



# Λειτουργικά Συστήματα

2019 - 2020

## 2η Εργαστηριακή Άσκηση

### Μέρος 1 [70 μονάδες]

**Ερώτημα Α [5]:** Εξηγήστε προσεκτικά τι κάνει το παρακάτω πρόγραμμα.

- Πόσες διεργασίες υπάρχουν σε κατάσταση sleeping 10 δευτερόλεπτα μετά την έναρξή του;
- Τροποποιήστε κατάλληλα το κώδικα έτσι ώστε κάθε διεργασία να τυπώνει το id της και το id του γονέα της πριν βρεθεί σε κατάσταση sleeping.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int pid1;
    int pid2;
    pid1 = fork();
    if (pid1 < 0)
        printf("Could not create any child\n");
    else
    {
        pid2 = fork();
        if (pid2 < 0)
            printf("Could not create any child\n");
        else
            if ( (pid1 < 0) && (pid2 < 0) ) kill(pid1,9);
    }
    sleep(20);

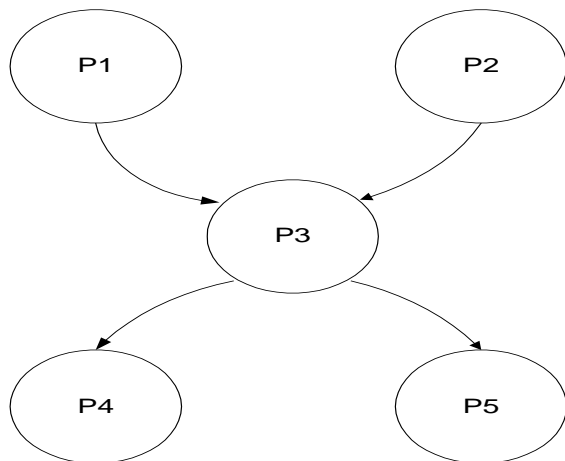
    return (0);
}
```

**Ερώτημα Β [25]:** Δημιουργήστε ένα πρόγραμμα στο οποίο μία διεργασία στο Linux/Unix παράγει άλλες 100 θυγατρικές της. Κάθε μία διαβάζει μία rand( ) ακέραια τιμή που δείχνει την προτεραιότητά της και τη γράφει **ατομικά** σε μία heap shared δομή. Συγχρονίστε τις παραπάνω διεργασίες, ώστε να εξασφαλίσετε αμοιβαίο αποκλεισμό στην προσπέλαση της heap δομής με απότερο στόχο την αποθήκευση της χαμηλότερης προτεραιότητας πάντα στη ρίζα της min heap δομής.

**Ερώτημα Γ [20]:** Γράψτε πρόγραμμα που να υλοποιεί με χρήση σημαφόρων το πρόβλημα συγχρονισμού των διαδικασιών ΑΝΑΓΝΩΣΗΣ (READ) και ΕΓΓΡΑΦΗΣ (WRITE) σε ένα κοινό διαμοιραζόμενο αρχείο (SHARED FILE). Θεωρείστε ότι πρέπει να επιτρέπουμε πολλές αναγνώσεις ταυτόχρονα, αλλά μόνο μία εγγραφή.



**Ερώτημα Δ [20]:** (1) Να γράψετε πρόγραμμα συγχρονισμού για τον παρακάτω γράφο προτεραιότητας κάνοντας χρήση σημαφόρων. Θεωρείστε ότι κάθε διεργασία εκτελεί μία εντολή του συστήματος της αρεσκείας σας . π.χ. `system("ls -l")` ή `system("ps -l")` κ.τ.λ... Επίσης θεωρείστε ότι οι P1, P2, P3, P4 και P5 είναι θυγατρικές μιας μόνο διεργασίας.



(2) Θεωρήστε ότι πρέπει να συγχρονίσετε την εκτέλεση των διαδικασιών Δ1, Δ2, Δ3, Δ4, Δ5 και Δ6 σύμφωνα με τους παρακάτω περιορισμούς:

- Η Δ1 εκτελείται πριν από τις Δ2 και Δ3.
- Η Δ2 εκτελείται πριν από τις Δ4 και Δ5.
- Η Δ3 εκτελείται πριν από την Δ5.
- Η Δ6 εκτελείται μετά από τις Δ3 και Δ4.

Η έκφραση «η διαδικασία Δ<sub>i</sub> εκτελείται πριν από τη διαδικασία Δ<sub>j</sub>» σημαίνει ότι η εκτέλεση της Δ<sub>i</sub> πρέπει να ολοκληρωθεί πριν αρχίσει η εκτέλεση της Δ<sub>j</sub>. Αναλόγως η έκφραση «η διαδικασία Δ<sub>i</sub> εκτελείται μετά από τη διαδικασία Δ<sub>j</sub>» σημαίνει ότι η εκτέλεση της Δ<sub>i</sub> μπορεί να αρχίσει αφού ολοκληρωθεί η εκτέλεση της Δ<sub>j</sub>.

Να γράψετε πρόγραμμα συγχρονισμού που θα ικανοποιεί τους προηγούμενους περιορισμούς κάνοντας χρήση σημαφόρων. Θεωρείστε ότι κάθε διεργασία εκτελεί μία εντολή του συστήματος της αρεσκείας σας . π.χ. `system("ls -l")` ή `system("ps -l")` κ.τ.λ... Επίσης θεωρείστε ότι οι Δ1, Δ2, Δ3, Δ4 και Δ5 είναι θυγατρικές μιας μόνο διεργασίας.

## Μέρος 2 [30 μονάδες]

### Ερώτημα Α [8]

Σε ένα σύστημα με σελιδοποίηση (paging) θεωρήστε ότι υποστηρίζονται λογικές διευθύνσεις των 32 bits όπου τα πρώτα (πιο σημαντικά) 18 bits αναπαριστούν τον αριθμό σελίδας κάθε διεύθυνσης.

(α) Έστω μία διεργασία η οποία αποτελείται από  $39500_{16}$  bytes. Αν θεωρήσουμε ότι η εν λόγω διεργασία είναι ολόκληρη φορτωμένη στη μνήμη, πόσα πλαίσια σελίδων καταλαμβάνει και πόση εσωτερική κλασματοποίηση προκαλεί;

(β) Υποθέστε στη συνέχεια ότι η εν λόγω διεργασία κατά το τρέχον χρονικό διάστημα έχει φορτωμένες στη μνήμη μόνο τις τελευταίες πέντε σελίδες της, στα ακόλουθα κατά σειρά πλαίσια (οι αριθμοί δίνονται στο δεκαδικό σύστημα): 16, 225, 170, 35, 51 (δηλαδή η τελευταία σελίδα της είναι φορτωμένη στο πλαίσιο 51, η προτελευταία στο πλαίσιο 35, η προ-προτελευταία στο πλαίσιο 170 κ.ο.κ.). Με βάση τα παραπάνω δεδομένα, υπολογίστε σε ποιες φυσικές διευθύνσεις αντιστοιχούν οι ακόλουθες λογικές διευθύνσεις της διεργασίας (δίνονται στο δεκαεξαδικό σύστημα / στο δεκαεξαδικό σύστημα θα πρέπει να δώσετε και τις απαντήσεις σας):

(i) 00031958<sub>16</sub>(ii) 0001E800<sub>16</sub>**Ερώτημα Β [8]**

Σε ένα σύστημα τμηματοποιημένης μνήμης (segmentation) θεωρήστε το παρακάτω μέρος του Πίνακα Τμημάτων μίας διεργασίας (όλοι οι αριθμοί του πίνακα δίνονται στο δεκαδικό σύστημα):

Αριθμός Τμήματος	Διεύθυνση Βάσης	Μήκος Τμήματος
0	1650	1110
1	3200	2350
2	10310	1290
.....	.....	.....
10	5950	2255
11	9050	1230
12	12270	5535
.....	.....	.....

Θεωρείστε επίσης ότι η κάθε λογική διεύθυνση αποτελείται από 32 bits και ότι το μέγιστο υποστηριζόμενο μέγεθος ενός τμήματος είναι 16 MBytes.

(α) Ποιος είναι ο μέγιστος υποστηριζόμενος αριθμός τμημάτων για μία διεργασία;

(β) Υπολογίστε σε ποιες φυσικές διευθύνσεις αντιστοιχούν οι λογικές διευθύνσεις που ακολουθούν (δίνονται στο δεκαεξαδικό σύστημα / στο δεκαεξαδικό σύστημα θα πρέπει να δώσετε και τις απαντήσεις σας):

(i) 0B00042A<sub>16</sub>(ii) 02000B6D<sub>16</sub>**Ερώτημα Γ [10]**

Έστω ότι στο παραπάνω σύστημα, με στόχο την αποδοτικότερη διαχείριση του συνολικού χώρου μνήμης, αποφασίστηκε για την εκχώρηση μνήμης στα τμήματα κάθε διεργασίας (λόγω του μεγάλου εν δυνάμει μεγέθους τους) να εφαρμοστεί η μέθοδος της σελιδοποίησης με μέγεθος σελίδας 512 bytes.<sup>1</sup>

(α) Υπολογίστε από πόσα bits αποτελείται κάθε ένα από τα τρία μέρη στα οποία χωρίζεται πλέον κάθε λογική διεύθυνση (αριθμός τμήματος, αριθμός σελίδας και μετατόπιση) στο νέο σχήμα μνήμης.

(β) Υπολογίστε από πόσες σελίδες μπορεί να αποτελείται κατά μέγιστο μία διεργασία στο νέο σχήμα μνήμης.

(γ) Έστω μία διεργασία η οποία αποτελείται από δύο τμήματα (Τμήμα 0 και Τμήμα 1), των οποίων οι Πίνακες Σελίδων (Π.Σ.) δίνονται παρακάτω (με '-' υπονοείται ότι η εν λόγω σελίδα δεν είναι φορτωμένη στη μνήμη, ενώ με '?' υπονοείται ότι είναι μεν φορτωμένη στη μνήμη αλλά δεν ξέρουμε σε ποιο πλαίσιο έχει φορτωθεί).

(i) Μετατρέψτε τη λογική διεύθυνση 010004CF<sub>16</sub> της διεργασίας αυτής στην αντίστοιχη φυσική διεύθυνση (δίνεται στο δεκαεξαδικό σύστημα / στο ίδιο σύστημα θα πρέπει να δώσετε και την απάντησή σας).

(ii) Συμπληρώστε τα ερωτηματικά '?' των δύο Π.Σ. που σας δίνονται, θεωρώντας ως δεδομένα ότι:

- η λογική διεύθυνση 010009FF<sub>16</sub> προκαλεί σφάλμα σελίδας (page fault), και

- η λογική διεύθυνση 000003F0<sub>16</sub> αντιστοιχεί στη φυσική διεύθυνση E0E1F0<sub>16</sub>

<sup>1</sup> Διαμορφώνοντας έτσι ένα σύστημα μνήμης σελιδοποιημένης τμηματοποίησης (paged segmentation) στο οποίο το εύρος σε bits του αριθμού τμήματος s παραμένει το ίδιο όπως ήταν αρχικά ενώ το εύρος σε bits της υπόλοιπης λογικής διεύθυνσης (μετατόπιση d) χωρίστηκε σε αριθμό σελίδας p και μετατόπιση d'.



Πίν. Σελίδων Τμήματος 0	
Αρ. Σελ.	Αρ. Πλαισίου
0	151F
1	?
2	34FE
3	7E11
4	2345
5	-
...	...

Πίν. Σελίδων Τμήματος 1	
Αρ. Σελ.	Αρ. Πλαισίου
0	-
1	2EE1
2	0B0B
3	0C11
4	?
5	1BA2
...	...

### Ερώτημα Δ [4]

Έστω η παρακάτω ακολουθία αναφοράς μίας διεργασίας:

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

Η διεργασία εκτελείται σε σύστημα που η μνήμη του διαθέτει τέσσερα (4) πλαίσια σελίδων, τα οποία αρχικά είναι κενά. Στον πίνακα που ακολουθεί δώστε την ακολουθία αναφοράς, σημειώνοντας **με μαύρο χρώμα** ανά χρονική στιγμή τις σελίδες που υπάρχουν στον πίνακα σελίδων και σημειώνοντας **με κόκκινο χρώμα** μόνο τους αριθμούς σελίδων στα σημεία στα οποία συμβαίνουν σφάλματα σελίδας για την πολιτική αντικατάστασης σελίδων LRU (Least Recently Used).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0																				
1																				
2																				
3																				

**Καλή Επιτυχία!!!**

### Ημερομηνία Παράδοσης: 13/1/2020

Η παράδοση της άσκησης θα πραγματοποιείται με αποστολή μηνύματος ηλεκτρονικού ταχυδρομείου ΚΑΙ ΣΤΙΣ ΤΡΕΙΣ ακόλουθες διευθύνσεις με ένα μήνυμα (με τρεις παραλήπτες και όχι τρία διακριτά μηνύματα):

[sioutas@ceid.upatras.gr](mailto:sioutas@ceid.upatras.gr), [makri@ceid.upatras.gr](mailto:makri@ceid.upatras.gr), [aristeid@ceid.upatras.gr](mailto:aristeid@ceid.upatras.gr)

Μπορείτε να συντάξετε την αναφορά σας σε όποια μορφή κειμένου επιθυμείτε (word, pdf, κ.λπ.). Στο ηλεκτρονικό μήνυμα που θα αποστείλετε θα έχετε συμπεριλάβει το αρχείο της αναφοράς σας καθώς και τα αρχεία των προγραμμάτων C Unix/Linux.



## Παράρτημα

**Υποδείξεις: Συμβουλευτείτε τους** πηγαίους κώδικες των παραδειγμάτων, που βρίσκονται στο δικτυακό τόπο του μαθήματος στο eclass.

Η μεταγλώττιση ενός προγράμματος στο Linux/Unix γίνεται με τη χρήση της εντολής:

`gcc -pthread -o <όνομα εκτελέσιμου> <όνομα αρχείου που περιέχει τον πηγαίο κώδικα>.`

Η εντολή αυτή θα πρέπει να δοθεί από τερματικό ενώ ο τρέχων κατάλογος εργασίας θα πρέπει να είναι αυτός στον οποίο βρίσκεται αποθηκευμένος ο πηγαίος κώδικας του προγράμματος. (Η εντολή με την οποία μπορούμε να βρούμε τον τρέχοντα κατάλογο είναι η `pwd` ενώ μετακινούμαστε μεταξύ καταλόγων με την εντολή `cd`. Τα περιεχόμενα των καταλόγων βρίσκονται με την εντολή `ls`. Περισσότερες πληροφορίες για τη λειτουργικότητα και τις παραμέτρους των εντολών (αλλά και των κλήσεων συστήματος) μπορούν να βρεθούν μέσω της βοήθειας του Linux που δίνεται αν πληκτρολογήσουμε `man <όνομα εντολής>`).

Η εκτέλεση ενός προγράμματος γίνεται πληκτρολογώντας:

`./<όνομα εκτελέσιμου>`

Η εκτέλεση ενός προγράμματος στο παρασκήνιο γίνεται πληκτρολογώντας:

`./<όνομα εκτελέσιμου>&`