



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
<http://www.cs1lab.ece.ntua.gr>

Λειτουργικά Συστήματα

7ο εξάμηνο, Ακαδημαϊκό Έτος 2015-2016

Επαναληπτική Εξέταση

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 Α4, 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 Α4 που είναι ατομικά.

Θέμα 1 (25%)

Δίνεται το πρόγραμμα C σε περιβάλλον UNIX που ακολουθεί μετά τα ζητούμενα. Θεωρήστε ότι οι κλήσεις συστήματος δεν αποτυγχάνουν και ότι κάθε νέα διεργασία κληρονομεί ακριβώς τον τρόπο χειρισμού σημάτων του πατέρα.

- i. Σχεδιάστε το γράφο διεργασιών στην τελική του μορφή, όταν δηλαδή όλες οι διεργασίες και τα νήματα έχουν φτάσει σε μόνιμη κατάσταση. Επισημάνετε τις διεργασίες με κυκλικό κόμβο και τα νήματα με τετράγωνο κόμβο στο γράφο. Για κάθε κόμβο φροντίστε να φαίνεται η γονική διεργασία με βέλος. Αν υπάρχουν, δείξτε και τις διεργασίες/νήματα που έζησαν προσωρινά, αλλά δεν υπάρχουν στη μόνιμη κατάσταση, με διακεκομμένο περίγραμμα. (10%)
- ii. Για κάθε κομβο του γράφου επισημάνετε: (i) την κλήση συστήματος ή συνάρτηση μέσα στην οποία βρίσκεται ο Program Counter της αντίστοιχης διεργασίας/νήματος, (ii) τα ορίσματα με τα οποία αυτή έχει κληθεί και (iii) τη γραμμή του προγράμματος από όπου έγινε η κλήση της. Για τις διεργασίες/νήματα που έζησαν προσωρινά επισημάνετε την τελευταία κλήση συστήματος ή συνάρτηση στην οποία βρέθηκε ο Program Counter. (5%)
- iii. Συμπληρώστε το σχήμα σας ώστε να φαίνεται η διαδιεργασιακή επικοινωνία: για κάθε μεταφορά δεδομένων ή αποστολή σήματος που συνέβη, σχεδιάστε ένα διακεκομμένο βέλος από τη διεργασία-αποστολέα στη διεργασία-παραλήπτη. Πάνω από το βέλος γράψτε την τιμή που μεταφέρεται κάθε φορά. (4%)
- iv. Τί θα τυπωθεί στις γραμμές 54 και 56 του προγράμματος; (6%)

Σημείωση: Από το man page της kill δίνεται: `int kill(pid_t pid, int sig);` *If pid equals 0, then sig is sent to every process in the process group of the calling process.*

```

1  int res = 0, num = 2, pd[2], arg[32];
2  pthread_t thread[32];
3  sem_t sem;
4
5  void h(int s)
6  {
7      read(pd[0], &num, sizeof(num));
8      sem_post(&sem);
9  }
10
11 void *f(void *var)
12 {
13     int tmp = *(int *)var;
14     res += tmp;
15     pthread_exit((void *)0);
16 }
17
18 int main()
19 {
20     int i, k, p, pstatus;
21     void *tstatus;
22
23     sem_init(&sem, 0, 0);
24     signal(SIGUSR1, h);
25     pipe(pd);
26
27     for (i = 0; i < num; i++) {
28         p = fork();
29
30         if (p == 0) {
31             sem_wait(&sem);
32             num++;
33             for (k = 0; k < num; k++) {
34                 arg[k] = k;
35                 pthread_create(&thread[k], NULL, f, (void *)&arg[k]);
36             }
37
38             for (k = 0; k < num; k++) pthread_join(thread[k], &tstatus);
39             if (num < 4) write(pd[1], &num, sizeof(num));
40
41             exit(res);
42         }
43     }
44
45     write(pd[1], &num, sizeof(num));
46     signal(SIGUSR1, SIG_IGN); /* Το SIGUSR1 αγνοείται από την καλούσα διεργασία */
47     kill(0, SIGUSR1); /* Δείτε επισήμανση στην εκφώνηση */
48
49     for (i = 0; i < num; i++) {
50         wait(&pstatus);
51         if (WIFEXITED(pstatus)) res += WEXITSTATUS(pstatus);
52     }
53
54     printf("Final result1: %d \n", res);
55     read(pd[0], &num, sizeof(num));
56     printf("Final result2: %d \n", num);
57
58     return 0;
59 }

```

Θέμα 2 (60%)

Δίνεται το ημιτελές πρόγραμμα που ακολουθεί μετά τα ερωτήματα, το οποίο εκτελείται σε σύστημα με τα παρακάτω χαρακτηριστικά:

- Το ΛΣ υποστηρίζει εικονική μνήμη με σελιδοποίηση και ακολουθεί το μοντέλο `fork()/exec()` με `Copy-on-Write (CoW)` για τη δημιουργία νέων διεργασιών.
- Το μέγεθος σελίδας/πλαίσου είναι 4096 bytes (4KiB). Η φυσική μνήμη περιλαμβάνει 10 πλαίσια.
- Σε κάθε πρόγραμμα που εκτελείται στο εν λόγω σύστημα, το ΛΣ παραχωρεί ξεχωριστή(ές) σελίδα(ες) μνήμης για τον κώδικα του προγράμματος, τις global μεταβλητές (αν υπάρχουν), τις σταθερές (αν υπάρχουν) και τη στοίβα, ενώ υποστηρίζει δυναμική παραχώρηση μνήμης από το σωρό. Η στοίβα κάθε προγράμματος είναι 4KiB.
- Το ΛΣ παραχωρεί φυσική μνήμη στο σωρό της διεργασίας όταν πραγματοποιείται εγγραφή στη μνήμη αυτή.

Ο κώδικας του παρακάτω προγράμματος είναι 4KiB, η βιβλιοθήκη της C (συναρτήσεις `fork()`, `malloc()`, `fprintf()`, `scanf()`, `fscanf()`) καταλαμβάνει 8KiB και είναι στατικά συνδεδεμένη με το πρόγραμμα, ενώ η μαθηματική βιβλιοθήκη (συνάρτηση `sqrt()`) καταλαμβάνει 4KiB και συνδέεται με το πρόγραμμα δυναμικά. Ο κώδικας και τα δεδομένα του ΛΣ καταλαμβάνουν επίσης από 4KiB στη μνήμη το καθένα. Υπενθυμίζεται ότι `sizeof(int) = 4`.

- Στους Πίνακες 1-3 σας δίνεται η εικόνα της φυσικής μνήμης ακριβώς πριν από την εκτέλεση του προγράμματος, καθώς και η μορφή του πίνακα σελίδων και του χάρτη μνήμης που το ΛΣ τηρεί για κάθε διεργασία. Δώστε την εικόνα της φυσικής μνήμης, καθώς και τον πίνακα σελίδων και το χάρτη μνήμης για κάθε διεργασία που βρίσκεται σε εκτέλεση στις γραμμές του κώδικα 9, 11, 18, 19 (αμέσως μετά τη `fork`), 23 και 28 (θεωρήστε ότι η γραμμή 22 εκτελείται πριν την 27). (30 %)
- Περιγράψτε αν και υπό ποιες συνθήκες στις γραμμές κώδικα 8, 10, 12, 13, 16, 19, 22 και 27 είναι δυνατόν να εμπλακεί ο χρονοδρομολογητής στην εκτέλεση του προγράμματος. Θεωρήστε ότι όλες οι συναρτήσεις και οι κλήσεις συστήματος επιτυγχάνουν. Αγνοήστε τις περιπτώσεις που ο χρονοδρομολογητής καλείται λόγω διακοπής από το χρονιστή. (10%)
- Συμπληρώστε το πρόγραμμα οπουδήποτε κρίνετε σκόπιμο, ώστε να πραγματοποιείται σωστά και παράλληλα (στις 2 διεργασίες που έχουν δημιουργηθεί) ο υπολογισμός. Το αποτέλεσμα της πράξης θα πρέπει να είναι ορατό στη διεργασία πατέρα. (10%)
- Δώστε έκδοση του παραπάνω προγράμματος χρησιμοποιώντας νήματα POSIX. (10%)

πλαίσιο	περιγραφή
0	κώδικας ΛΣ
1	δεδομένα ΛΣ
2	free
3	free
4	math lib
5	free
6	free
7	free
8	free
9	free

Πίνακας 1: Πλαίσια στη φυσική μνήμη

page <i>page number</i>	permissions <i>π.χ. r, rw, x, κλπ</i>	frame <i>αριθμός πλαισίου στη φυσική μνήμη</i>
0
...

Πίνακας 2: Πίνακας Σελίδων

page <i>page number</i>	permissions <i>π.χ. r, rw, x, κλπ</i>	block <i>αριθμός block στο δίσκο - ενδεικτικά</i>
0
...

Πίνακας 3: Χάρτης Μνήμης

```

1 #define SIZE 1024 // constant
2
3 int *a, i, t, ITER;
4
5 int main(void)
6 {
7     FILE *inFile;
8     inFile = fopen("1.txt", "r");
9
10    a = malloc(SIZE*sizeof(int));
11
12    fprintf(stdout, "Give me the number of iterations\n");
13    scanf("%d", &ITER);
14
15    for (i = 0; i < SIZE; i++) {
16        fscanf(inFile, "%d\n", &a[i]);
17    }
18
19    if (fork() == 0) {
20        for (t = 0; t < ITER; t++)
21            for (i = 1; i < SIZE/2; i++)
22                a[i] = sqrt((a[i-1] + a[i+1]) / 2);
23    }
24    else {
25        for (t = 0; t < ITER; t++)
26            for (i = SIZE/2; i < SIZE - 1; i++)
27                a[i] = sqrt((a[i-1] + a[i+1]) / 2);
28    }
29
30    return 0;
31 }
```

Θέμα 3 (15%)

Απαντήστε συνοπτικά στις παρακάτω ερωτήσεις:

- i. Αναφέρετε μηχανισμούς με τους οποίους το υλικό υποστηρίζει τις παρακάτω λειτουργίες:
 - Διαχωρισμός χώρου χρήστη και χώρου πυρήνα. (1%)
 - Διακοπή χρονοδρομολόγηση. (1%)
 - Εικονική μνήμη. (1%)
 - Αποδοτικός συγχρονισμός. (1%)
 - Εικονικοποίηση. (1%)
- ii. Ποια ζητήματα πρέπει να λάβει υπόψη του ο σχεδιαστής του χρονοδρομολογητή όταν στο σύστημα υπάρχουν πολλαπλοί επεξεργαστές; (4%)
- iii. Τι είναι το πρόβλημα της ενεργού αναμονής; Δώστε ψευδοκώδικα για την υλοποίηση των σημαφόρων με και χωρίς ενεργό αναμονή. (6%)