

# ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

## Διαχείριση μνήμης Εργαστηριακές Ασκήσεις

Υλικό από:

*Modern Operating Systems Laboratory Exercises, Shrivakan Mishra*

Σύνθεση

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

1. Εκτελέστε τα animations που σχετίζονται με την ενότητα από την ιστοσελίδα του Εκπαιδευτικού Υλικού στο <http://www.it.uom.gr/teching.html/>

2. Η εντολή free εμφανίζει πληροφορίες για τη μνήμη του συστήματος. Μελετήστε το εγχειρίδιο χρήσης της εντολής εισάγοντας

```
man free
```

Εκτελέστε την εντολή

```
free -o
```

αρκετές φορές, ενώ στο ενδιάμεσο εκκινείτε ή τερματίζετε διάφορα προγράμματα. Αποθηκεύετε τα διάφορα αποτελέσματα της εντολής free. Σχεδιάστε ένα διάγραμμα με την χρονική εξέλιξη των μετρήσεών σας ως εξής.

Άξονας X: MB μνήμης,

Άξονας Y: (i) Memory Used

(ii) Memory Used – Memory Buffered – Memory Cached;

(iii) Swap Used

Εξηγείστε τα αποτελέσματα με βάση την κατάσταση των διαφόρων εφαρμογών που εκτελούνταν στο σύστημα τις διάφορες χρονικές στιγμές..

3. Μεταφορτώστε και εκτελέστε την εφαρμογή Process Explorer από τη θέση

<http://www.microsoft.com/technet/sysinternals/utilitiesindex.mspx>

Επιλέξτε το κουμπί System Information (Ctrl+I) για να δείτε λεπτομέρειες χρήσης της μνήμης. Καταγράψτε πληροφορίες σχετικές με τη χρήση της μνήμης από τον πυρήνα, τις διάφορες εφαρμογές, τη σελιδοποίηση κλπ.

Κατόπιν ξεκινήστε μια νέα εφαρμογή πχ το Microsoft Excel. Περιγράψτε τις αλλαγές στη χρήση της μνήμης κατά την εκκίνηση της εφαρμογής, μετά από ένα λεπτό και κατά τον τερματισμό της εφαρμογής. Με τη βοήθεια του κουμπιού System Information button (Ctrl+I) εξηγήστε το πρότυπο χρήσης της μνήμης.

# Άσκηση 4

- Υποθέστε ότι έχετε ελεύθερη μνήμη σε τμήματα μεγέθους 100KB, 500KB, 200KB, 300KB, and 600KB (με αυτή τη σειρά) και υπάρχουν κατά σειρά απαιτήσεις μνήμης για 212KB, 417KB, 112KB, and 426KB.
- **Δείξτε πως** ο αλγόριθμος πρώτης τοποθέτησης θα διευθετήσει τις απαιτήσεις στην ελεύθερη μνήμη. Υπολογίστε το ελεύθερο τμήμα μνήμης μετά από κάθε άφιξη.
- **Δείξτε πως** ο αλγόριθμος βέλτιστης τοποθέτησης θα διευθετήσει τις ίδιες απαιτήσεις στην ελεύθερη μνήμη. Υπολογίστε το ελεύθερο τμήμα μνήμης μετά από κάθε άφιξη.
- **Δείξτε πως** ο αλγόριθμος επόμενης τοποθέτησης θα διευθετήσει τις απαιτήσεις στην ελεύθερη μνήμη (last allocated block 200KB). Υπολογίστε το ελεύθερο τμήμα μνήμης μετά από κάθε άφιξη.

# Άσκηση 5

- Ένα σύστημα τοποθετεί διεργασίες στη μνήμη χρησιμοποιώντας δυναμική πολιτική τοποθέτησης. Κατά την πλέον πρόσφατη χρονική στιγμή έγινε φόρτωση μιας διεργασίας που χρειαζόταν 12KB μνήμης και η εικόνα μνήμης του συστήματος διαμορφώθηκε ως εξής :



- Οι σκιασμένες περιοχές δηλώνουν αχρησιμοποίητα τμήματα μνήμης, οι λευκές τα κενά τμήματα ενώ η περιοχή με μαύρο χρώμα τη θέση όπου έγινε η τελευταία τοποθέτηση. Οι αριθμοί δηλώνουν το μέγεθος σε KB.
- Να σχεδιάσετε την εικόνα μνήμης μετά την τοποθέτηση μιας νέας διεργασίας που χρειάζεται 22KB μνήμης σύμφωνα με τους αλγορίθμους : first-fit, best-fit, next-fit.

# Άσκηση 6

Ο διαχειριστής μνήμης ενός συστήματος που χρησιμοποιεί στρατηγική τμημάτων μεταβλητού μεγέθους διαθέτει ελεύθερα τμήματα μεγέθους 600, 400, 1000, 2200, 1600, 1050 bytes.

Θεωρείστε κατά σειρά τις παρακάτω διαδοχικές απαιτήσεις:

- ποιο τμήμα θα επιλεγεί για απαίτηση 1603 bytes με τη μέθοδο best-fit;
- ποιο τμήμα θα επιλεγεί για απαίτηση 949 bytes με τη μέθοδο best-fit;
- ποιο τμήμα θα επιλεγεί για απαίτηση 963 bytes με τη μέθοδο first-fit;
- ποιο τμήμα θα επιλεγεί για απαίτηση 349 bytes με τη μέθοδο first-fit?
- Υποθέστε ότι η λίστα ελεύθερων τμημάτων διατάσσεται κατά αύξουσα σειρά μεγέθους των τμημάτων που αναφέρονται στην αρχή της άσκησης. Ποιο τμήμα θα επιλεγεί για απαίτηση 1603 bytes με τη μέθοδο first-fit;

# Άσκηση 7

- Θεωρείστε ένα σύστημα με 1MB διαθέσιμης μνήμης και απαιτήσεις για 42KB, 396KB, 10KB, και 28KB. Δείξτε το ποσό μνήμης που εκχωρείται για κάθε απαίτηση και την κατάσταση της μνήμης μετά από κάθε απαίτηση, **χρησιμοποιώντας το σύστημα ζευγών**.
- Πόσος εσωτερικός κατακερματισμός υπάρχει με το σενάριο αυτό;
- Πόσος εξωτερικός κατακερματισμός υπάρχει με το σενάριο αυτό;

# Άσκηση 8

- Ποιο είναι το μεγαλύτερο μέγεθος προγράμματος που χωρά σε ένα πίνακα σελίδων σε ένα σύστημα που χρησιμοποιεί φυσικές διευθύνσεις 32-bit και μέγεθος σελίδας 1K;
  - ΛΥΣΗ
    - ΦΥΣ. ΔΙΕΥΘΥΝΣΗ 32 bits  $\rightarrow$  χώρος διευθύνσεων  $2^{32}$  διαφορετικές διευθύνσεις
    - Μέγεθος σελίδας 1K =  $2^{10}$  διευθύνσεις μνήμης
    - ΑΡΑ : μέγιστο πλήθος σελίδων =  $2^{32}/2^{10}=2^{22}$
    - Κάθε διεύθυνση των 32 bits (=4 bytes) πρέπει να χωρά σε κάθε μια από τις  $2^{22}$  θέσεις του πίνακα σελίδων άρα μέγεθος του πίνακα σελίδων :  $2^{22} \times 4\text{bytes} = 16\text{Mb}$
    - μεγαλύτερο μέγεθος προγράμματος : **16Mb**



# Άσκηση 9

Δίνεται ο πίνακας σελίδων μιας διεργασίας :

Ποιο είναι το μικρότερο δυνατό μέγεθος σελίδας;

Δώστε σε δυαδική μορφή τη φυσική διεύθυνση της εικονικής διεύθυνσης 1234

Δώστε σε δεκαδική μορφή τη φυσική διεύθυνση της εικονικής διεύθυνσης 3333

| Page No | Frame No |
|---------|----------|
| 0       | 5        |
| 1       | 3        |
| 2       | 4        |
| 3       | 0        |
| 4       | 1        |
| 5       | 2        |

# Λύση 9

Γιατί το μέγεθος σελίδας είναι 1K;

Έστω μέγεθος σελίδας 512 bytes ( $=2^9$  bytes)

Η εικονική διεύθυνση 3333 απαιτεί **6 γεμάτες σελίδες** ( $=512 \times 6 = 3072$ )  
και στην **7η σελίδα μετατόπιση** (offset)  $= 3333 - 3072 = 261$

Άρα απαιτούνται αριθμοί σελίδων **0,1,2,3,4,5,6,7**

Ο πίνακας όμως δεν έχει αριθμό σελίδας μεγαλύτερο του 5. Στην περίπτωση που θεωρηθεί ως μέγεθος σελίδας 512 bytes αυτομάτως η διεύθυνση 3333 είναι εκτός μνήμης, ΑΔΥΝΑΤΟ ΔΙΟΤΙ μας δίνεται ότι ο πίνακας σελίδων της διεργασίας υφίσταται.

ΕΠΟΜΕΝΩΣ Ο ΕΠΟΜΕΝΟΣ ΑΡΙΘΜΟΣ ΠΟΥ ΥΠΑΡΧΕΙ ΩΣ ΔΥΝΑΜΗ ΤΟΥ 2 ΕΊΝΑΙ  $2^{10} = 1024$  bytes=1KB

# Άσκηση 10

- Θεωρείστε ένα λογικό χώρο διευθύνσεων που αποτελείται από οκτώ σελίδες με μέγεθος κάθε σελίδας 1024 bytes, που αντιστοιχείται σε μια φυσική μνήμη που αποτελείται από 32 frames.
- Πόσα bits υπάρχουν στη λογική διεύθυνση?
- Πόσα bits υπάρχουν στη φυσική διεύθυνση?

# Άσκηση 11

- Σε ένα σύστημα που χρησιμοποιεί σελιδοποίηση μια διεργασία δεν μπορεί να έχει πρόσβαση σε μνήμη που δεν της ανήκει. Γιατί;

# Λύση 11

Μια διεύθυνση σε ένα σύστημα με σελιδοποίηση αποτελείται από έναν αριθμό σελίδας και μια μετατόπιση. Η φυσική σελίδα (πλαίσιο) υπολογίζεται μέσω της αναζήτησης στον πίνακα σελίδων της διεργασίας. Επειδή το Λ.Σ. ελέγχει τα περιεχόμενα αυτού του πίνακα μπορεί να περιορίσει μια διεργασία ώστε να έχει πρόσβαση μόνον σε εκείνες τις φυσικές σελίδες (πλαίσια) που της έχουν εκχωρηθεί. Δεν υπάρχει τρόπος για μια διεργασία να αναφερθεί σε σελίδα που δεν της ανήκει επειδή ο αριθμός αυτός δεν βρίσκεται στον πίνακα σελίδων της.

Για να επιτραπεί μια τέτοια πρόσβαση το Λ.Σ. απλά χρειάζεται να επιτρέψει σε τμήματα της μνήμης που δεν ανήκουν σε μια διεργασία να προστεθούν στον πίνακα σελίδων της διεργασίας. Αυτό είναι χρήσιμο όταν δύο ή περισσότερες διεργασίες χρειάζεται να ανταλλάσσουν δεδομένα που μόλις διάβασαν και να τα γράψουν στην ίδια φυσική διεύθυνση (που είναι δυνατόν να αντιστοιχεί σε διαφορετικές λογικές διευθύνσεις). Η διαδικασία αυτή καθιστά πολύ αποτελεσματική την δια-διεργασιακή (Interprocess) επικοινωνία.

# Άσκηση 12

Θεωρείστε ένα σύστημα που χρησιμοποιεί :

απλή σελιδοποίηση και  
τεχνική TLB

Αν μια αναφορά στη μνήμη απαιτεί 400ns, μια αναφορά στο TLB απαιτεί 50ns και το ποσοστό επιτυχίας (hit - rate) στο TLB είναι 80% ποιος είναι ο πραγματικός χρόνος αναφοράς στη μνήμη; Πόση είναι η βελτίωση στην ταχύτητα (speed-up) λόγω χρήσης της τεχνικής TLB;

## Λύση 12

Απλή σελιδοποίηση : Κάθε αναφορά στη μνήμη απαιτεί 2 προσπελάσεις άρα συνολικός χρόνος :  $400\text{ns} + 400\text{ns} = 800\text{ ns}$

Απλή σελιδοποίηση και TLB :

Επιτυχία (hit) στο TLB :  $50\text{ns} + 400\text{ns} = 450\text{ns}$

Αποτυχία (miss) στο TLB :  $50\text{ns} + 400\text{ns} + 400\text{ns} = 850\text{ns}$

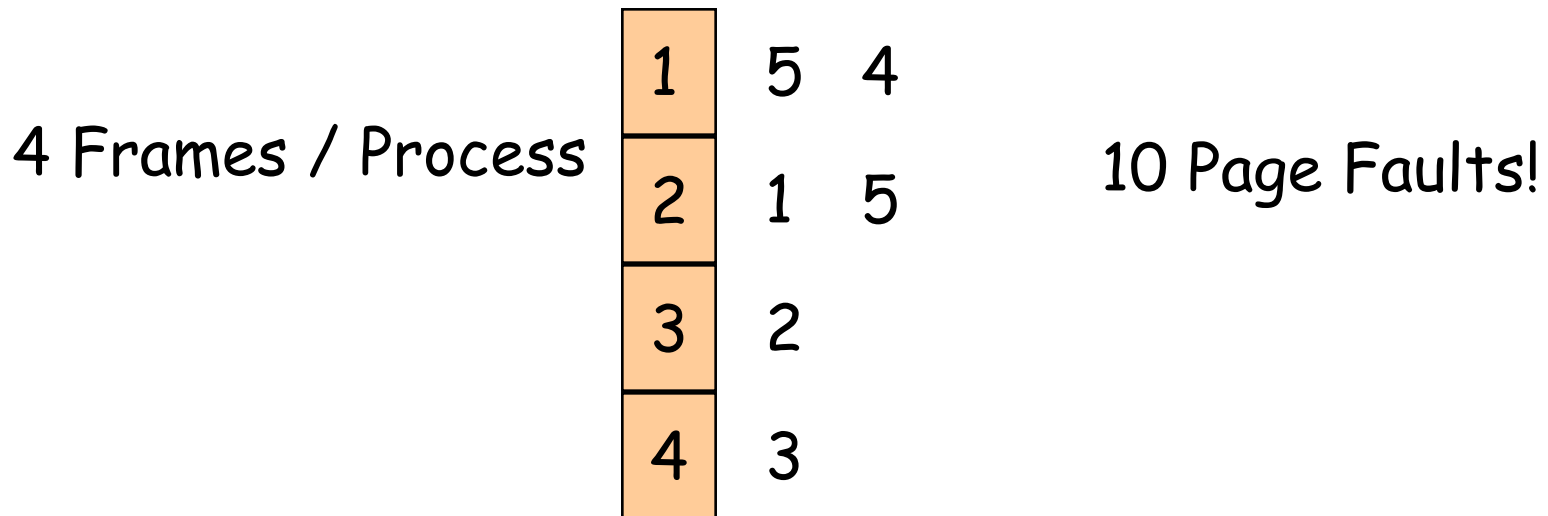
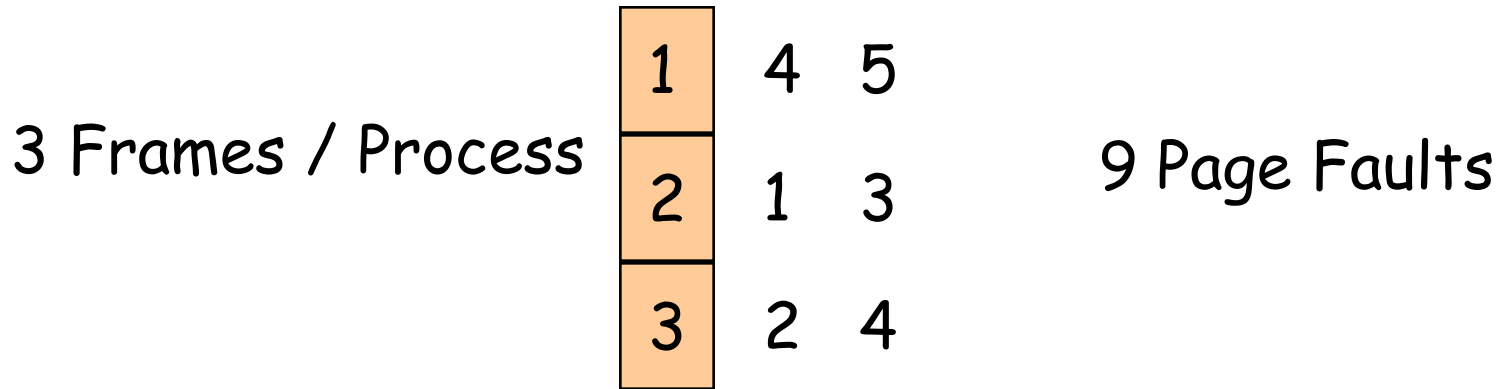
Πραγματικός χρόνος προσπέλασης με απλή σελιδοποίηση και TLB :

$$450 * 80\% + 850 * 20\% = 530\text{ns}$$

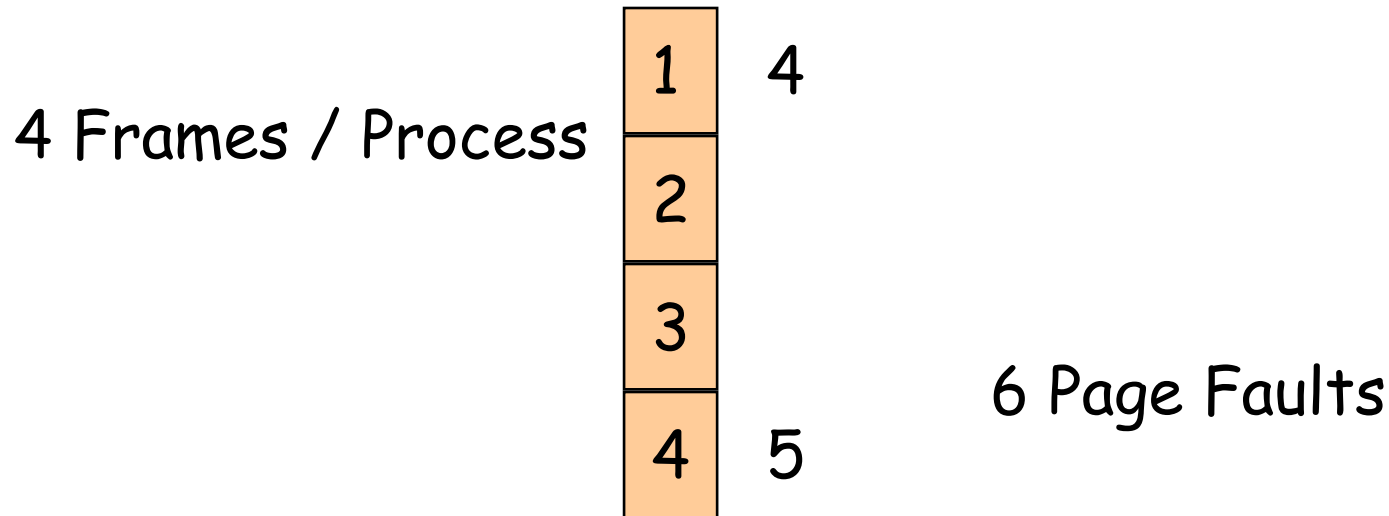
$$\text{Speed-up} = 800 / 530 = 1.51$$

# Άσκηση 13 (FIFO)

Reference string : 1,2,3,4,1,2,5,1,2,3,4,5

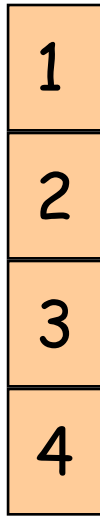


# Άσκηση 13 (Optimal)



Reference string :1,2,3,4,1,2,5,1,2,3,4,5

# Άσκηση 13 (LRU)



5

5

4

3

8 Page Faults

Reference string :1,2,3,4,1,2,5,1,2,3,4,5



# Άσκηση 14 (FIFO)

- Αν η μνήμη αποτελείται από 3 frames και η διεργασία έχει 5 pages, με την ακόλουθη χρονική σειρά αναφοράς :

**/ 2 / 3 / 2 / 1 / 5 / 2 / 4 / 5 / 3 / 2 / 5 / 2 /**

Τότε θα έχουμε τα παρακάτω σφάλματα σελίδας (σημειώνονται με \*) :

**/ 2\* / 3\* / 2 / 1\* / 5\* / 2\* / 4\* / 5 / 3\* / 2 / 5\* / 2\* /**

# Άσκηση 14 (Optimal)

- Αν η μνήμη αποτελείται από 3 frames και η διεργασία έχει 5 pages, με την ακόλουθη χρονική σειρά αναφοράς :

**/ 2 / 3 / 2 / 1 / 5 / 2 / 4 / 5 / 3 / 2 / 5 / 2 /**

- Τότε θα έχουμε τα παρακάτω σφάλματα σελίδας (σημειώνονται με \*) :

**/ 2\* / 3\* / 2 / 1\* / 5\* / 2 / 4\* / 5 / 3 / 2\* / 5 / 2 /**

# Άσκηση 14 (LRU)

- Αν η μνήμη αποτελείται από 3 frames και η διεργασία έχει 5 pages, με την ακόλουθη χρονική σειρά αναφοράς :

**/ 2 / 3 / 2 / 1 / 5 / 2 / 4 / 5 / 3 / 2 / 5 / 2 /**

Τότε θα έχουμε τα παρακάτω σφάλματα σελίδας (σημειώνονται με \*) :

**/ 2\* / 3\* / 2 / 1\* / 5\* / 2 / 4\* / 5 / 3\* / 2\* / 5 / 2 /**

# Άσκηση 15

Reference string: A B C A B D A D B C B

|       | FIFO  | OPT   | LRU   |
|-------|-------|-------|-------|
|       |       |       |       |
| A B C | A B C | A B C | A B C |
| A     | A B C | A B C | A B C |
| B     | A B C | A B C | A B C |
| D     | D B C |       |       |
| A     | D A C |       |       |
| D     | D A C |       |       |
| B     | D A B |       |       |
| C     | C A B |       |       |
| B     | C A B |       |       |

Faults:  
FIFO 7  
OPT 5  
LRU 5

Να συμπληρωθούν τα κενά στους πίνακες.

# Άσκηση 16

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων:

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.

Πόσα σφάλματα σελίδας θα γίνουν για τους παρακάτω αλγόριθμους αντικατάστασης, υποθέτοντας 1,2,3,4,5,6,7 πλαίσια; Όλα τα πλαίσια είναι καταρχήν κενά, έτσι οι πρώτες σελίδες θα στοιχίσουν ένα σφάλμα σελίδας.

1. LRU replacement
2. FIFO replacement
3. Optimal (βέλτιστη)

# Λύση 16

| Πλήθος<br>πλαισίων | LRU | FIFO | ΒΕΛΤΙΣΤΗ |
|--------------------|-----|------|----------|
| 1                  | 20  | 20   | 20       |
| 2                  | 17  | 18   | 15       |
| 3                  | 15  | 16   | 11       |
| 4                  | 10  | 14   | 8        |
| 5                  | 8   | 10   | 7        |
| 6                  | 7   | 9    | 7        |
| 7                  | 7   | 7    | 7        |

# Άσκηση 17

Θεωρείστε την παρακάτω ακολουθία αναφοράς σελίδων :

**A, B, C, A, D, E, C, B, F, E, A, C**

Θεωρείστε επίσης ότι υπάρχουν 4 πλαίσια, καταρχήν κενά.

Πόσα σφάλματα σελίδων θα συμβούν και ποια αν χρησιμοποιηθούν οι παρακάτω αλγόριθμοι αντικατάστασης; Να συμπληρώσετε τους παρακάτω πίνακες

- αλγόριθμος LRU (λιγότερο πρόσφατα χρησιμοποιούμενη σελίδα)
- βέλτιστη πολιτική.

# Άσκηση 18

- Μετρήσεις έχουν δείξει ότι το πλήθος των εντολών (instructions) που εκτελούνται μεταξύ σφαλμάτων σελίδων είναι ανάλογο του πλήθους των πλαισίων σελίδας που εκχωρούνται στο πρόγραμμα που εκτελείται. Αν η διαθέσιμη μνήμη διπλασιαστεί ο μέσος χρόνος μεταξύ των σφαλμάτων σελίδας επίσης διπλασιάζεται. Υποθέστε ότι μια κανονική (συνήθης) εντολή απαιτεί για την εκτέλεσή της χρόνο 1 microsecond (μsec), αλλά αν συμβεί σφάλμα σελίδας απαιτούνται 2001 microseconds (χρειάζονται 2 msec για τον χειρισμό του σφάλματος σελίδας). (Προφανώς 1msec=1000μsec).
- Αν ένα πρόγραμμα χρειάζεται 60 seconds για να ολοκληρωθεί η εκτέλεσή του, και κατά τη διάρκεια της εκτέλεσης συμβαίνουν 15,000 σφάλματα σελίδας να βρείτε:
- Πόσο χρόνο θα χρειαστεί το ίδιο πρόγραμμα για να ολοκληρωθεί η εκτέλεσή του αν του παραχωρηθεί διπλάσιο ποσό μνήμης.
- Σε τι ποσοστό % θα μεταβληθεί (αύξηση ή μείωση) ο χρόνος εκτέλεσης του προγράμματος.



# Άσκηση 19

- Σε μια αρχιτεκτονική χρησιμοποιούνται λογικές διευθύνσεις των 32 bits χωρισμένες ως εξής :

**4-bit segment number** | **12-bit page number** | **16-bit offset**

Ποιο είναι το μέγεθος σελίδας;

Ποιο είναι το μέγιστο μήκος τμήματος;

# Άσκηση 20

Υποθέστε ότι μια διεργασία χωρίζεται σε 4 ίσου μεγέθους τμήματα, και ότι το σύστημα δημιουργεί έναν πίνακα σελίδων 8 θέσεων για κάθε τμήμα (συνδυασμός κατάτμησης και σελιδοποίησης). Υποθέστε επίσης ότι το μέγεθος σελίδας είναι 2 Kb.

Ποιο είναι το μέγιστο μέγεθος κάθε τμήματος?

Ποια είναι ο μέγιστος λογικός χώρος διευθύνσεων για κάθε διεργασία;

Ποια είναι η μορφή μιας λογικής διεύθυνσης στην περίπτωση αυτή;

# Άσκηση 21

- Θεωρείστε τον ακόλουθο πίνακα τμημάτων:

| τμήμα | βάση | μήκος |
|-------|------|-------|
| 0     | 219  | 600   |
| 1     | 2300 | 14    |
| 2     | 90   | 100   |
| 3     | 1327 | 580   |
| 4     | 1952 | 96    |

Ποιες φυσικές διευθύνσεις αντιστοιχούν στις παρακάτω λογικές διευθύνσεις;

- **0,430**
- **1,10**
- **2,500**
- **3,400**
- **4,112**

# Άσκηση 22

Τρεις διεργασίες A,B,C φορτώνονται στην κεντρική μνήμη. Οι απαιτήσεις τους φαίνονται στον παρακάτω πίνακα:

| Process | Code Segment | Data Segment | Stack Segment |
|---------|--------------|--------------|---------------|
| A       | 492          | 438          | 2009          |
| B       | 4034         | 1030         | 610           |
| C       | 8900         | 914          | 1120          |

Το Λ.Σ. υποστηρίζει σελιδοποίηση με μέγεθος σελίδας 512 bytes. Ποιο ποσό φυσικής μνήμης σπαταλιέται για καθεμία από τις διεργασίες λόγω κατακερματισμού; Τι είδους είναι ο κατακερματισμός;

Υποθέστε ότι το Λ.Σ. (και το hardware) μπορεί να υποστηρίξει κατάτμηση με μέγεθος σελίδας 512 bytes. Κάθε τμήμα μπορεί να φορτωθεί ξεχωριστά. Ποιο ποσό φυσικής μνήμης σπαταλιέται λόγω κατακερματισμού; (θεωρείστε ότι κάθε τμήμα καταλαμβάνει ακέραιο αριθμό σελίδων και ότι ξεκινά στα όρια της σελίδας).

# Άσκηση 23

- Ένα σύστημα χρησιμοποιεί ως εικονικές διευθύνσεις 2048 σελίδες μεγέθους η κάθε μία 256 bytes και αντιστοιχείται σε μια φυσική μνήμη 512 πλαισίων. Η μικρότερη μονάδα προσπέλασης είναι 1 byte.
  1. μια διεργασία P1 χρησιμοποιεί 1053 bytes. Πόσα πλαίσια σελίδων θα απαιτήσει από την MMU (Memory Management Unit – μονάδα διαχείρισης μνήμης);
  2. μια άλλη διεργασία P2 απαιτεί 4000 bytes. Θεωρίστε ότι η P2 έχει ήδη αποκτήσει μέγεθος φυσικής μνήμης 2048 bytes, που ξεκινά από τη διεύθυνση 0. Έστω ότι ο επεξεργαστής χρειάζεται να προσπελάσει τη διεύθυνση : 10η σελίδα, μετατόπιση 34, στο πρόγραμμα της διεργασίας P2. Μπορεί να υπάρξει σφάλμα σελίδας; Δικαιολογείστε την απάντησή σας.

# Άσκηση 24

- Ένα σύστημα εικονικής μνήμης χρησιμοποιεί σελιδοποίηση με εικονικές διευθύνσεις μεγέθους 64 bits και μέγεθος σελίδας 16KB. Κάθε είσοδος στον πίνακα σελίδων απαιτεί 128 bits. Ο πίνακας σελίδων πρέπει να χωρά σε μία σελίδα. Πόσα επίπεδα πινάκων σελίδων απαιτούνται και πόσες εισοδοι υπάρχουν στον πίνακα σελίδων σε κάθε επίπεδο;

# Άσκηση 25

- Σε ένα σύστημα που χρησιμοποιεί σελιδοποίηση κάθε είσοδος στον πίνακα σελίδων έχει μέγεθος 4 bytes. Το μέγεθος σελίδας είναι 4K. Αν ο εικονικός χώρος διευθύνσεων χρησιμοποιεί 64 bits και με την προϋπόθεση ότι ο πίνακας σελίδων χωρά σε μια σελίδα :
- μπορεί το σύστημα να χρησιμοποιεί απλή σελιδοποίηση ; Να δικαιολογήσετε την απάντησή σας.
- αν απαιτείται σελιδοποίηση πολλαπλών επιπέδων , διατηρώντας τις προηγούμενες παραδοχές (δηλαδή ότι ο εικονικός χώρος διευθύνσεων χρησιμοποιεί 64 bits και κάθε πίνακας σελίδων χωρά σε μια σελίδα μεγέθους 4K) να βρείτε πόσα τουλάχιστον επίπεδα πρέπει να χρησιμοποιήσει μια σελιδοποίηση πολλαπλών επιπέδων.

# Άσκηση 26

Θεωρείστε μια αρχιτεκτονική λογικής μνήμης με τις ακόλουθες παραμέτρους :

- Μνήμη που διευθυνσιοδοτείται ανά byte
- 24-bit λογικές διευθύνσεις
- Μέγεθος segment 2 MB
- Μέγεθος σελίδας 512 bytes

Κάθε πίνακας σελίδων πρέπει να χωρά σε μια σελίδα. Υποθέστε ότι κάθε καταχώρηση στον πίνακα σελίδων περιέχει ένα read και ένα write bit προστασίας και ότι το μέγιστο ποσό φυσικής μνήμης που υποστηρίζεται από την αρχιτεκτονική είναι 8MB. Σχεδιάστε τη λογική διεύθυνση, δείχνοντας τον αριθμό των bits που χρησιμοποιούνται ως δείκτης σε κάθε επίπεδο των πινάκων σελίδων. (Υπόδειξη: Πρέπει να υπολογίσετε το μέγεθος κάθε καταχώρησης στον πίνακα σελίδων. Ως ενδιάμεσο βήμα, ίσως θεωρήσετε χρήσιμο να υπολογίσετε τον αριθμό που χωρούν στη φυσική μνήμη. Μετά, πόσα bits σε κάθε PTE απαιτούνται για να σχεδιάσετε μία από αυτές τις σελίδες; Προσθέστε τα 2 bits προστασίας για να υπολογίσετε το μέγεθος κάθε καταχώρησης.)



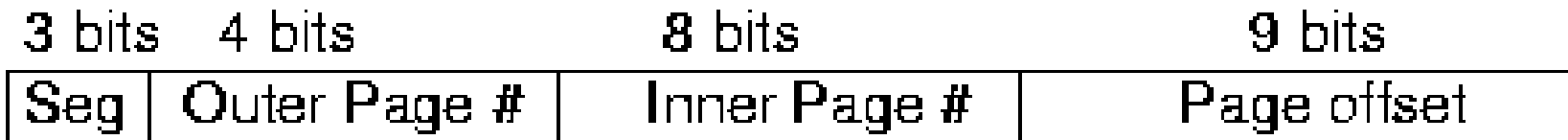
# Λύση 26

- Από τις αρχικές παραμέτρους προκύπτει :
- Μέγεθος segment  $2\text{MB}=2*1\text{MB}=2*2^{20}=2^{21}$  bits
- Εφόσον η λογική διεύθυνση έχει 24 bits απομένουν  $24-21=3$  bits για τον αριθμό segments
- Επειδή μέγεθος σελίδας = 512 bytes =  $2^9$  bits =offset και απομένουν  $21-9=12$  bits για τον αριθμό σελίδας (page number)

|        |         |             |
|--------|---------|-------------|
| 3 bits | 12 bits | 9 bits      |
| Seg    | Page #  | Page offset |

- Με μέγεθος φυσικής μνήμης  $8\text{MB}=2^{23}$  bits για τον αριθμό πλαισίων απομένουν  $23-9=14$  bits (ο αριθμός πλαισίων αναφέρεται και ως πλήθος των σελίδων φυσικής μνήμης (Physical Page Number))
- (με τον γνωστό τρόπο το πλήθος των σελίδων φυσικής μνήμης είναι  $2^{23}/2^9=2^{14}$  δηλαδή απαιτούνται 14 bits για το πλήθος των σελίδων φυσικής μνήμης).

- Άρα κάθε καταχώρηση (entry) στον πίνακα σελίδων απαιτεί 14 +2 (read-write) = 16 bits = 2 bytes
- Για να χωρά κάθε πίνακας σελίδων σε μια σελίδα (=2<sup>9</sup> bits) ο μέγιστος αριθμός καταχωρήσεων θα είναι 2<sup>9</sup>/2=2<sup>8</sup>
- Άρα το πολύ 8 bits μπορούν να χρησιμοποιηθούν ως δείκτης στον πίνακα σελίδων που χωρά σε μια σελίδα.
- Επειδή βρήκαμε ότι το πλήθος των bits που χρησιμοποιούνται για σελίδες είναι 12 έχουμε τουλάχιστον σελιδοποίηση με ένα επίπεδο 8 bits και ένα με 4 bits (δηλαδή συνολικά 2 επίπεδα).
- Από αυτά το επίπεδο των 4 bits είναι το εξωτερικό (πλησιέστερα προς το segment) και το επίπεδο των 8 bits το εσωτερικό (πλησιέστερα προς το offset) :



# Άσκηση 27

- Δίνεται η παρακάτω μορφή λογικής διεύθυνσης :

|        |         |             |
|--------|---------|-------------|
| 2 bits | 16 bits | 8 bits      |
| Seg    | Page #  | Page offset |

Να βρεθούν :

2. Το πλήθος των segments
3. Το μέγιστο μέγεθος κάθε segment
4. Το μέγεθος κάθε σελίδας
5. Το μέγιστο πλήθος σελίδων ανά segment
6. Το μέγιστο μέγεθος κάθε πίνακα σελίδας (ανά τμήμα). Υποθέστε ότι κάθε καταχώρηση στον πίνακα σελίδων απαιτεί 4 bytes

# Άσκηση 28

- Ένα σύστημα χρησιμοποιεί διευθύνσεις των 32 bits και έχει κεντρική μνήμη 4MB. Το μέγεθος σελίδας είναι 1K. Ποιο είναι το μέγεθος του πίνακα σελίδων;

## Λύση 28

- Ο πίνακας σελίδων περιέχει το πεδίο του αριθμού σελίδας της εικονικής διεύθυνσης. Έτσι το πλήθος των γραμμών στον πίνακα σελίδων είναι ίσο με τον αριθμό των εικονικών σελίδων:
  - Εικονικές σελίδες =  $2^{32}/2^{10}=2^{22}=4\text{M}$  σελίδες
- Το πλάτος κάθε γραμμής στον πίνακα σελίδων είναι ίσο με το πλάτος του αριθμού σελίδας στην εικονική διεύθυνση δηλ. 22 bits. Άρα το μέγεθος του πίνακα σελίδων είναι :
  - Μέγεθος πίνακα σελίδων = πλήθος γραμμών \* μέγεθος γραμμής
  - Μέγεθος πίνακα σελίδων =  $2^{22} * 22 \text{ bits} = 11.5 \text{ Mbytes!!}$

## Άσκηση 29

- Ένα σύστημα χρησιμοποιεί χώρο διευθύνσεων 32 bits και έχει μέγεθος σελίδας 8K. Ο πίνακας σελίδων βρίσκεται εξ ολοκλήρου στο υλικό και κάθε είσοδος του έχει μήκος 32 bits. Όταν μια διεργασία καθίσταται εκτελέσιμη ο πίνακας σελίδων αντιγράφεται από το υλικό στη μνήμη με ταχύτητα 100nsec για κάθε είσοδό του. Αν κάθε διεργασία εκτελείται για 100msec (περιλαμβάνεται ο χρόνος φόρτωσης του πίνακα σελίδων), να βρεθεί το ποσοστό % του χρόνου της CPU που αφιερώνεται για τη φόρτωση του πίνακα σελίδων, για κάθε διεργασία.

## Άσκηση 30

- Ένα σύστημα κρατά τους πίνακες σελίδων των διεργασιών στην κύρια μνήμη. Ο χρόνος ανάγνωσης μιας εισόδου του πίνακα σελίδων είναι 50nsec. Για να μειωθεί η καθυστέρηση το σύστημα χρησιμοποιεί TLB στο οποίο η αναζήτηση διαρκεί 10nsec. Ποιο πρέπει να είναι το ποσοστό επιτυχίας (hit – rate) του TLB ώστε ο χρόνος ανάγνωσης να μειωθεί σε 20nsec;

# Άσκηση 31

- Ένα σύστημα σελιδοποίησης χρησιμοποιεί διευθύνσεις των 16 bits, με μέγεθος σελίδας 4K και συνολικά 8 πλαίσια φυσικής μνήμης. Δίνονται οι πίνακες σελίδων των διεργασιών P1 και P2.

|   |    |   |    |
|---|----|---|----|
|   | P1 |   | P2 |
| 0 | 0  | 0 | 3  |
| 1 | 4  | 1 | 1  |
| 2 | 5  | 2 | 7  |
| 3 | 2  | 3 | 6  |

- Να βρεθούν οι φυσικές διευθύνσεις των παρακάτω λογικών διευθύνσεων:
  - Λογική διεύθυνση 15000 της διεργασίας P1
  - Λογική διεύθυνση 12000 της διεργασίας P2
- Να δώσετε αναλυτικά τα βήματα που οδηγούν στο τελικό αποτέλεσμα.

# Άσκηση 32

- Ένα σύστημα χρησιμοποιεί απλή σελιδοποίηση μνήμης με μέγεθος σελίδας 1024bytes. Το μέγιστο μέγεθος του χώρου διευθύνσεων μιας διεργασίας είναι 16 MB ενώ η μνήμη του συστήματος είναι 2MB. Ο πίνακας σελίδων της τρέχουσας διεργασίας περιέχει τα εξής :

| page | Frame number |
|------|--------------|
| 0    | 4            |
| 1    | 8            |
| 2    | 16           |
| 3    | 17           |
| 4    | 9            |

1. πόσα bits απαιτούνται για κάθε είσοδο του πίνακα σελίδων?
2. ποιο είναι το μέγιστο μέγεθος εισόδων σε ένα πίνακα σελίδων;
3. πόσα bits υπάρχουν σε μια εικονική διεύθυνση;
4. σε ποια φυσική διεύθυνση αντιστοιχεί η εικονική διεύθυνση 1524;
5. σε ποια εικονική διεύθυνση μεταφράζεται η φυσική διεύθυνση 10020?