

ΣΥΓΧΡΟΝΑ ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

3<sup>η</sup> έκδοση

ANDREW S. TANENBAUM

Ασκήσεις - Ερωτήσεις  
3<sup>ου</sup> Κεφαλαίου

## Άσκηση 3.1 (Πρ. 3)

Μνήμη μεγέθους **128 MB** εκχωρείται σε μονάδες των  $n$  **byte**. Πόσος αποθηκευτικός χώρος (σε byte) χρειάζεται για την παρακολούθηση της ελεύθερης μνήμης *(α) με χάρτη bit (bitmap)*, και *(β) με συνδεδεμένη λίστα*. Ποιά μέθοδος είναι καλύτερη;

Για τη συνδεδεμένη λίστα, υποθέστε ότι:

1. Η μνήμη αποτελείται από μια εναλλασσόμενη ακολουθία τμημάτων Δεδομένων και Κενών (οπών) σε μέγεθος **64 KB**.
2. Κάθε κόμβος στη συνδεδεμένη λίστα χρειάζεται μια διεύθυνση μνήμης των **32 bit**, ένα πεδίο **16 bit** για το μήκος, και ένα πεδίο **16 bit** για τον επόμενο κόμβο.

# Απάντηση 3.1 (Πρ. 3)

1 KB =  $2^{10}$  byte και 1 MB =  $2^{20}$  byte.

Άρα η μνήμη 128 MB =  $2^7 * 2^{20}$  byte =  $2^{27}$  byte.

**(α)** Ο χάρτης bit (bitmap) χρειάζεται 1 bit για κάθε μονάδα κατανομής.

Άρα για  $2^{27}$  byte μνήμης, ο χάρτης χρειάζεται χώρο  **$2^{27}/n$  bit.**

**(β)** Η συνδεδεμένη λίστα χρειάζεται  $(32+16+16) = 64$  bit/κόμβο = 8 byte =  $2^3$  byte.

Εφόσον κάθε τμήμα έχει μήκος 64 KB (=  $2^6 * 2^{10}$  byte =  $2^{16}$  byte), συνολικά θα υπάρχουν  $2^{27}/2^{16} = 2^{11}$  κόμβοι στη λίστα.

Άρα συνολικό χώρος  **$2^{11}$  κόμβοι \*  $2^3$  byte/κόμβο =  $2^{14}$  byte.**

Συνεπώς, εάν κάθε μονάδα έχει μήκος  $n > 2^{13}$  byte (1KB), ο χάρτης bit είναι καλύτερος.

## Άσκηση 3.2 (Πρ.4 )

Θεωρήστε ένα σύστημα εναλλαγής, στο οποίο η μνήμη περιέχει τα παρακάτω μεγέθη κενών (οπών) κατά σειρά:

10 KB, 4 KB, 20 KB, 18 KB, 7 KB, 9 KB, 12 KB, και 15 KB.

Ποιά οπή θα χρησιμοποιηθεί αν γίνουν συνεχόμενες αιτήσεις τμημάτων με μέγεθος

(α) 12 KB

(β) 10 KB

(γ) 9 KB

και χρησιμοποιείται ο αλγόριθμος της πρώτης προσαρμογής; Επαναλάβετε την άσκηση για τους αλγόριθμους της βέλτιστης προσαρμογής, επόμενης προσαρμογής και χείριστης προσαρμογής.

## Απάντηση 3.2 (Πρ.4 )

- Πρώτη προσαρμογή: 20 KB, 10 KB, 18 KB.
- Βέλτιστη προσαρμογή: 12 KB, 10 KB, 9 KB.
- Επόμενη προσαρμογή: 18 KB, 20 KB, 9 KB.
- Χείριστη προσαρμογή: 20 KB, 18 KB, 15 KB.

## Άσκηση 3.3 (Πρ.9 )

Μια μηχανή έχει χώρο διευθύνσεων 32 bit και σελίδες μεγέθους 8KB. Ο πίνακας σελίδων βρίσκεται εξολοκλήρου στο υλικό και χρησιμοποιεί μία λέξη των 32 bit για κάθε καταχώρηση. Όταν ξεκινάει μια διεργασία, ο πίνακας σελίδων αντιγράφεται από τη μνήμη στο υλικό, με ρυθμό μία λέξη ανά 100 nsec.

Αν κάθε διεργασία εκτελείται για 100 msec (συμπεριλαμβάνεται και ο χρόνος που χρειάζεται για να φορτωθεί ο πίνακας σελίδων), ποιο ποσοστό του χρόνου της CPU αφιερώνεται στη φόρτωση των πινάκων σελίδων;

# Απάντηση 3.3 (Πρ.9 )

- Με χώρο διευθύνσεων 32 bit, έχουμε συνολικά  $2^{32}$  εικονικές διευθύνσεις (virtual address space).
- Κάθε σελίδα έχει χώρο μνήμης  $8\text{KB} = 2^3 * 2^{10} \text{ byte} = 2^{13} \text{ byte} = 2^{13} * 2^3 \text{ bit} = \mathbf{2^{16} \text{ bit}}$
- Πόσες καταχωρήσεις πρέπει να περιλαμβάνει ο πίνακας σελίδων;
  - Όσος είναι ο χώρος διευθύνσεων / μέγεθος σελίδας ή
  - $2^{32} \text{ bit} / 2^{16} \text{ bit} = 2^{16} (= 65.536)$  καταχωρίσεις των 32 bit η κάθε μία.
- Χρόνος φόρτωσης του πίνακα σελίδων:
  - Για 1 σελίδα χρειάζεται χρόνος 100 nsec
  - Για 65.536 σελίδες χρειάζονται χρόνος  $65.536 * 100 \text{ nsec} \approx 6,6 \text{ msec}$ .
- Αν μια διεργασία εκτελείται για 100 msec, χρειάζονται 6,6 msec για τη φόρτωση του πίνακα σελίδων και 93,4 msec για την εκτέλεση. Επομένως **6,6% του χρόνου** αφιερώνεται στη φόρτωση των πινάκων σελίδων.

## Άσκηση 3.4 (Πρ.12 )

Ένας υπολογιστής με διευθύνσεις των 32 bit χρησιμοποιεί έναν πίνακα σελίδων δύο επιπέδων. Οι εικονικές διευθύνσεις χωρίζονται σε ένα πεδίο 9 bit για τον πίνακα σελίδων του υψηλότερου επιπέδου, ένα πεδίο 11 bit για τον πίνακα σελίδων του δεύτερου επιπέδου, και μια σχετική διεύθυνση (offset).

Πόσο μεγάλες είναι οι σελίδες και πόσες υπάρχουν στο χώρο διευθύνσεων;



## Απάντηση 3.4 (Πρ.12 )

- Συνολικά, 20 bit χρησιμοποιούνται για τις εικονικές διευθύνσεις, αφήνοντας 12 για την σχετική διεύθυνση (offset).
- Αυτό σημαίνει ότι κάθε σελίδα έχει μέγεθος  $2^{12}$  byte ή 4KB.
- Ένας χώρος εικονικών διευθύνσεων των 20 bit έχει  $2^{20}$  σελίδες.

## Άσκηση 3.5 (Πρ.14 )

Ένας υπολογιστής έχει εικονικές διευθύνσεις 32 bit και σελίδες μεγέθους 4 KB. Το πρόγραμμα και τα δεδομένα χωρούν μαζί στη χαμηλότερη σελίδα (0-4095). Η στοίβα χωράει στην υψηλότερη σελίδα.

(α) Πόσες καταχωρίσεις χρειάζονται στον πίνακα σελίδων αν χρησιμοποιείται παραδοσιακή (ενός επιπέδου) σελιδοποίηση;

(β) Πόσες καταχωρίσεις στον πίνακα σελίδων χρειάζονται αν χρησιμοποιείται σελιδοποίηση δύο επιπέδων, με 10 bit σε κάθε τμήμα;

# Απάντηση 3.5 (Πρ.14 )

- Κάθε σελίδα έχει μέγεθος 4KB, δηλαδή χρειάζονται 12 bit για το offset ( $4KB = 2^{12}$ )

(α) Εάν έχουμε πίνακα σελίδων ενός επιπέδου:

- Χρειάζονται  $2^{32}/2^{12} = 2^{20}$  σελίδες για να περιγραφεί όλο το εύρος των εικονικών διευθύνσεων (ή 1M σελίδες).
- Επομένως, ο πίνακας σελίδων πρέπει να έχει **1M καταχωρίσεις**.

(β) Για σελιδοποίηση δύο επιπέδων:

- Ο κύριος πίνακας σελίδων έχει  $2^{10}$  ή **1K καταχωρίσεις** με την κάθε μία να δείχνει σε ένα δεύτερο πίνακα σελίδων. Μόνο δύο από αυτές χρησιμοποιούνται.
- Έτσι συνολικά χρειάζονται μόνο 3 καταχωρίσεις του πίνακα σελίδων, μία στον πίνακα ανωτέρου επιπέδου και μία για κάθε πίνακα κατώτερου επιπέδου.

## Άσκηση 3.6 (Πρ.15)

Ένας υπολογιστής με διεργασίες που έχουν 1024 σελίδες στο χώρο διευθύνσεών τους διατηρεί τους πίνακες σελίδων του στη μνήμη. Η επιβάρυνση που απαιτείται για να διαβαστεί μια λέξη από τον πίνακα σελίδων είναι 5 nsec.

Για να μειωθεί αυτή η επιβάρυνση, ο υπολογιστής διαθέτει μία TLB η οποία διατηρεί 32 ζευγάρια (εικονική σελίδα, φυσικό πλαίσιο σελίδας) και μπορεί να κάνει μία αναζήτηση σε 1 nsec.

Τι ποσοστό ευστοχίας απαιτείται για την TLB ώστε να μειωθεί η μέση επιβάρυνση σε 2 nsec;

## Απάντηση 3.6 (Πρ.15 )

Θα πρέπει  $1 \cdot \alpha + 5 \cdot (1 - \alpha) = 2 \Rightarrow \alpha = \frac{3}{4}$  ή 75%.

## Άσκηση 3.7 (Πρ.18 )

Μια μηχανή έχει εικονικές διευθύνσεις 48 bit και φυσικές διευθύνσεις 32 bit. Οι σελίδες έχουν μέγεθος 8 KB.

Πόσες καταχωρίσεις απαιτούνται για τον πίνακα σελίδων;

Πόσα πλαίσια σελίδας υπάρχουν στη RAM;

# Απάντηση 3.7 (Πρ.18 )

- Μέγεθος σελίδας  $8\text{KB} = 2^3 * 2^{10} = 2^{13}$  byte.
- Χώρος εικονικών διευθύνσεων 48 bit, άρα  $2^{48}$  εικονικές διευθύνσεις.
- Χώρος φυσικών διευθύνσεων 32 bit, άρα  $2^{32}$  φυσικές διευθύνσεις.
- Οι καταχωρήσεις του πίνακα σελίδων είναι  $2^{48} / 2^{13} = 2^{35}$  (περίπου 34 δισ.)
- Στην πράξη οι συνολικά  $2^{35}$  εικονικές σελίδες θα εναλλάσσονται στα  $2^{32} / 2^{13} = 2^{19}$  πλαίσια σελίδας στη μνήμη.

## Άσκηση 3.8 (Πρ.22 )

Αν ο αλγόριθμος αντικατάστασης σελίδας FIFO χρησιμοποιηθεί με τέσσερα πλαίσια σελίδας και οκτώ σελίδες, πόσα σφάλματα σελίδας θα προκύψουν αν η συμβολοσειρά αναφορών είναι 0-1-7-2-3-2-7-1-0-3, με δεδομένο ότι τα τέσσερα πλαίσια είναι αρχικά άδεια;

Επαναλάβετε το ίδιο πρόβλημα για τον αλγόριθμο LRU (υπόδειξη: θεωρείστε ότι κάθε καταχώρηση του πίνακα σελίδων, διατηρεί ένα μετρητή  $M$  όπου δείχνει το πλήθος των εντολών κατά τη χρονική στιγμή της τελευταίας αναφοράς στη σελίδα)



# Απάντηση 3.8 (Πρ.22 )

Για τον αλγόριθμο FIFO προκύπτουν 6 σφάλματα:

0-1-7-2-3-2-7-1-0-3

**x0x1x7x2x3271x03**

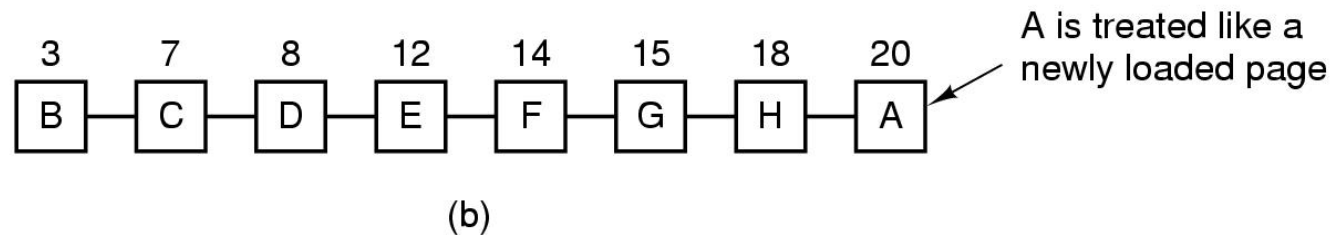
Για τον αλγόριθμο LRU προκύπτουν 7 σφάλματα:

**x0x1x7x2x3271x0x3**

## Άσκηση 3.9 (Πρ.23 )

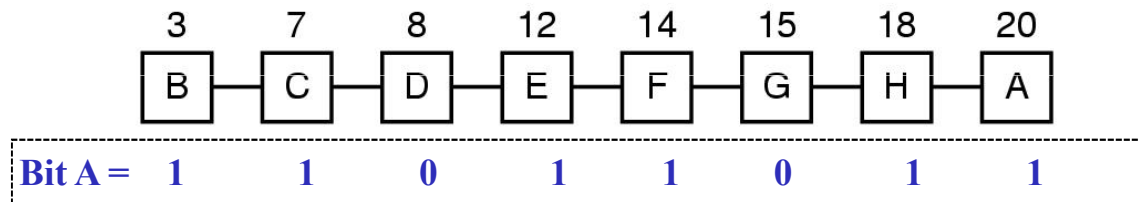
Θεωρήστε την ακολουθία σελίδων της Εικόνας 3-15(β). Υποθέστε ότι τα bit  $A$  για τις σελίδες  $B$  έως  $A$  είναι 11011011, αντίστοιχα.

Ποια σελίδα θα αφαιρέσει ο αλγόριθμος της δεύτερης ευκαιρίας;



# Απάντηση 3.9 (Πρ.23 )

Θα επιλεγεί η 1<sup>η</sup> σελίδα με bit A=0, άρα η D



## Άσκηση 3.10 (Πρ.24 )

Ένας μικρός υπολογιστής έχει τέσσερα πλαίσια σελίδας. Στον πρώτο χτύπο του ρολογιού τα bit  $A$  έχουν τις τιμές 0111 (η σελίδα 0 έχει την τιμή 0, ενώ οι υπόλοιπες 1). Στους επόμενους χτύπους ρολογιού, οι τιμές είναι 1011, 1010, 1101, 0010, 1010, 1100, και 0001.

Αν χρησιμοποιείται οι αλγόριθμοι γήρανσης με ένα μετρητή 8 bit, βρείτε τις τιμές των τεσσάρων μετρητών μετά από τον τελευταίο χτύπο ρολογιού.

# Απάντηση 3.10 (Πρ.24 )

Εφαρμόζουμε τον αλγόριθμο γήρανσης στις σελίδες (πρόσθεση του bit A και μετά ολίσθηση δεξιά. Οι μετρητές θα είναι τελικά:

Σελίδα 0: 01101100

Σελίδα 1: 01001001

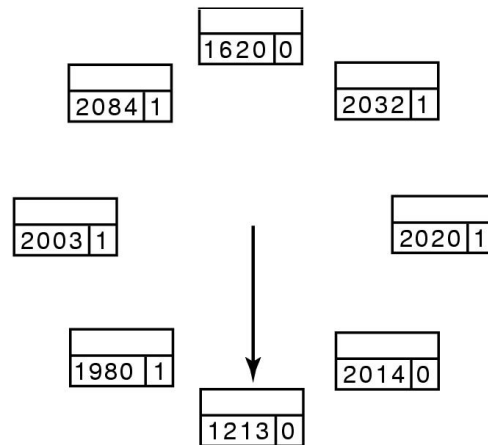
Σελίδα 2: 00110111

Σελίδα 3: 10001011

# Άσκηση 3.11 (Πρ.26 )

Στον αλγόριθμο WSClock της Εικόνας 3-21(γ), ο δείκτης δείχνει σε σελίδα με  $A = 0$ .

Αν  $\tau = 400$ , όπου  $\tau$  είναι η ηλικία του συνόλου εργασίας, θα αφαιρεθεί η συγκεκριμένη σελίδα; Τι θα γίνει αν  $\tau = 1000$ ;



## Απάντηση 3.11 (Πρ.26)

- Η ηλικία της σελίδας είναι  $2204 - 1213 = 991$ . Αν  $\tau = 400$ , δεν ανήκει στο σύνολο εργασίας, επομένως θα εξεταστεί το bit τροποποίησης.
  - Εάν είναι  $T=0$  η σελίδα είναι καθαρή και θα αφαιρεθεί αμέσως. Διαφορετικά θα αναζητηθεί καλύτερη σελίδα.
- Αν  $\tau=1000$ , η κατάσταση είναι διαφορετική. Τώρα η σελίδα ανήκει στο σύνολο εργασίας (οριακά), άρα δεν θα αφαιρεθεί.

## Άσκηση 3.12 (Πρ.28 )

Ένας υπολογιστής έχει τέσσερα πλαίσια σελίδας. Ο χρόνος φόρτωσης, ο χρόνος τελευταίας προσπέλασης, και οι τιμές των bit  $A$  και  $T$  για κάθε σελίδα φαίνονται παρακάτω (οι χρονικές στιγμές είναι σε χτύπους ρολογιού):

Σελίδα	Φορτώθηκε	Τελευταία αναφορά	A	T
0	126	280	1	0
1	230	265	0	1
2	140	270	0	0
3	110	285	1	1

- (α) Ποια σελίδα θα αντικαταστήσει ο αλγόριθμος NRU;
- (β) Ποια σελίδα θα αντικαταστήσει ο αλγόριθμος FIFO;
- (γ) Ποια σελίδα θα αντικαταστήσει ο αλγόριθμος LRU;
- (δ) Ποια σελίδα θα αντικαταστήσει ο αλγόριθμος της δεύτερης ευκαιρίας;



## Απάντηση 3.12 (Πρ.28 )

Ο NRU τη σελίδα 2 (λόγω των bit A, T)

Ο FIFO τη σελίδα 3 (λόγω παλαιότερου χρόνου φόρτωσης)

Ο LRU τη σελίδα 1 (αναφέρθηκε λιγότερο πρόσφατα)

Ο αλγόριθμος της δεύτερης ευκαιρίας τη σελίδα 2 (η 3 είναι παλαιότερη αλλά το bit A της 3 είναι 1 ενώ το bit A της 2 είναι 0)

Σελίδα	Φορτώθηκε	Τελευταία αναφορά	A	T
0	126	280	1	0
1	230	265	0	1
2	140	270	0	0
3	110	285	1	1

## Άσκηση 3.13 (Πρ.31 )

Ένας υπολογιστής παρέχει σε κάθε διεργασία **χώρο διευθύνσεων 65536 byte**, ο οποίος διαιρείται σε σελίδες των **4096 byte**. Ένα συγκεκριμένο πρόγραμμα έχει κώδικα μεγέθους **32768 byte**, δεδομένα μεγέθους **16386 byte**, και στοίβα μεγέθους **15870 byte**.

Θα μπορέσει το πρόγραμμα αυτό να χωρέσει στη μνήμη;

Αν το μέγεθος της σελίδας ήταν **512 byte**, θα χωρούσε;

Θυμηθείτε ότι μια σελίδα δεν μπορεί να περιέχει κομμάτια από δύο διαφορετικά τμήματα.

# Απάντηση 3.13 (Πρ.31 )

- Ο κώδικας χρειάζεται  $32768/4096 = 8$  σελίδες ακριβώς.
- Τα δεδομένα χρειάζονται  $16386/4096=4,00048$  σελίδες, συνεπώς 5 σελίδες.
- Η στοίβα χρειάζεται  $15870/4096 = 3,87$ , συνεπώς 4 σελίδες.
- Το πρόγραμμα έχει χώρο διευθύνσεων  $65536/4096 = 16$  σελίδες.
- Συνολικά χρειάζεται  $8+5+4 = 17$  σελίδες, συνεπώς δεν χωράει.
  
- Με σελίδες των 512-byte, ο κώδικας, τα δεδομένα και η στοίβα χρειάζονται συνολικά 128 σελίδες ( $64+33+31$ ).
- Με συνολικό χώρο διευθύνσεων της διεργασίας  $65536/512 = 128$  σελίδες, χωράει οριακά.

## Άσκηση 3.14 (Πρ.32 )

Μπορεί μια σελίδα να βρίσκεται σε δύο σύνολα εργασίας την ίδια χρονική στιγμή; Εξηγήστε την απάντησή σας.

## Απάντηση 3.14 (Πρ.32 )

- Σε περίπτωση που υπάρχει διαμοιρασμός σελίδων, είναι δυνατό μία σελίδα να βρίσκεται σε περισσότερα σύνολα εργασίας.
- Για παράδειγμα εάν ένα πρόγραμμα χρησιμοποιείται ταυτόχρονα από πολλούς χρήστες σε ένα σύστημα, τότε **οι σελίδες του κώδικα του προγράμματος θα είναι κοινόχρηστες** (εφόσον το bit εγγραφής είναι 0, δεν υπάρχει λόγος να δημιουργήσει το ΛΣ αντίγραφα).
- Συνεπώς κάποιες από αυτές τις κοινόχρηστες σελίδες κώδικα είναι πολύ πιθανό να βρίσκονται ταυτόχρονα στο σύνολο εργασίας και των δύο χρηστών.

# Άσκηση 3.15 (Πρ.33 )

Έχει παρατηρηθεί ότι ο αριθμός εντολών που εκτελούνται ανάμεσα στα σφάλματα σελίδας είναι ευθέως ανάλογος με τον αριθμό των πλαισίων που έχουν εκχωρηθεί σε ένα πρόγραμμα. Αν η διαθέσιμη μνήμη διπλασιαστεί, το μέσο χρονικό διάστημα ανάμεσα στα σφάλματα σελίδας διπλασιάζεται επίσης και ο αριθμός σφαλμάτων υποδιπλασιάζεται.

Υποθέστε ότι μία συνηθισμένη εντολή χρειάζεται **1  $\mu\text{sec}$** , ενώ αν προκύψει σφάλμα σελίδας χρειάζεται **2001  $\mu\text{sec}$**  (δηλαδή επιπλέον 2 msec για το χειρισμό του σφάλματος).

Έστω ότι ένα πρόγραμμα χρειάζεται **60 sec** για να εκτελεστεί, στη διάρκεια των οποίων συμβαίνουν **15.000 σφάλματα** σελίδας. Πόσος χρόνος θα χρειαστεί για την εκτέλεση του προγράμματος, αν διπλασιαστεί η διαθέσιμη μνήμη του;

# Απάντηση 3.15 (Πρ.33 )

- Το πρόγραμμα προκαλεί 15.000 σφάλματα και κάθε σφάλμα απαιτεί 2 msec επιπλέον χρόνο επεξεργασίας
- Συνεπώς η επιβάρυνση που προκαλείται από τα σφάλματα σελίδων είναι  $15.000 * 2 \text{ msec} = 30 \text{ sec}$
- Καθαρός χρόνος εκτέλεσης:  $60 \text{ sec} - 30 \text{ sec} = 30 \text{ sec}$
  
- Εάν το πρόγραμμα πάρει διπλάσια μνήμη, τότε θα προκύψουν  $15.000/2 = 7.500$  σφάλματα
- άρα η επιβάρυνση που προκαλείται από τα σφάλματα σελίδων είναι  $7.500 * 2 \text{ msec} = 15 \text{ sec}$
- Χρόνος εκτέλεσης = Πραγματικός χρόνος εκτέλεσης + επιβάρυνση σφαλμάτων =  $30 \text{ sec} + 15 \text{ sec} = 45 \text{ sec}$