

**Θέμα 1 (10)**

Σε ένα σύστημα που χρησιμοποιεί σελιδοποίηση, τα όρια των σελίδων/πλαισίων δε γίνονται αντιληπτά από τον προγραμματιστή. Εξηγήστε πως ένα loop σε ένα πρόγραμμα μπορεί να προκαλέσει σταδιακή κάμψη της απόδοσης του συστήματος (thrashing) σε ένα σύστημα σελιδοποίησης με στατική ανάθεση σελίδων, όταν ανατίθεται λίγη μνήμη στις διεργασίες.

**Θέμα 2 (10)**

Θεωρείστε τους ακόλουθους πίνακες τμημάτων (segment tables) για τις διεργασίες A και B:

**Πίνακας Τμημάτων Διεργασίας A**

# Τμήματος	Διεύθυνση Βάσης	Μέγεθος
0	700	200
1	400	200
2	1000	400

**Πίνακας Τμημάτων Διεργασίας B**

# Τμήματος	Διεύθυνση Βάσης	Μέγεθος
0	0	400
1	600	100
2	2000	300
3	1500	200

α) Αν μια λογική διεύθυνση αποτελείται από τέσσερα δεκαδικά ψηφία και το πρώτο ψηφίο είναι ο αριθμός του τμήματος ενώ τα τελευταία τρία η μετατόπιση, μεταφράστε τις ακόλουθες λογικές διευθύνσεις σε φυσικές, εφόσον αυτό είναι εφικτό. Αλλιώς αναφέρετε το είδος της μη έγκυρης πρόσβασης που δημιουργεί η φυσική διεύθυνση.

Διεργασία	Λογική Διεύθυνση
A	2350
A	1125
A	0021
B	0055
B	1036
B	4005
B	3305

β) Η προσέγγιση της κατάτμησης της μνήμης μπορεί να δημιουργήσει εξωτερικό κατακερματισμό; Αν, ναι εξηγήστε σε ποια περίπτωση και δώστε μια λύση για το πρόβλημα του εξωτερικού κατακερματισμού στην κατάτμηση της μνήμης.

**Θέμα 3 (15)**

Υποθέστε έναν αλγόριθμο χρονοπρογραμματισμού που να ευνοεί τα προγράμματα που έχουν χρησιμοποιήσει λίγο χρόνο επεξεργαστή στο πρόσφατο παρελθόν. Να εξηγήσετε γιατί αυτός ο αλγόριθμος ευνοεί τα προγράμματα που προορίζονται για E/E (I/O bound) χωρίς να στερεί μόνιμα το χρόνο του επεξεργαστή στα προγράμματα που προορίζονται για επεξεργασία (CPU bound).

**Θέμα 4 (10)**

Θεωρείστε μνήμη τεσσάρων πλαισίων αρχικά άδεια. Υπολογίστε το πλήθος των σφαλμάτων σελίδας που θα δημιουργηθούν από την ακόλουθη συμβολοσειρά αναφοράς ενός προγράμματος σε σελίδες, για τους αλγόριθμους FIFO, LRU και LFU.

1 2 3 4 5 3 4 1 6 7 8 7 8 9 7 8 9 5 4 5 4 2

**Θέμα 5 (10)**

Παρουσιάστε τρία πλεονεκτήματα που έχει η σπονδυλωτή δομή στα Λειτουργικά Συστήματα σε σχέση με τα μονολιθικά συστήματα.

**Θέμα 6 (10)**

Μια τυπική τυπωμένη σελίδα αποτελείται από 50 γραμμές των 80 χαρακτήρων η κάθε μια. Φανταστείτε ότι ένας συγκεκριμένος εκτυπωτής μπορεί να τυπώνει 6 σελίδες/λεπτό και ο χρόνος για να γραφεί ένας χαρακτήρας στον καταχωρητή εξόδου του εκτυπωτή είναι τόσο μικρός ώστε να μπορεί να αγνοηθεί. Προτείνετε μια λύση για τη λειτουργία Ε/Ε του εκτυπωτή, αν αυτός διαθέτει εσωτερική περιοχή προσωρινής αποθήκευσης για να αντιγράψει το χαρακτήρα. Αν υποθέσουμε τώρα ότι ο εκτυπωτής εκτυπώνει αμέσως το χαρακτήρα μόλις φτάσει σε αυτόν και κάθε τυπωμένος χαρακτήρας απαιτεί μια διακοπή που χρειάζεται 50  $\mu\text{sec}$  (συνολικά) να εξυπηρετηθεί, τι ποσοστό του χρόνου της ΚΜΕ «χάνεται» για την εκτύπωση μιας σελίδας αν χρησιμοποιηθεί Ε/Ε οδηγούμενη από τις διακοπές;

**Θέμα 7 (10)**

Ένας υπολογιστής του οποίου οι διεργασίες έχουν μέσα στο χώρο τους 1024 σελίδες, κρατάει τους πίνακες σελίδων του στη μνήμη. Η επιβάρυνση που απαιτείται για να διαβαστεί μια λέξη από τον πίνακα σελίδων είναι 500nsec. Για να μειωθεί αυτή η επιβάρυνση, ο υπολογιστής διαθέτει μια συσχετιστική μνήμη, η οποία διατηρεί 32 ζευγάρια (ιδεατή σελίδα – φυσικό πλαίσιο σελίδας) και μπορεί να προσπελαστεί σε 100nsec. Ποιος ρυθμός αστοχιών απαιτείται έτσι ώστε να μειωθεί η συνολική επιβάρυνση της σελιδοποίησης σε 200nsec;

**Θέμα 8 (5)**

Πόσες ενέργειες δίσκου χρειάζονται για να μεταφερθεί στο Λειτουργικό Σύστημα ο κόμβος *i* του αρχείου /usr/ast/sources/os/handout.t; Υποθέστε ότι ο κόμβος *i* που αντιστοιχεί στο βασικό κατάλογο (/) βρίσκεται στη μνήμη, αλλά δε βρίσκεται εκεί τίποτε άλλο που έχει να κάνει με τη διαδρομή. Υποθέστε επίσης ότι όλοι οι κατάλογοι χωρούν σε ένα μπλοκ δίσκου (συμπεριλαμβανομένου της πληροφορίας του κόμβου *i* για τον κατάλογο). Θεωρείστε ότι ο κόμβος *i* του αρχείου βρίσκεται στο πρώτο του μπλοκ.

**Θέμα 9 (15)**

Ένα σύστημα έχει τέσσερις διεργασίες και πέντε πόρους. Η τρέχουσα ανάθεση και οι μέγιστες ανάγκες φαίνονται παρακάτω:

	Allocated	Maximum	Available
Process A	1 0 2 1 1	1 1 2 1 3	0 0 x 1 2
Process B	2 0 1 1 0	2 2 2 1 0	
Process C	1 1 0 1 0	2 1 3 1 0	
Process D	1 1 1 1 0	1 1 2 2 1	

Ποια η μικρότερη τιμή του  $x$  για την οποία αυτή είναι μια ασφαλής κατάσταση;

**Θέμα 10 (5)**

Παρουσιάστε ορισμένους λόγους οι οποίοι να εξηγούν την ευρεία χρήση του χρονοπρογραμματισμού πολλών επιπέδων.