ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΠΛ 222 — ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ (4 Δ.Μ.)

Ακαδημαϊκό Έτος 1999-2000, 4ο-6ο Εξάμηνο

Τελικές Εξετάσεις

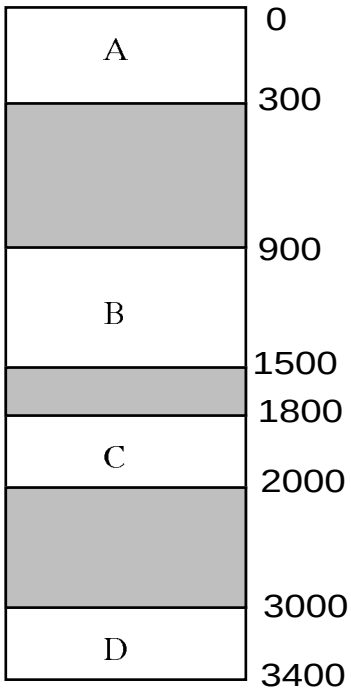
Ημερομηνία	:	19 Μαΐου 2000
Διάρκεια εξέτασης	:	2 ώρες
Διδάσκων καθηγητής	:	Γιώργος Α. Παπαδόπουλος

Απαντήστε ΟΛΕΣ τις ερωτήσεις. Όλες οι ερωτήσεις είναι ισοδύναμες σε αριθμό μονάδων.

1. Θεωρείστε το ακόλουθο πρόβλημα συγχρονισμού που προκύπτει στην υλοποίηση της ακόλουθης πολιτικής διαχείρησης της κύριας μνήμης: Σε έναν Η/Υ υπάρχει `M` διαθέσιμη κύρια μνήμη. Κάθε καινούργια διεργασία `i` μόλις ενεργοποιηθεί για πρώτη φορά, παίρνει από το Λ. Σ. ένα αρχικό ποσό μνήμης `K` < `M`. Το Λ. Σ. δεν θα αφήσει μία νέα διεργασία να ενεργοποιηθεί αν δίνοντάς της `K` μνήμη θα ξεπερνούσε τη συνολική διαθέσιμη ποσότητα `M`. Όταν μία διεργασία τερματίσει, απελευθερώνει όλη τη μνήμη που έχει δεσμεύσει. Κατά τη διάρκεια εκτέλεσής της, μία διεργασία `j` (που έχει ήδη αρχικά δεσμεύσει `K` μνήμη) μπορεί να ζητήσει επιπλέον μνήμη `A`. Αν το ζητούμενο ποσόν μνήμης `A` μπορεί να δοθεί στη διεργασία `j` (δηλαδή δεν υπερβαίνεται ο αριθμός `M`) τότε αυτό και γίνεται. Αν το Λ. Σ. δεν μπορεί να ικανοποιήσει την αίτηση της `j`, τότε αφαιρεί από τη `j` όλη τη μνήμη που μέχρι εκείνη τη στιγμή έχει αυτή η διεργασία δεσμεύσει (η `j` μπορεί στο μέλλον να ξαναζητήσει μνήμη από το Λ. Σ.)

   Γράψτε έναν παρακολουθητή ο οποίος να υποστηρίζει τις ακόλουθες διαδικασίες: `start(i,k)` η οποία δείνει σε μία διεργασία `i` το αρχικό ποσόν μνήμης `K`, `add(i,a)` η οποία δείνει επιπλέον μνήμη `a` στη διεργασία `i`, και `terminate(i)` η οποία τερματίζει την εκτέλεση της διεργασία `i`. Η ιδέα είναι ότι τα διάφορα μέρη του Λ. Σ. μπορούν με μη προκαθορισμένο τρόπο (και χωρίς συννενόηση μεταξύ τους) να προσπαθούν να δίνουν μνήμη σε διεργασίες, καλώντας αυτές τις διαδικασίες του παρακολουθητή. Βεβαιωθείτε ότι η λύση σας δεν υποφέρει από προβλήματα παρατεταμένης στέρησης και αδιέξοδου.

2. Α) Η κύρια μνήμη σε έναν Η/Υ βρίσκεται στην ακόλουθη κατάσταση, όπου τα γράμματα αναφέρονται σε διεργασίες που έχουν ήδη καταλάβει κάποιον χώρο στη μνήμη και ο σκαιώδης χώρος είναι διαθέσιμος:



Θεωρείστε ότι το σύστημα μπορεί να μετατοπίσει διεργασίες στη μνήμη και όταν χρειάζεται να κάνει συγχώνευση (compaction). Κατά τη συγχώνευση, οι διεργασίες μετατοπίζονται μία-μία από τις υψηλότερες στις χαμηλότερες διευθύνσεις μνήμης, ξεκινώντας από αυτή τη διεργασία που βρίσκεται πιο κοντά στη θέση 0. Η συγχώνευση τερματίζεται μόλις δημιουργηθεί αρκετά μεγάλος χώρος για να ικανοποιηθεί η αίτηση που προκάλεσε την έναρξη του μηχανισμού συγχώνευσης.

Ας υποθέσουμε ότι θα γίνουν τα ακόλουθα: (i) Μία διεργασία `E` ξεκινάει και ζητάει 300 μονάδες μνήμης. (ii) Η διεργασία `A` ζητάει 400 επιπλέον μονάδες μνήμης. (iii) Η διεργασία `B` τερματίζει την εκτέλεσή της. (iv) Μία διεργασία `F` ξεκινάει και ζητάει 800 μονάδες μνήμης. (v) Η διεργασία `C` τερματίζει την εκτέλεσή της. (vi) Μία διεργασία `G` ξεκινάει και ζητάει 900 μονάδες μνήμης.

Περιγράψτε την κατάσταση της κύριας μνήμης (σχηματικά ή με αναφορά σε διευθύνσεις μνήμης), μετά από την ικανοποίηση κάθε μιας από τις ανωτέρω 6 αιτήσεις, με βάση τις πολιτικές: α) first-fit, β) best-fit, γ) worst-fit. Κατόπιν, επιχειρηματολογήστε για το ποια από αυτές τις πολιτικές είναι η πιο αποτελεσματική για το συγκεκριμένο παράδειγμα.

Β) Ο χώρος των λογικών διευθύνσεων μνήμης μιας διεργασίας είναι 4 σελίδες, με 1024 bytes ανά σελίδα, απεικονιζόμενη σε μία φυσική μνήμη 64 πλαισίων σελίδων. (i) Πόσα bits υπάρχουν σε μία λογική διεύθυνση; (ii) Πόσα bits υπάρχουν σε μία φυσική

διεύθυνση; (iii) Αν υπάρχει ο ακόλουθος πίνακας σελίδων: `Σελίδα 0 -> Πλαίσιο 3, Σελίδα 1 -> Πλαίσιο 14, Σελίδα 2 -> Πλαίσιο 6, Σελίδα 3 -> Πλαίσιο 33`, σε ποια φυσική διεύθυνση αντιστοιχεί η λογική διεύθυνση `σελίδα 2, byte 256`.

3. Μία ομάδα πέντε διεργασιών καταφθάνει για εκτέλεση στο σύστημα την ίδια χρονική στιγμή 0 και κάθε μια από αυτές έχει τα ακόλουθα χαρακτηριστικά:

Διεργασία	**Συνολικός χρόνος εκτέλεσης**	**Προτεραιότητα**
Π1	10	3
Π2	1	1
Π3	2	3
Π4	1	4
Π5	5	2

Για κάθε έναν από τους αλγόριθμους χρονοδρομολόγησης (i) FCFS, (ii) SJF, (iii) Προτεραιότητα χωρίς προεκχώρηση (μικρότερος αριθμός = μεγαλύτερη προτεραιότητα), (iv) Εκ περιτροπής (κβάντο = 1), υπολογήστε για κάθε διεργασία:

- Τον χρόνο του κύκλου διεκπεραίωσης (turnaround time)
- Τον χρόνο αναμονής (waiting time)

Τέλος, υπολογήστε ποιος από τους ανωτέρω αλγόριθμους είναι ο καλύτερος με την έννοια ότι αυτός ο αλγόριθμος έχει τον ελάχιστο μέσο χρόνο αναμονής για όλες τις διεργασίες.

Καλή Επιτυχία!