

# Ιδεατή Μνήμη

## Άσκηση 1:

Ένας υπολογιστής έχει τέσσερα πλαίσια σελίδων. Οι χρονικές στιγμές φόρτωσης, τελευταίας προσπέλασης, ο αριθμός αναφορών και τα bit αναφοράς (R) και μεταβολής (M ή “dirty”) φαίνονται στον πιο κάτω πίνακα (οι χρονικές στιγμές αναφέρονται σε «τικ» εσωτερικού ρολογιού).

Σελίδα	Φορτώθηκε	Προσπελάστηκε	Συχνότητα	R	M (dirty)
0	126	280	10	1	0
1	230	265	6	0	1
2	140	270	5	0	0
3	110	285	9	1	1

Ποιά σελίδα θα διαλέξουν οι αλγόριθμοι (α) MFU (Most Frequently Used), (β) FIFO (First-In-First-Out), (γ) LRU (Least Recently Used) και (δ) Δεύτερης Ευκαιρίας όταν γίνει το επόμενο σφάλμα σελίδας;

ΣΗΜ: Ο MFU αλγόριθμος εξετάζει την πληροφορία που δίνει η στήλη Συχνότητα για να βρει τη σελίδα με τη μεγαλύτερη συχνότητα χρήσης. Ο FIFO αλγόριθμος εξετάζει το χρόνο φόρτωσης στη στήλη Φορτώθηκε και βγάζει την αρχαιότερη σελίδα στη μνήμη.

## Λύση:

(α) MFU: Πιο συχνά χρησιμοποιούμενη είναι η 0 με 10 αναφορές.

(β) FIFO: Φευγει πρώτος όποιος ήρθε πρώτος. Πρώτη σελίδα ήρθε η 3. Μικρότερος χρόνος φόρτωσης (110) σημαίνει αρχαιότερη σελίδα στη μνήμη.

(γ) LRU: Η σελίδα 1 προσπελάστηκε πιο πριν από οποιαδήποτε άλλη (μικρότερος χρόνος προσπέλασης είναι ο 265) άρα είναι η λιγότερο πρόσφατα χρησιμοποιημένη.

(δ) Δεύτερη Ευκαιρία. Η σελίδα 3 που θα διάλεγε ο αλγόριθμος FIFO (χρόνος φόρτωσης 110) παίρνει μια δεύτερη ευκαιρία, γιατί έχει το bit R ίσο με 1. Η επόμενη στη σειρά σελίδα (χρόνος φόρτωσης 126) είναι η 0 που έχει και αυτή αναφερθεί πρόσφατα (R = 1). Καταλήγουμε λοιπόν στη σελίδα 2 (χρόνος φόρτωσης 140), που δεν έχει αναφερθεί.

## Άσκηση 2:

Δίνεται η ακολουθία αναφορών: 1 2 3 4 5 6 7 8 7 8 7 8 2 1 5 2 4 5 3 2 5 2. Πόσα σφάλματα σελίδας θα δημιουργήσουν οι αλγόριθμοι (α)LRU, (β)FIFO και (γ) Optimal, αν τα διαθέσιμα πλαίσια είναι ένα, τέσσερα, έξη και επτά; Υποθέτουμε ότι αρχικά τα πλαίσια είναι κενά.



0	1	1	3	3	5	6	7	8	7	8	7	8	2	1	5	2	4	5	3	2	5	2
1	1	2	3	4	5	6	7	8	7	8	7	8	2	1	5	2	4	5	3	2	5	2
Σφάλμα	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

**Αλγόριθμος:FIFO, Αριθμός πλαισίων: 4 Αριθμός Σφαλμάτων: 14**

Αναφορά-Πλαίσιο	1	2	3	4	5	6	7	8	7	8	7	8	2	1	5	2	4	5	3	2	5	2
0	1	1	1	1	5	5	5	5	5	5	5	5	2	2	2	2	2	2	3	3	3	3
1	-	2	2	2	2	6	6	6	6	6	6	6	6	1	1	1	1	1	1	2	2	2
2	-	-	3	3	3	3	7	7	7	7	7	7	7	7	5	5	5	5	5	5	5	5
3	-	-	-	4	4	4	4	8	8	8	8	8	8	8	8	8	4	4	4	4	4	4
Σφάλμα	X	X	X	X	X	X	X	X	-	-	-	-	X	X	X	-	X	-	X	X	-	-

**Αλγόριθμος:FIFO, Αριθμός πλαισίων: 6 Αριθμός Σφαλμάτων: 13**

Αναφορά-Πλαίσιο	1	2	3	4	5	6	7	8	7	8	7	8	2	1	5	2	4	5	3	2	5	2
0	1	1	1	1	1	1	7	7	7	7	7	7	7	7	7	7	7	7	3	3	3	3
1	-	2	2	2	2	2	2	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
2	-	-	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2
3	-	-	-	4	4	4	4	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1
4	-	-	-	-	5	5	5	5	5	5	5	5	5	5	5	5	5	4	4	4	4	4
5	-	-	-	-	-	6	6	6	6	6	6	6	6	6	6	6	6	6	5	5	5	5
Σφάλμα	X	X	X	X	X	X	X	X	-	-	-	-	X	X	-	-	X	X	X	-	-	-

**Αλγόριθμος:FIFO, Αριθμός πλαισίων: 7 Αριθμός Σφαλμάτων: 11**

Αναφορά-Πλαίσιο	1	2	3	4	5	6	7	8	7	8	7	8	2	1	5	2	4	5	3	2	5	2
0	1	1	1	1	1	1	1	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
1	-	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1
2	-	-	3	3	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2
3	-	-	-	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	3	3	3
4	-	-	-	-	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
5	-	-	-	-	-	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
6	-	-	-	-	-	-	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
Σφάλμα	X	X	X	X	X	X	X	X	-	-	-	-	-	X	-	X	-	-	X	-	-	-

**Αλγόριθμος:Optimal, Αριθμός πλαισίων: 1 Αριθμός Σφαλμάτων: 22**

Αναφορά-Πλαίσιο	1	2	3	4	5	6	7	8	7	8	7	8	2	1	5	2	4	5	3	2	5	2
0	1	2	3	4	5	6	7	8	7	8	7	8	2	1	5	2	4	5	3	2	5	2
Σφάλμα	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

**Αλγόριθμος: Optimal, Αριθμός πλαισίων: 4 Αριθμός Σφαλμάτων: 11**

Αναφορά-Πλαίσιο	1	2	3	4	5	6	7	8	7	8	7	8	2	1	5	2	4	5	3	2	5	2
0	1	1	1	1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
1	-	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	-	-	3	3	3	6	7	7	7	7	7	7	7	1	1	1	1	1	3	3	3	3
3	-	-	-	4	4	4	4	8	8	8	8	8	8	8	8	8	8	4	4	4	4	4

<b>Σφάλμα</b>	X	X	X	X	X	X	X	X	-	-	-	-	-	X	-	-	X	-	X	-	-	-
---------------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

**Αλγόριθμος: Optimal, Αριθμός πλαισίων: 6 Αριθμός Σφαλμάτων: 9**

<b>Αναφορά-Πλαίσιο</b>	1	2	3	4	5	6	7	8	7	8	7	8	2	1	5	2	4	5	3	2	5	2
<b>0</b>	1	1	1	1	1	1	1	8	8	8	8	8	8	1	1	1	1	1	1	1	1	1
<b>1</b>	-	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
<b>2</b>	-	-	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
<b>3</b>	-	-	-	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
<b>4</b>	-	-	-	-	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
<b>5</b>	-	-	-	-	-	6	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
<b>Σφάλμα</b>	X	X	X	X	X	X	X	X	-	-	-	-	-	X	-	-	-	-	-	-	-	-

**Αλγόριθμος: Optimal, Αριθμός πλαισίων: 7 Αριθμός Σφαλμάτων: 8**

<b>Αναφορά-Πλαίσιο</b>	1	2	3	4	5	6	7	8	7	8	7	8	2	1	5	2	4	5	3	2	5	2
<b>0</b>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
<b>1</b>	-	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
<b>2</b>	-	-	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
<b>3</b>	-	-	-	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
<b>4</b>	-	-	-	-	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
<b>5</b>	-	-	-	-	-	6	6	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
<b>6</b>	-	-	-	-	-	-	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
<b>Σφάλμα</b>	X	X	X	X	X	X	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Συγκεντρωτικά αποτελέσματα:

Ο παρακάτω πίνακας δίνει τον αριθμό σφαλμάτων ανά αλγόριθμο, με παράμετρο τα διαθέσιμα πλαίσια.

Διαθέσιμα πλαίσια	Αλγόριθμοι		
	LRU	FIFO	Optimal
1	22	22	22
4	13	14	11
6	12	13	9
7	10	11	8

### Άσκηση 3:

Παρακάτω δίνεται η ακολουθία εντολών ενός προγράμματος ενός υπολογιστή με σελίδες των 512 bytes. Το πρόγραμμα έχει φορτωθεί στη διεύθυνση 1020 και ο δείκτης στίβας στη διεύθυνση 8192 (η στίβα μεγαλώνει προς τη διεύθυνση 0). Δώστε τη σειρά αναφορών στη μνήμη που δημιουργείται από το πρόγραμμα. Κάθε εντολή καταλαμβάνει 4 bytes (1 λέξη) και εμπεριέχει τις απόλυτες σταθερές. Και οι αναφορές σε εντολές και οι αναφορές σε δεδομένα μετρώνε στη σειρά αναφορών.

Φόρτωσε τη λέξη 6144 στον καταχωρητή 0

Σπρώξε τον καταχωρητή 0 στη στίβα

Κάλεσε την υπορουτίνα από το 5120 και στίβαξε τη διεύθυνση επιστροφής

Αφαίρεσε την απόλυτη σταθερά 6 από το δείκτη στίβας  
 Σύγκρισε την πραγματική παράμετρο με τη απόλυτη σταθερά 4  
 Πήδα σε περίπτωση ισότητας στη διεύθυνση 5152

### **Λύση:**

Στον πιο κάτω πίνακα παρουσιάζονται οι καταχωρητές που χρησιμοποιούνται για την προσπέλαση μνήμης: Ο μετρητής προγράμματος δείχνει την διεύθυνση της επόμενης προς εκτέλεση εντολής. Ο δείκτης στοίβας, αυξάνεται αφού χρησιμοποιηθεί για την αποθήκευση στη στοίβα και ο δείκτης δεδομένων υπαγορεύεται από τη εκτελούμενη εντολή. Οι διευθύνσεις με (\*) δημιουργούν προσπελάσεις στη μνήμη.

Εντολή	Μετρητής προγράμματος	Δείκτης στοίβας	Δείκτης δεδομένων	Σχόλιο
0	1020 *	8192		Κατάσταση προ εκτελέσεως
1	1024 *	8192	6144 *	Διάβασμα από τη μνήμη
2	1028 *	8191	8191 *	Ο δείκτης μειώνεται προ εκτελέσεως
3	5120 *	8190	8190 *	Κλήση υπορουτίνας
4	5124 *	8184		Μείωση του δείκτη
5	5128 *	8184	8184 *	Σύγκριση με δεδομένο από τη μνήμη
6	5152 *	8184		Υπόθετικό αποτέλεσμα σύγκρισης θετικό

Ο πιο κάτω πίνακας συνοψίζει τις αναφορές και τις μετατρέπει σε αριθμούς σελίδας (πηλίκιο διαίρεσης με το 512, αρχίζοντας από 0)

<b>Αναφορά</b>	1020	1024	6144	1028	8191	5120	8190	5124	5128	8184	5152
<b>Σελίδα</b>	2	2	12	2	15	10	15	10	10	15	10

# Αλγόριθμοι Χρονοδρομολόγησης

## Άσκηση 4:

Έστω 3 διαδικασίες A, B, Γ οι οποίες καταφτάνουν στο σύστημα κατά τις χρονικές στιγμές 0, 2 και 4 αντίστοιχα, με σκοπό να εκτελέσουν υπολογισμούς στη CPU με απαιτήσεις χρόνου  $T_A=7\text{sec}$ ,  $T_B=4\text{sec}$  και  $T_\Gamma=1\text{sec}$ .

1. Δείξτε πώς θα εκτελεστούν οι ανωτέρω διεργασίες στη CPU για κάθε έναν από τους ακόλουθους αλγόριθμους χρονοδρομολόγησης:

(α) Round Robin με κβάντο χρόνου 2 sec

(β) Shortest Job First – preemptive (δηλαδή επιτρέποντας ‘διακοπή/επανεισαγωγή’)

(γ) Shortest Job First – non-preemptive (δηλαδή μη επιτρέποντας ‘διακοπή/επανεισαγωγή’)

2. Υπολογίστε το ‘μέσο χρόνο διεκπεραίωσης’ και το ‘μέσο χρόνο αναμονής’ των τριών διαδικασιών μετά την εκτέλεσή τους με κάθε έναν από τους ανωτέρω αλγόριθμους χρονοδρομολόγησης.

## Λύση:

1(α) Εκτέλεση των διεργασιών με Round-Robin (κβάντο χρόνου 2 sec)

	0	2	4	6	8	9	11	12
<b>RR – κβάντο 2 sec</b>		A	A	B	A	Γ	B	A

Τη χρονική στιγμή 0 στην ουρά βρίσκεται μόνο η A οπότε και εισέρχεται για εκτέλεση. Τη χρονική στιγμή 2 που τελειώνει το κβάντο χρόνου της επιστρέφει στην ουρά. Την ίδια χρονική στιγμή όμως έρχεται και η B και μπαίνει και αυτή στην ουρά – πίσω από την A που μόλις επέστρεψε σε αυτήν (ready queue: A,B). Έτσι για το επόμενο κβάντο χρόνου εκτελείται πάλι η A και στην ουρά μένει η B (ready queue: B). Τη χρονική στιγμή 4 που τελειώνει το επόμενο κβάντο χρόνου της A επιστρέφει ξανά στην ουρά (ready queue: B,A). Την ίδια χρονική στιγμή όμως έρχεται και η Γ και μπαίνει και αυτή στην ουρά – πίσω από την A που μόλις επέστρεψε σε αυτήν (ready queue: B,A,Γ). Έτσι στη συνέχεια (στο επόμενο κβάντο χρόνου) εκτελείται η B, μετά η A, μετά η Γ κ.ο.κ. εκ περιτροπής με τη σειρά που βρίσκονται στην ουρά (για κβάντο χρόνου 2 κάθε φορά η κάθε μία και μέχρι να ολοκληρωθεί η εκτέλεσή τους).

1(β) Εκτέλεση των διεργασιών με SJF (preemptive)

	0	2	4	5	7	12
<b>SJF - preemptive</b>		A	B	Γ	B	A

Τη χρονική στιγμή 0 στην ουρά βρίσκεται μόνο η A οπότε και εισέρχεται για εκτέλεση. Τη χρονική στιγμή 2 έρχεται η B και επειδή η διάρκειά της (4sec) είναι μικρότερη από τον εναπομείναντα χρόνο της A ( $7-2=5\text{sec}$ ), εισέρχεται αυτή για

εκτέλεση και η A επιστρέφει στην ουρά. Τη χρονική στιγμή 4 έρχεται η Γ και επειδή η διάρκειά της (1sec) είναι μικρότερη από τον εναπομείναντα χρόνο της B (4-2=2sec), εισέρχεται αυτή για εκτέλεση και η B επιστρέφει στην ουρά (πριν την A γιατί της απομένει λιγότερος χρόνος από αυτόν που απομένει στην A). Τη χρονική στιγμή 5 που τελειώνει και βγαίνει η Γ, εισέρχονται στη συνέχεια οι B και η A (με τη σειρά αυτή που βρίσκονται στην ουρά – γιατί στη B απομένει λιγότερος χρόνος απότι στην A – βλ. παραπάνω) και εκτελούνται μέχρι την ολοκλήρωσή τους.

1(γ) Εκτέλεση των διεργασιών με SJF (non-preemptive)

0	7	8	12
<b>SJF – non-preemptive</b>	A	Γ	B

Τη χρονική στιγμή 0 στην ουρά βρίσκεται μόνο η A οπότε και εισέρχεται για εκτέλεση μέχρι την ολοκλήρωσή της. Τη χρονική στιγμή 7 που τελειώνει και βγαίνει η A, έχουν αφιχθεί στην ουρά τόσο η B όσο και η Γ οπότε επιλέγεται πρώτα η μικρότερη σε διάρκεια (Γ) και μετά η B, και εκτελούνται με αυτή τη σειρά (Γ, B) μέχρι την ολοκλήρωσή τους.

2. Χρόνος Διεκπεραίωσης = (στιγμή ολοκλήρωσης – στιγμή άφιξης)

Χρόνος Αναμονής = (χρόνος διεκπεραίωσης – χρόνος εκτέλεσης στη CPU)

	Round-Robin		SJF - preemptive		SJF – non-preemptive	
	Χρ. Διεκπερ.	Χρ. Αναμονής	Χρ. Διεκπερ.	Χρ. Αναμονής	Χρ. Διεκπερ.	Χρ. Αναμονής
<b>A</b>	12-0 = 12	12-7 = 5	12-0 = 12	12-7 = 5	7-0 = 7	7-7 = 0
<b>B</b>	11-2 = 9	9-4 = 5	7-2 = 5	5-4 = 1	12-2 = 10	10-4 = 6
<b>Γ</b>	9-4 = 5	5-1 = 4	5-4 = 1	1-1 = 0	8-4 = 4	4-1 = 3
<b>M.X</b>	26/3 = 8,66	14/3 = 4,66	18/3 = 6	6/3 = 2	21/3 = 7	9/3 = 3