



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

www.cslab.ece.ntua.gr

ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ ΥΠΟΛΟΓΙΣΤΩΝ

Εξέταση Φεβρουαρίου 2010

Διάρκεια 2,5 ώρες

Οι εξετάσεις θα πραγματοποιηθούν ΧΩΡΙΣ την παρουσία βιβλίων, βοηθημάτων ή άλλου είδους σημειώσεων. Το μόνο που επιτρέπεται να έχετε μαζί σας είναι ένα φύλλο A4 στο οποίο μπορείτε να έχετε γράψει ό,τι έχετε κρίνει πιο σημαντικό για το μάθημα και θέλετε να το έχετε ως βοήθημά σας. Απαγορεύεται η ανταλλαγή οποιουδήποτε αντικειμένου κατά την ώρα της εξέτασης, ούτε και των φύλλων A4 που είναι ατομικά.

Θέμα 1 (20%)

α. (10%) Δίνεται ο παρακάτω κώδικας:

```
for (i = 0; i < 3; i++) {
    ret = fork();
    if (ret == 0) {
        for (j = 0; j < 2; j++) {
            ret = fork();
            if (ret == 0) {
                Fn(i, j);
                exit(0);
            }
        }

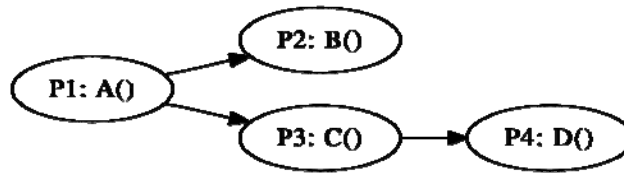
        for (j = 0; j < 2; j++)
            wait(NULL);
    }
}

for (i = 0; i < 3; i++)
    wait(NULL);
```

Θεωρούμε ότι η συνάρτηση **Fn()** δεν επιστρέφει ποτέ.

Σχεδιάστε το δέντρο διεργασιών που τελικά προκύπτει. Για κάθε κόμβο του δέντρου, να γράψετε την κλήση συστήματος ή συνάρτηση που εκτελεί εκείνη τη στιγμή η αντίστοιχη διεργασία, μαζί με τις τιμές των ορισμάτων της.

β. (10%) Σε αντιστοιχία με το προηγούμενο ερώτημα, ζητείται κώδικας που να υλοποιεί το παρακάτω δέντρο:



Θεωρήστε ότι οι συναρτήσεις $A()$, $B()$, $C()$, $D()$ δεν επιστρέφουν ποτέ κι ότι η `fork()` επιτυγχάνει πάντοτε.

Θέμα 2 (30%)

α. (10%) Έστω δύο νήματα (T_0 και T_1) που εκτελούν δύο συναρτήσεις:

| T_0 | T_1 |
|----------|----------|
| $A_0();$ | $A_1();$ |
| $B_0();$ | $B_1();$ |

Προσθέστε κατάλληλες εντολές συγχρονισμού με σημαφόρους, ώστε:

- Η συνάρτηση $B_0()$ να εκτελεστεί μετά την επιστροφή της $A_1()$
- Η συνάρτηση $B_1()$ να εκτελεστεί μετά την επιστροφή της $A_0()$

Υπόδειξη: χρησιμοποιήστε δύο σημαφόρους αρχικοποιημένους στην τιμή 0.

β. (20%) Έστω η γενίκευση του παρακάτω σχήματος συγχρονισμού για N νήματα. Το νήμα n εκτελεί δύο συναρτήσεις:

| T_n |
|----------|
| $A_n();$ |
| $B_n();$ |

Στόχος είναι να υλοποιηθεί σχήμα συγχρονισμού που να εξασφαλίζει ότι καμία συνάρτηση $B()$ δεν θα εκτελεστεί πριν επιστρέψουν όλες οι συναρτήσεις $A()$.

Έστω η παρακάτω λύση:

Μοιραζόμενες μεταβλητές:

```

count = 0;
lock = Semaphore(1);
barrier = Semaphore(0);
  
```

Κώδικας νήματος i :

```

Ai();

wait(lock);
count = count + 1;
signal(lock);

if (count == N)
    signal(barrier);

wait(barrier);

Bi();
  
```

- Δείξτε ότι οδηγεί σε αδιέξοδο.
- Υπάρχει αλληλουχία εκτέλεσης που δεν δημιουργεί αδιέξοδο; Δικαιολογήστε.
- Περιγράψτε σύντομα το ρόλο του σημαφόρου `lock`.
- Διορθώστε την παραπάνω λύση, ώστε να επιτυγχάνει το επιθυμητό αποτέλεσμα.

Θέμα 3 (25%)

α. (10%) Θεωρήστε υπολογιστικό σύστημα με έναν επεξεργαστή (ΚΜΕ) και μία μονάδα E/E. Έστω τρεις διεργασίες. Ο χρόνος άφιξης και τα ξεσπάσματά τους, σε μονάδες χρόνου, είναι:

| | P1 | P2 | P3 |
|---------------|---------|--------|--------|
| Χρόνος Άφιξης | 0 | 2 | 4 |
| Ξεσπάσματα | KME: 10 | KME: 6 | KME: 1 |
| | E/E: 10 | E/E: 5 | E/E: 1 |
| | | KME: 2 | KME: 2 |
| | | E/E: 2 | E/E: 3 |

Για την ΚΜΕ το Λειτουργικό Σύστημα χρησιμοποιεί χρονοδρομολόγηση με βάση τη μικρότερη διάρκεια εκτέλεσης (Shortest-Job-First – SJF) με διακοπές (διακοπτός αλγόριθμος – preemptive). Οι αιτήσεις E/E εξυπηρετούνται με βάση τη σειρά άφιξης (First Come, First Served – FCFS).

Αντιγράψτε και συμπληρώστε τον παρακάτω πίνακα εκτέλεσης:

| t | Ουρά ΚΜΕ | ΚΜΕ | Ουρά E/E | E/E |
|----|------------|-----|----------|-----|
| 0 | P1/10 | P1 | – | – |
| 2 | P1/8, P2/6 | ? | ? | ? |
| ? | | | | |
| 8 | P1/8, P2/3 | P2 | P3/3 | P3 |
| ? | | | | |
| 31 | – | – | – | – |

Σημείωση: Η στήλη «ΚΜΕ» δείχνει τη διεργασία που εκτελείται. Η στήλη «E/E» δείχνει τη διεργασία που κάνει E/E.

β. (10%) Προσθέτουμε ακόμη μία επεξεργαστική μονάδα. Ο χρονοδρομολογητής ΚΜΕ επιλέγει τις δύο διεργασίες με τη μικρότερη διάρκεια εκτέλεσης.

Αντιγράψτε και συμπληρώστε τον παρακάτω πίνακα εκτέλεσης:

| t | Ουρά ΚΜΕ | ΚΜΕ ₁ | ΚΜΕ ₂ | Ουρά E/E | E/E |
|----|------------------|------------------|------------------|--------------------------------|-----|
| 0 | P1/10 | P1 | – | – | – |
| ? | | | | | |
| 6 | P1/5, P2/2, P3/2 | P2 | P3 | – | – |
| 8 | P1/5 | P1 | – | P3/3, P2/5 (επιλέγουμε την P2) | P2 |
| ? | | | | | |
| 28 | – | – | – | – | – |

γ. (5%) Υπολογίστε το βαθμό χρησιμοποίησης της (των) ΚΜΕ στις περιπτώσεις (α), (β). Γιατί ο συνολικός χρόνος εκτέλεσης στην περίπτωση (β) δεν είναι ο μισός, παρόλο που προσθέσαμε ένα δεύτερο επεξεργαστή;

Θέμα 4 (25%)

α. (15%) Μια διεργασία έχει στη διάθεσή της τρία πλαίσια μνήμης κι εκτελεί την παρακάτω ακολουθία αναφορών σε σελίδες:

1, 3, 4, 3, 6, 1, 3, 4, 3

Θεωρώντας ότι τα πλαίσια είναι αρχικά άδεια, εκτελέστε τον αλγόριθμο αντικατάστασης σελίδων και βρείτε τον αριθμό σφαλμάτων σελίδας που συμβαίνουν με χρήση των ακόλουθων πολιτικών αντικατάστασης: (i) βέλτιστη, (ii) First-In First-Out (FIFO), (iii) Least-Recently Used (LRU).

β. (5%) Ένα λειτουργικό σύστημα διακρίνει τις εξής καταστάσεις για τις διεργασίες του:

Σε αναμονή, Έτοιμη, Υπό εκτέλεση

Μία διεργασία που εκτελείται προκαλεί σφάλμα σελίδας και τα δεδομένα της σελίδας χρειάζεται να ανακληθούν από το δίσκο σελιδοποίησης. Σε τι κατάσταση περνά η διεργασία έως ότου έρθουν τα δεδομένα από το δίσκο; Σε τι κατάσταση θα περάσει η διεργασία όταν η σελίδα έρθει τελικά στην κεντρική μνήμη; Δικαιολογήστε *σύντομα* τις απαντήσεις σας.

γ. (5%) Έστω Λειτουργικό Σύστημα το οποίο για την εκτέλεση νέων προγραμμάτων χρησιμοποιεί το μοντέλο **fork()/exec()**. Σχολιάστε την επίδραση της τεχνικής «Αντιγραφή κατά την εγγραφή» (Copy-on-Write) στην επίδοση της παραπάνω λειτουργίας.