ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΠΛ 222 — ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ (3 Δ.Μ.)

Ακαδημαϊκό Έτος 1996-97, 6ο Εξάμηνο

Εξέταση Ημιεξαμήνου

Ημερομηνία	:	17 Απριλίου 1997
Διάρκεια εξέτασης	:	1:30 ώρες
Διδάσκων καθηγητής	:	Γιώργος Α. Παπαδόπουλος

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

1. Σε ένα σύστημα καταφθάνει για εκτέλεση μία ομάδα διεργασιών με τα ακόλουθα χαρακτηριστικά:

Διεργασία	Εκτιμ. Χρόνος Εκτέλεσης	Προτεραιότητα	Χρόνος Άφιξης
P1	50 ms	4	0 ms
P2	20 ms	1	20 ms
P3	100 ms	3	40 ms
P4	40 ms	2	60 ms

Θεωρείστε τους ακόλουθους αλγόριθμους χρονοδρομολόγησης: (i) Η μικρότερη διεργασία πρώτη (SPN) με δικαίωμα προεκχώρησης, (ii) Με βάση την προτεραιότητα *χωρίς* δικαίωμα προεκχώρησης (1= μεγαλύτερη, 4=μικρότερη), (iii) Εκ περιτροπής (RR) με κβάντο 30 ms.

Γ'ια κάθε έναν από τους ανωτέρω αλγόριθμους:

α) Δώστε συνοπτικά το διάγραμμα χρονοδρομολόγησης των διεργασιών όπου να φαίνεται σε ποια χρονική στιγμή γίνεται αντικατάσταση διεργασιών στην ΚΜΕ (process switching) και σε ποια χρονική στιγμή ολοκληρώνει την εκτέλεσή της η κάθε διεργασία.

β) Υπολογείστε το μέσο χρόνο αναμονής κάποιας διεργασίας πριν αρχίσει να εκτελείται.

γ) Αν οι διεργασίες καταφθάνουν στο σύστημα με ρυθμό 10 διεργασιών ανά δευτερόλεπτο, υπολογίστε τον ελάχιστο αριθμό διεργασιών που θα βρίσκονται στην κατάσταση (ουρά) έτοιμη-για-εκτέλεση.

2. Θεωρείστε την ύπαρξη των ακόλουθων μεταβλητών με αντίστοιχες αρχικές τιμές:

```
integer x=0, y=23;     general semaphore m=1, ml=0, mx=1;
```

Επίσης, θεωρείστε την ύπαρξη των ακόλουθων μικρών προγραμμάτων όπου κάθε πρόγραμμα είναι της μορφής `cobegin <ομάδα εντολών 1> || <ομάδα εντολών 2> coend` με την ερμηνεία ότι οι εντολές που αποτελούν μία ομάδα εντολών εκτελούνται *σειριακά* η μία μετά την άλλη αλλά οι δύο ομάδες εντολών που περικλείονται σε ένα μπλοκ `cobegin…coend` εκτελούνται ταυτόχρονα με μη προκαθορισμένη συμπεριφορά:

i.
```
cobegin x:=x+1 || x:=y+1 coend
```

ii.
```
cobegin
wait(m); x:=x+1; signal(m)
||
wait(m); x:=y+1; signal(m)
coend
```

iii.
```
cobegin
wait(ml); x:=x+1; signal(m)
||
wait(m); x:=y+1; signal(ml)
coend
```

iv.
```
cobegin
wait(ml); x:=x+1; signal(m)
||
wait(ml); x:=y+1; signal(ml)
coend
```

v.
```
cobegin
wait(mx); wait(m); x:=x+1; signal(mx); signal(m)
||
wait(m); wait(mx); x:=y+1; signal(m); signal(mx)
coend
```

Γ`ια κάθε ένα από τα ανωτέρω προγράμματα (τα οποία, παρεμπιπτόντως, είναι τελείως ανεξάρτητα το ένα από το άλλο) κάνετε ένα από τα ακόλουθα:

α) Αν το πρόγραμμα τερματίζει κανονικά πάντα, δώστε την τελική τιμή της μεταβλητής x.

β) Αν το πρόγραμμα τερματίζει κανονικά μερικές φορές μόνο, αναφέρατέ το και για αυτές τις περιπτώσεις που τερματίζει δώστε πάλι την τελική τιμή της μεταβλητής x.

γ) Αν το πρόγραμμα δεν τερματίζει ποτέ, αναφέρατέ το.

3. Υλοποιείστε το πρόβλημα του κοιμώμενου κουρέα (sleepy barber) κάνοντας χρήση σημαφόρων για την επίτευξη συγχρονισμού μεταξύ κουρέα και πελατών: Ένα κουρείο διαθέτει έναν κουρέα, μία θέση για κούρεμα και Ν θέσεις αναμονής. Όταν δεν υπάρχουν πελάτες, ο κουρέας κοιμάται. Όταν έλθει κάποιος πελάτης ξυπνάει τον κουρέα για να τον εξυπηρετήσει. Αν εν τω μεταξύ έλθουν και άλλοι πελάτες περιμένουν στις θέσεις αναμονής. Αν έλθει κάποιος πελάτης ενώ όλες οι θέσεις αναμονής είναι κατειλημμένες, ο πελάτης φεύγει.

Γράψτε δύο συναρτήσεις: μία για τον κουρέα και μία για κάθε πελάτη. Ο κουρέας ή κοιμάται ή κουρεύει. Ο πελάτης ή περιμένει σε μία από τις θέσεις ή κουρεύεται.

Καλή Επιτυχία!