

ΤΜΑ478: Λειτουργικά Συστήματα

Χειμερινό Εξάμηνο 2022-23

# Προσομοίωση του υποσυστήματος χρονοπρογραμματισμού ενός Λ.Σ. σε γλώσσα C

---

Εξαμηνιαία Εργασία

Διδάσκουσα: Ελευθερία Κατσίρη

Γουγούσης Παναγιώτης 58198

Δερμεντζής Ιωάννης 58042



ΔΗΜΟΚΡΙΤΕΙΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΑΣ

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών  
Department of Electrical and Computer Engineering

## Μέρος Α : Θεωρητική εξοικείωση

Σκοπός του πρώτου μέρους της εργασίας είναι να υπολογίσετε και να συγκρίνετε την πρακτική συμπεριφορά τεσσάρων από τους πιο δημοφιλείς αλγόριθμους χρονοδρομολόγησης διαδικασιών (First Come First Serve, Preemptive Priority, Shortest Remaining Time First και Round Robin), με βάση το ακόλουθο σύνολο διαδικασιών:

Διαδικασία	Χρόνος Άφιξης	Χρόνος Ολοκλήρωσης	Προτεραιότητα
P <sub>1</sub>	0	5	3
P <sub>2</sub>	1	6	5
P <sub>3</sub>	3	2	2
P <sub>4</sub>	9	4	1
P <sub>5</sub>	12	3	4

### 1.1 First Come First Serve (FCFS) χρονοδρομολόγηση

Gantt	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	P <sub>1</sub>	P <sub>1</sub>	P <sub>1</sub>	P <sub>1</sub>	P <sub>1</sub>	P <sub>2</sub>	P <sub>2</sub>	P <sub>2</sub>	P <sub>2</sub>	P <sub>2</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>4</sub>	P <sub>4</sub>	P <sub>4</sub>	P <sub>5</sub>	P <sub>5</sub>	P <sub>5</sub>

	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>
Χρόνος αναμονής	0	4	8	4	5
Συνολικός χρόνος	5	11	13	17	20
Χρόνος διεκπεραίωσης	5	10	10	8	8
Αριθμός εναλλαγών	0	0	0	0	0

**1.2 Priority based Preemptive (PP) χρονοδρομολόγηση**

Gantt	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	P <sub>1</sub>	P <sub>1</sub>	P <sub>1</sub>	P <sub>3</sub>	P <sub>3</sub>	P <sub>1</sub>	P <sub>1</sub>	P <sub>2</sub>	P <sub>2</sub>	P <sub>4</sub>	P <sub>4</sub>	P <sub>4</sub>	P <sub>4</sub>	P <sub>5</sub>	P <sub>5</sub>	P <sub>5</sub>	P <sub>2</sub>	P <sub>2</sub>	P <sub>2</sub>	P <sub>2</sub>

	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>
Χρόνος αναμονής	2	13	0	0	1
Συνολικός χρόνος	7	20	5	13	16
Χρόνος διεκπεραίωσης	7	19	2	4	4
Αριθμός εναλλαγών	1	1	0	0	0

**1.3 Shortest Remaining Time First (SRTF) χρονοδρομολόγηση**

Gantt	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	P <sub>1</sub>	P <sub>1</sub>	P <sub>1</sub>	P <sub>1</sub>	P <sub>1</sub>	P <sub>3</sub>	P <sub>3</sub>	P <sub>2</sub>	P <sub>2</sub>	P <sub>2</sub>	P <sub>2</sub>	P <sub>2</sub>	P <sub>2</sub>	P <sub>5</sub>	P <sub>5</sub>	P <sub>5</sub>	P <sub>4</sub>	P <sub>4</sub>	P <sub>4</sub>	P <sub>4</sub>

	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>
Χρόνος αναμονής	0	6	2	7	1
Συνολικός χρόνος	5	13	7	20	16
Χρόνος διεκπεραίωσης	5	12	4	11	4
Αριθμός εναλλαγών	0	0	0	0	0

**1.4 Round Robin (RR) χρονοδρομολόγηση**

Gantt	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	P <sub>1</sub>	P <sub>1</sub>	P <sub>2</sub>	P <sub>2</sub>	P <sub>1</sub>	P <sub>1</sub>	P <sub>3</sub>	P <sub>3</sub>	P <sub>2</sub>	P <sub>2</sub>	P <sub>1</sub>	P <sub>4</sub>	P <sub>4</sub>	P <sub>2</sub>	P <sub>2</sub>	P <sub>5</sub>	P <sub>5</sub>	P <sub>4</sub>	P <sub>4</sub>	P <sub>5</sub>

	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>
Χρόνος αναμονής	6	8	3	6	5
Συνολικός χρόνος	11	15	8	19	20
Χρόνος διεκπεραίωσης	11	14	5	10	8
Αριθμός εναλλαγών	2	2	0	1	1

### 3.2 Έλεγχος (testing)

Θα εκτελέσουμε το πρόγραμμα βάζοντας ως είσοδο τα νούμερα από τις παραπάνω διεργασίες για τους αλγορίθμους FCFS, PP και SRTF. Παρατηρούμε ότι τα αποτελέσματα που μας δίνει το πρόγραμμα είναι ίδια με αυτά που υπολογίσαμε θεωρητικά.

```
Time: 1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20
FCFS: P1 P1 P1 P1 P1 P2 P2 P2 P2 P2 P2 P3 P3 P4 P4 P4 P4 P5 P5 P5
PP  : P1 P1 P1 P3 P3 P1 P1 P2 P2 P4 P4 P4 P4 P5 P5 P5 P2 P2 P2 P2
SRTF: P1 P1 P1 P1 P1 P3 P3 P2 P2 P2 P2 P2 P2 P5 P5 P5 P4 P4 P4 P4
-----
Process exited after 10.08 seconds with return value 0
Press any key to continue . . .
```

Θα συνεχίσουμε με κάποια ακόμα παραδείγματα που επιδεικνύουν τα πλεονεκτήματα και τα μειονεκτήματα του κάθε αλγορίθμου χρονοπρογραμματισμού και τους συγκρίνει.

#### Για τον FCFS:

#### Παράδειγμα 1:

Process	Arrival Time	Burst Time
P1	0	8
P2	1	4
P3	2	9
P4	3	5

```
Select a method.
1) First come first served.
2) Priority based Preemptive.
3) Shortest Remaining Time First.
4) All of the above.
0) Close program.
1
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
FCFS: P1 P1 P1 P1 P1 P1 P1 P1 P2 P2 P2 P2 P3 P3 P3 P3 P3 P3 P3 P3 P4 P4 P4 P4 P4
```

Σε αυτό το παράδειγμα οι διεργασίες φτάνουν με τη σειρά P1, P2, P3, P4 και εκτελούνται με την ίδια σειρά. Ο χρόνος ολοκλήρωσης για κάθε διαδικασία είναι ο εξής:

- P1: 8 μονάδες χρόνου
- P2: 12 μονάδες χρόνου
- P3: 21 μονάδες χρόνου
- P4: 26 μονάδες χρόνου

Ο χρόνος αναμονής για κάθε διεργασία είναι:

- P1: 0 μονάδες χρόνου
- P2: 8 μονάδες χρόνου
- P3: 16 μονάδες χρόνου
- P4: 21 μονάδες χρόνου

Σε αυτό το παράδειγμα, το πλεονέκτημα του FCFS είναι ότι είναι εύκολο να κατανοηθεί και να εφαρμοστεί. Ωστόσο, το μειονέκτημα είναι ότι οι P2, P3 και P4 πρέπει να περιμένουν να τελειώσει το P1, παρόλο που το P2 θα μπορούσε να είχε τελειώσει νωρίτερα. Ως αποτέλεσμα, ο μέσος χρόνος αναμονής των διεργασιών είναι  $(0+8+16+21)/4 = 11,25$  μονάδες χρόνου, που είναι αρκετά υψηλός.

### Παράδειγμα 2:

Process	Arrival Time	Burst Time
P1	0	2
P2	1	4
P3	2	6
P4	3	8
P5	4	10

```
Select a method.
1) First come first served.
2) Priority based Preemptive.
3) Shortest Remaining Time First.
4) All of the above.
0) Close program.
1
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
FCFS: P1 P1 P2 P2 P2 P2 P3 P3 P3 P3 P3 P3 P4 P4 P4 P4 P4 P4 P4 P5 P5 P5 P5 P5 P5 P5 P5 P5
```

Σε αυτό το παράδειγμα, οι διεργασίες φτάνουν με τη σειρά P1, P2, P3, P4, P5 και εκτελούνται με την ίδια σειρά. Ο χρόνος ολοκλήρωσης για κάθε διαδικασία είναι ο εξής:

- P1: 2 μονάδες χρόνου
- P2: 6 μονάδες χρόνου
- P3: 12 μονάδες χρόνου
- P4: 20 μονάδες χρόνου
- P5: 30 μονάδες χρόνου

Ο χρόνος αναμονής για κάθε διαδικασία είναι:

- P1: 0 μονάδες χρόνου
- P2: 2 μονάδες χρόνου
- P3: 6 μονάδες χρόνου
- P4: 12 μονάδες χρόνου
- P5: 20 μονάδες χρόνου

Σε αυτό το παράδειγμα, δεν υπάρχει σημαντική διαφορά μεταξύ των burst time των διεργασιών, επομένως ο αλγόριθμος FCFS μπορεί να είναι μια καλή επιλογή. Το burst time όλων των διεργασιών αυξάνεται και ο χρόνος αναμονής αυξάνεται επίσης με τον ίδιο λόγο, πράγμα που σημαίνει ότι το σύστημα λειτουργεί ομαλά. Σε αυτό το σενάριο, το πλεονέκτημα του FCFS είναι ότι είναι απλό στην εφαρμογή του και δεν προκαλεί επιπλέον επιβάρυνση για τη διαχείριση των διαδικασιών.

**Σύγκριση:** Ωστόσο, γενικά, το FCFS μπορεί να μην είναι η καλύτερη επιλογή όταν οι διεργασίες έχουν διαφορετικά burst time και το σύστημα αναμένεται να έχει υψηλή απόδοση. Σε αυτήν την περίπτωση, άλλοι αλγόριθμοι όπως SRTF ή Round Robin μπορεί να είναι πιο κατάλληλοι.

Για την SRTF:

### Παράδειγμα 1:

Process	Arrival Time	Burst Time
P1	0	10
P2	1	5
P3	2	15
P4	3	20

[illegible]

Σε αυτό το παράδειγμα, οι διεργασίες φτάνουν με τη σειρά P1, P2, P3, P4. Ωστόσο, εκτελούνται με τη σειρά P2, P1, P3, P4. Ο χρόνος ολοκλήρωσης για κάθε διαδικασία είναι ο εξής:

- P1: 12 μονάδες χρόνου
- P2: 5 μονάδες χρόνου
- P3: 20 μονάδες χρόνου
- P4: 40 μονάδες χρόνου

Ο χρόνος αναμονής για κάθε διαδικασία είναι:

- P1: 2 μονάδες χρόνου
- P2: 0 μονάδες χρόνου
- P3: 7 μονάδες χρόνου
- P4: 20 μονάδες χρόνου

Σε αυτό το παράδειγμα, το SRTF βελτιώνει την απόδοση επιλέγοντας τη διαδικασία με τον συντομότερο χρόνο εκτέλεσης. Τα P1 και P3 έχουν εκτελεστεί μετά το P2, γεγονός που βελτιώνει τη συνολική απόδοση του συστήματος. Ο μέσος χρόνος αναμονής των διαδικασιών είναι  $(0+2+7+20)/4 = 6,75$  μονάδες χρόνου που είναι πολύ καλύτερος από το FCFS.

**Παράδειγμα 2:**

Process	Arrival Time	Burst Time
P1	0	10
P2	1	10
P3	2	10
P4	3	10
P5	4	10

```

Select a method.
1) First come first served.
2) Priority based Preemptive.
3) Shortest Remaining Time First.
4) All of the above.
5) Close program.
3
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 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
SRTF: P1 P1 P1 P1 P1 P1 P1 P1 P1 P2 P2 P2 P2 P2 P2 P2 P2 P3 P3 P3 P3 P3 P3 P3 P3 P3 P3 P4 P4 P4 P4 P4 P4 P4 P5 P5 P5 P5 P5 P5 P5 P5 P5

```

Σε αυτό το σενάριο, οι διεργασίες φτάνουν με τη σειρά P1, P2, P3, P4, P5. Το υπολειπόμενο burst time για κάθε διεργασία είναι επίσης 10 μονάδες χρόνου.

Στον αλγόριθμο SRTF, ο χρονοπρογραμματισμός επιλέγει τη διαδικασία με τον μικρότερο burst time για εκτέλεση. Σε αυτήν την περίπτωση, όλες οι διεργασίες έχουν το ίδιο burst time, επομένως ο χρονοπρογραμματιστής δεν θα μπορεί να αποφασίσει με βάση αυτό.

Σε αυτό το σενάριο, το μειονέκτημα του SRTF είναι ότι μπορεί να οδηγήσει σε κακή απόδοση για ορισμένα σύνολα διεργασιών, όπως εκείνα με πολλές διεργασίες με ίδια burst time. Είναι επίσης ένας πιο σύνθετος αλγόριθμος για εφαρμογή και διαχείριση από τον FCFS και μπορεί να προκαλέσει πρόσθετη επιβάρυνση και διαχείριση των διεργασιών.

**Σύγκριση:** Στο συγκεκριμένο παράδειγμα θα ήταν καλύτερο να χρησιμοποιηθεί ο FCFS ή ο Round Robin.







**Starvation:** Το Starvation είναι μια κατάσταση κατά την οποία μια διαδικασία παραμένει σε αναμονή για μεγάλο χρονικό διάστημα πριν δοθεί η ευκαιρία να εκτελεστεί. Αυτό μπορεί να συμβεί σε αλγόριθμους προγραμματισμού που χρησιμοποιούν προτεραιότητες, όπως το Priority Preemptive, όπου μια διεργασία με χαμηλότερη προτεραιότητα δεν έχει την ευκαιρία να εκτελεστεί εγκαίρως.

Ένα ακόμα παράδειγμα του Starvation είναι:

Process	Arrival Time	Burst Time	Priority
P1	0	15	2
P2	1	10	3
P3	2	20	1
P4	3	5	4
P5	4	25	5

```
Select a method.
1) First come first served.
2) Priority based Preemptive.
3) Shortest Remaining Time First.
4) All of the above.
0) Close program.
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 31 32 33 34 35 36 37 38 39 40
PP : P1 P1 P3 P3 P3 P3 P3 P3 P3 P3 P3 P3 P3 P3 P3 P3 P3 P3 P3 P3 P3 P1 P1 P1 P1 P1 P1 P1 P1 P1 P1 P1 P1 P1 P1 P1 P1
```

[illegible]

Σε αυτό το παράδειγμα, οι διεργασίες φτάνουν με τη σειρά P1, P2, P3, P4, P5. Η προτεραιότητα κάθε διαδικασίας δίνεται επίσης ως αριθμός, με το 1 να είναι η υψηλότερη προτεραιότητα και το 5 να είναι η χαμηλότερη. Το P5 έχει τη χαμηλότερη προτεραιότητα, επομένως θα είναι η τελευταία διεργασία που θα εκτελεστεί. Πρέπει να περιμένει να ολοκληρωθούν όλες οι άλλες διεργασίες, πράγμα που σημαίνει ότι θα βρίσκεται σε starvation για πολύ μεγάλο χρονικό διάστημα, γεγονός που μπορεί να οδηγήσει σε κακή απόδοση. Το starvation είναι ένα σοβαρό ζήτημα στον προγραμματισμό αλγορίθμων που χρησιμοποιούν προτεραιότητα. Μπορεί να προκαλέσει την αναμονή μιας διεργασίας για αόριστο χρονικό διάστημα, οδηγώντας σε κακή απόδοση του συστήματος και μπορεί επίσης να προκαλέσει τη μη ολοκλήρωση της εκτέλεσής της.

