε διεργασία και η προτεραιότητα της P2 είναι υψηλότερη της P1. Να εξετάσετε και τις δύο περιπτώσεις (quantum=1 και quantum=3).

Λύση 2 (1)

```
xxx   Process is running
---   Process is ready
III   Process is blocked on I/O device
iii   CPU idle

Time  0 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
q=1
P1:   xxx---xxx---xxxIIIIIIxxx---xxxIIIIIIIIIxxx---xxxTerm
P2:      xxx---xxxIIIIIIxxx---xxx---xxxIIIIIIIIIxxx---xxxTerm
CPU:                 iii               iiiiii

q=3
P1:   xxxxxxxxxIIIIIIxxxxxxIIIIIIIIIxxxxxxTerm
P2:      ------xxxxxxIIIIIIxxxxxxxxxIIIIIIIIIxxxxxxTerm
```

Λύση 2 (2)

```
xxx   Process is running
---   Process is ready
III   Process is blocked on I/O device

Time   0 1   2   3   4   5   6   7   8   9 10  11  12  13  14  15  16  17  18
q=1
P1:    xxx------xxxxxxIIIIII---xxxxxxIIIIIIIIIxxxxxxTerm
P2:       xxxxxxIIIIIxxxxxxxxxIIIIIIIIIxxxxxxTerm

q=3
P1:    xxx------xxxxxxIIIIII---xxxxxxIIIIIIIIIxxxxxxTerm
P2:       xxxxxxIIIIIxxxxxxxxxIIIIIIIIIxxxxxxTerm
```

Λύση 2 (3)

Waiting times:

```
(i)  q = 1:  P1:4;  P2:4;  Mean value: 4
(ii) q = 3:  P1:0;  P2:2;  Mean value: 1
```

Turn around times:

```
(i)  q = 1:  P1:16;  P2:16;  Mean value:16
(ii) q = 3:  P1:12;  P2:14;  Mean value:13
```

Άσκηση 3

Να σχεδιάσετε το διάγραμμα Gantt για καθένα από τους παρακάτω αλγορίθμους: FCFS, RR με quantum=2 χρονικές μονάδες, Preemptive και Nonpreemptive SJF και να υπολογίσετε τον μέσο χρόνο επιστροφής για τον κάθε αλγόριθμο.

Process	Arrival time	Burst time	Priority
A	0	3	2
B	2	6	4
C	4	4	1
D	6	5	5
E	8	2	3

Λύση 3(1) FCFS : Gantt Chart

A	B	C	D	E

0 3 9 13 18 20

$$\text{Ave. TAT} = \frac{3 + 9 + 13 + 18 + 20}{5} = \frac{63}{5}$$

$$= 12.6 \text{ time units}$$

Λύση 3(2) RR : Time Quantum = 2 time units

A	B	C	D	E	A	B	C	D	B	D

0 2 4 6 8 10 11 13 15 17 19 20

$$\text{Ave. TAT} = \frac{11 + 19 + 15 + 20 + 10}{5} = \frac{75}{5}$$

$$= 15 \text{ time units}$$

Λύση 3(3) Nonpreemptive SJF: Gantt Chart

E	A	C	D	B

0 2 5 9 14 20

$$\text{Ave. TAT} = \frac{2 + 5 + 9 + 14 + 20}{5} = \frac{50}{5}$$

$$= 10 \text{ time units}$$

Λύση 3(4) Preemptive SJF: Gantt Chart

A	A	B	C	C	E	B	D

0 2 3 4 6 8 10 15 20

$$\text{Ave. TAT} = \frac{(3 - 0) + (15 - 2) + (8 - 4) + (20 - 6) + (10 - 8)}{5}$$

$$= 7.2 \text{ time units}$$

΄Ασκηση 4

Πέντε batch διεργασίες (A – E), φθάνουν σε ένα υπολογιστικό κέντρο ακριβώς την ίδια στιγμή. Έχει υπολογιστεί ότι οι χρόνοι εκτέλεσής τους είναι 10, 6, 2, 4, και 8 χρονικές μονάδες. Οι προτεραιότητές τους είναι αντίστοιχα 3, 5, 2, 1, και 4, η μεγαλύτερη προτεραιότητα είναι η 5. Για κάθε έναν από τους παρακάτω αλγόριθμους να βρεθεί ο μέσος χρόνος επιστροφής (mean turnaround time). Να αγνοηθεί η επιβάρυνση λόγω εναλλαγής πλαισίου διεργασίας.

(a) Round robin.
(b) First-come, first-served (εκτέλεση κατά σειρά 10, 6, 2, 4, 8).
(c) Shortest job first.

Για την περίπτωση (a), υποθέστε ότι το σύστημα είναι πολυπρογραμματισμού και ότι κάθε διεργασία χρησιμοποιεί την CPU εξ ίσου με τις υπόλοιπες. Για τις περιπτώσεις (b) και (c) υποθέστε ότι κάθε στιγμή μόνον μια διεργασία μπορεί να εκτελείται, μέχρι να τερματιστεί. Όλες οι διεργασίες είναι εξ ολοκλήρου CPU bound.

Λύση 4 (1)

Σχόλια για Round Robin

Κατά τη διάρκεια των πρώτων 10 λεπτών κάθε διεργασία αποκτά τον έλεγχο της CPU για το 1/5 του χρόνου. Στο τέλος του 8ου λεπτού, η C τερματίζεται.

Κατά τη διάρκεια των επόμενων 8 λεπτών κάθε διεργασία αποκτά τον έλεγχο της CPU για το ¼ του χρόνου μετά από το οποίο η διεργασία D τερματίζεται.

Τότε καθεμία από τις εναπομένουσες διεργασίες αποκτά τον έλεγχο της CPU για το 1/3 του χρόνου για 6 λεπτά, μέχρι να τερματιστεί η διεργασία B κλπ.

Οι χρόνοι ολοκλήρωσης είναι : A: 30, B:23, C:8, D:17 και E:28, με μέση τιμή 21.2 λεπτά.

Λύση 4 (2)

Time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Job	A	B	C	D	E	A	B	C	D	E	A	B	D	E	A	B	D	E	A	B	E	A	B	E	A	E	A	E	A	A
A	1					2					3				4				5			6			7		8		9	**10**
B		1					2					3				4				5			**6**							
C			1					**2**																						
D				1					2				3				**4**													
E					1					2				3				4			5			6		7		**8**		

5. Να συγκρίνετε τους αλγορίθμους FCFS και RR ως προς το αν μεροληπτούν και σε τι βαθμό υπέρ των μικρής διάρκειας διεργασιών.

6. Ποιο είναι το πλεονέκτημα της ύπαρξης διαφορετικού μεγέθους quantum χρόνου στα επίπεδα ενός συστήματος με ουρές πολλαπλών επιπέδων (multilevel queueing system) που όλες χρησιμοποιούν τον αλγόριθμο RR;

Λύση: Οι διεργασίες που χρειάζονται συχνά εξυπηρέτηση Ε/Ε, δηλαδή αλληλεπιδραστικές διεργασίες όπως είναι οι editors, μπορούν να βρίσκονται σε μια ουρά με μικρό quantum χρόνου. Οι διεργασίες που δεν απαιτούν συχνά εξυπηρέτηση μπορούν να βρίσκονται σε μια ουρά με μεγαλύτερο quantum, απαιτώντας λιγότερες εναλλαγές πλαισίου για να ολοκληρώσουν την επεξεργασία τους, εξασφαλίζοντας έτσι περισσότερο αποδοτική χρήση του συστήματος.

7. Ποια είναι η επίδραση της συνεχούς αύξησης του quantum χρόνου στον αλγόριθμο δρομολόγησης Round Robin ?

Ο μέσος χρόνος αναμονής μειώνεται

Ο μέσος χρόνος επιστροφής αυξάνεται

Ο αλγόριθμος συμπεριφέρεται ακριβώς το ίδιο με τον FCFS

Όλα τα παραπάνω

Κανένα από τα παραπάνω

8. Υποθέστε ότι ένα σύστημα έχει μια CPU και μέσο ρυθμό αφίξεων διεργασιών 20 jobs/minute. Κάθε διεργασία έχει μέσο χρόνο εξυπηρέτησης 2 seconds. Ποιο είναι το αναμενόμενο ποσοστό χρόνου κατά το οποίο η CPU είναι απασχολημένη;

40%

66.7%

83.3%

87.5%

Κανένα από τα ανωτέρω

9. Μια πολιτική δρομολόγησης προσπαθεί να βελτιστοποιήσει :
CPU utilization
Average wait time
Average turnaround time
Average throughput rate
Όλα τα παραπάνω

10. Ποιο από τα παρακάτω είναι αληθές για τον αλγόριθμο FCFS ;
Ελαχιστοποιεί τον χρόνο αναμονής
Μεγιστοποιεί την διεκπεραίωση (throughput)
Μεγιστοποιεί την δικαιοσύνη κατά τη χρήση της CPU.
Τίποτε από τα παραπάνω
Όλα τα παραπάνω

Άσκηση 11

Για τα παρακάτω προβλήματα, χρησιμοποιείστε τα εξής δεδομένα για τις διεργασίες (όλες οι διεργασίες έχουν ταυτόχρονη άφιξη στη χρονική στιγμή 0):

process	Service time
P0	150
P1	30
P2	130
P3	80
P4	90

FCFS, the average wait time?

96

180

206

302

Κανένα από τα παραπάνω

FCFS, the average turnaround time?

96

206

302

480

Κανένα από τα παραπάνω

SJN, the average wait time?

96

134

230

330

Κανένα από τα παραπάνω

SJN, the average turnaround time?

96

134

240

330

Κανένα από τα παραπάνω

RR scheduling, time quantum=40

no scheduling overhead

the average wait time?

40

74

164

188

Κανένα από τα παραπάνω

RR scheduling, time quantum = 40

overhead =10

the average wait time?

40

74

86

94

Κανένα από τα παραπάνω

RR scheduling, time quantum=40

no scheduling overhead

the average turnaround time?

40

74

172

258

Κανένα από τα παραπάνω

RR scheduling, time quantum = 40

overhead =10

the average turnaround time?

296

354

378

442

Κανένα από τα παραπάνω

Άσκηση 12

Ένας χρονοπρογραμματιστής χρησιμοποιεί αλγόριθμο RR με προτεραιότητες. Οι νέες διεργασίες λαμβάνουν ένα αρχικό quantum μεγέθους q. Κάθε φορά που μια διεργασία χρησιμοποιεί όλο το quantum χωρίς να ανασταλεί, το νέο της quantum γίνεται διπλάσιο από το τρέχον. Αν μια διεργασία ανασταλεί πριν λήξει το quantum της, το νέο της quantum είναι πάλι q. Υποθέστε επίσης ότι κάθε διεργασία απαιτεί ένα πεπερασμένο χρονικό διάστημα στην CPU.

Υποθέστε ότι ο δρομολογητής δίνει μεγαλύτερη προτεραιότητα στις διεργασίες που έχουν μεγάλα quanta. Είναι πιθανή η παρατεταμένη στέρηση στο σύστημα;

Υποθέστε ότι ο δρομολογητής δίνει μεγαλύτερη προτεραιότητα στις διεργασίες που έχουν μικρότερα quanta. Είναι πιθανή η παρατεταμένη στέρηση στο σύστημα;

Λύση 12

Αν μια διεργασία χρησιμοποιεί όλο το quantum ο χρονοπρογραμματιστής θα της επιτρέπει να εκτελείται μέχρι να τελειώσει ή να ανασταλεί μια και έχει μεγαλύτερη προτεραιότητα από τις υπόλοιπες. Αν ανασταλεί θα αναμένει στην ουρά. Δηλαδή είναι ένας κανονικός αλγόριθμος RR, που δίνει σε κάποιες διεργασίες ένα μεγαλύτερο quantum χρόνου. Δεν προκύπτει παρατεταμένη στέρηση.

Υποθέστε ότι μια διεργασία P1 χρησιμοποιεί το συνολικό quantum q, και ένα σταθερό σύνολο νέων διεργασιών φθάνει στο σύστημα, τέτοιο ώστε να υπάρχει πάντοτε μια τουλάχιστον διεργασίες έτοιμη προς εκτέλεση. Αυτές οι διεργασίες θα έχουν μεγαλύτερη προτεραιότητα από την P1 και ο χρονοπρογραμματιστής θα επιλέγει πάντοτε για εκτέλεση από αυτές παρά την P1 . Η P1 θα είναι σε παρατεταμένη στέρηση.

Άσκηση 13

Ένα λ.σ. χρησιμοποιεί αλγόριθμο εκ περιτροπής. Το κβάντο χρόνου είναι Q και ο χρόνος που απαιτείται για την θεματική εναλλαγή είναι C. Κατά τη διάρκεια μιας χρονικής περιόδου T>> Q+C, υπάρχουν στο σύστημα N έτοιμες διεργασίες και κατά τη διάρκεια αυτής της περιόδου καμία νέα δεν ξεκινά ούτε κάποια από τις έτοιμες τερματίζεται. Υποθέστε επίσης ότι όλες οι διεργασίες χωρούν στη κύρια μνήμη. Να βρείτε και να δικαιολογήσετε ποιες από τις παρακάτω προτάσεις είναι αληθείς και κάτω από ποιες προϋποθέσεις :

1. όλες οι διεργασίες λαμβάνουν ίσο μερίδιο χρόνου της CPU.
2. αν η διεργασία P είναι έτοιμη, τότε θα δρομολογηθεί για εκτέλεση εντός χρόνου NC+(N-1)Q.
3. αν η διεργασία P έχει εξαντλήσει τον χρόνο καταιγισμού της θα πρέπει να περιμένει τουλάχιστον χρόνο NC + (N-1)Q.
4. αν η διεργασία P εκτελείται και μπλοκαριστεί, και η διεργασία W είναι γενικά έτοιμη τότε η W θα εκτελεστεί πριν η P εκτελεστεί ξανά.
5. η αποδοτικότητα του συστήματος , δηλαδή το ποσοστό του χρόνου κατά τον οποίο η CPU εκτελεί μια διεργασία, είναι πάντοτε ίση με Q/(Q+C).

Άσκηση 14

Υποθέστε ότι οι παρακάτω διεργασίες εισέρχονται σε ένα σύστημα στους χρόνους που αναγράφονται και απαιτούν τους αντίστοιχους χρόνους εκτέλεσης:

ΔΙΕΡΓΑΣΙΑ	ΧΡΟΝΟΣ ΕΚΚΙΝΗΣΗΣ	ΧΡΟΝΟΣ ΕΚΤΕΛΕΣΗΣ
A	0	10
B	2	3
C	4	8
D	7	6
E	10	1
F	17	4

Να σχεδιάσετε το διάγραμμα εκτέλεσης των διεργασιών και να βρείτε τους χρόνους επιστροφής (turnaround time) κάθε διεργασίας καθώς και τους μέσους χρόνους επιστροφής για τους εξής αλγορίθμους δρομολόγησης :

1. Η διεργασία με τον μικρότερο εναπομείναντα χρόνο πρώτη, με προεκτόπιση

2. εκ περιτροπής χωρίς κόστος θεματικής εναλλαγής πλαισίου. Το κβάντο χρόνου δεν είναι σταθερό αλλά εξαρτάται από το πλήθος των διεργασιών που βρίσκονται στο σύστημα κάθε στιγμή και :

- είναι αντιστρόφως ανάλογος του πλήθους των ενεργών διεργασιών
- μοιράζεται σε όλες εξ ίσου

π.χ. αν οι διεργασίες P και Q είναι ενεργές κατά τη χρονική στιγμή 4 και η P χρειάζεται ακόμη 4 seconds για να τελειώσει και η Q 3 seconds, τότε τη χρονική στιγμή 5 η P θα χρειάζεται ακόμη 3.5 seconds για να τελειώσει και η Q 2.5 seconds.