



# Λειτουργικά Συστήματα

2018 - 2019

## 2η Εργαστηριακή Άσκηση

### Μέρος 1 [70 μονάδες]

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

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int pid;
    int x,y;

    x = 10;
    y = 10;

    pid = fork();

    if (pid != 0)
    {
        x++;
        y--;
    }

    printf("x = %i y = %i\n",x,y);

    pid = fork();

    if (pid != 0)
    {
        x++;
        y--;
    }

    printf("x = %i y = %i\n",x,y);

    return (0);
}
```

**Ερώτημα Β [5]:** Δημιουργείστε ένα πρόγραμμα στο οποίο μία διεργασία στο Linux/Unix παράγει άλλες 4 θυγατρικές της.



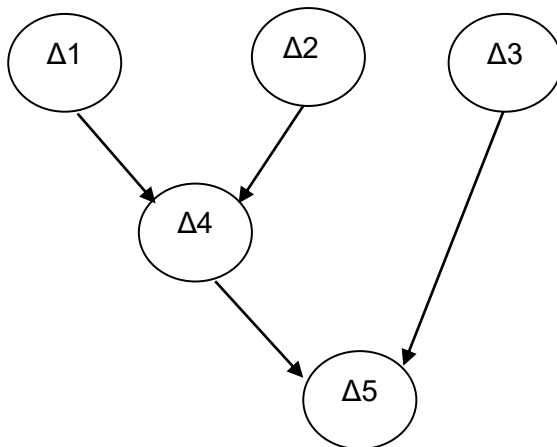
**Ερώτημα Γ [5]:** Να δημιουργήσετε ένα πρόγραμμα στο οποίο να δημιουργούνται 5 διεργασίες (αλυσίδα – κάθε μία να είναι παιδί της άλλης). Κάθε διεργασία να τυπώνει, το id του πατέρα της, το id της και το id του μοναδικού παιδιού που δημιουργεί.

**Ερώτημα Δ [15]:** Να δημιουργήσετε πρόγραμμα στο οποίο:

1. Θα ορίσετε μια συνάρτηση `nothing()` η οποία θα ορίζει μια μεταβλητή `int x=0` και θα κάνει την πράξη `x=x+1`. Αυτή η συνάρτηση δεν κάνει τίποτα χρήσιμο, αλλά απλώς υπάρχει για να μας βοηθήσει στη μέτρηση.
2. Μέσα στη `main()` θα εκτελεί μια φορά τη συνάρτηση `time()` με τις κατάλληλες παραμέτρους, θα αποθηκεύεται ο αριθμός των δευτερολέπτων στη μεταβλητή `start` και αμέσως μετά θα εκτυπώνεται το μήνυμα “Αρχική τιμή δευτερολέπτων” ακολουθούμενο από τη `start`.
3. Στη συνέχεια θα υπάρχει ένας βρόχος `while()` ο οποίος θα δημιουργεί 100 διεργασίες (δοκιμάστε και για 5000, 10000) με τη `fork()` οι οποίες όλες θα εκτελούν τη `nothing()`. Προσοχή, ο πατέρας θα δημιουργήσει όλες τις διεργασίες και όχι η κάθε διεργασία θα δημιουργεί άλλη διεργασία.
4. Μόλις δημιουργηθούν οι 100 διεργασίες και μόνο τότε θα εκτελεί ο πατέρας 100 φορές τη `waitpid()` ώστε να περιμένει την επιτυχή ολοκλήρωση όλων των θυγατρικών.
5. Στη συνέχεια θα καλείται η `time()`, θα αποθηκεύεται ο αριθμός των δευτερολέπτων στη μεταβλητή `end` και αμέσως μετά θα εκτυπώνεται το μήνυμα “Τελική τιμή δευτερολέπτων” ακολουθούμενη από την `end`. Θα γίνεται η πράξη `end-start`, θα εκτυπώνεται το αποτέλεσμα ενώ θα εκτυπώνεται και το αποτέλεσμα « $(end-start)/100$ » για να εμφανιστεί ο μέσος χρόνος δημιουργίας, εκτέλεσης και τερματισμού των 100 διεργασιών.

Πόσος είναι ο συνολικός χρόνος και ο μέσος χρόνος δημιουργίας εκτέλεσης και τερματισμού των 100 διεργασιών στο σύστημα σας (μη ξεχάσετε τη μονάδα μέτρησης);

**Ερώτημα Ε [20]:** Να γράψετε πρόγραμμα συγχρονισμού για τον παρακάτω γράφο προτεραιότητας κάνοντας χρήση δύο μόνο 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.



**Ερώτημα ΣΤ [20]:** Δημιουργείστε n-διεργασίες (θυγατρικές) κάθε μία από τις οποίες να εκτελεί το παρακάτω κομμάτι ψευδο-κώδικα:

```
Char *p;  
t = *p;  
    CONCATENATE TO p A CHILD[i]--text;  
    sleep (nSeconds[i]);  
*p = t;
```



Δηλαδή, κάθε νέα (θυγατρική) διεργασία περιμένει ένα τυχαίο αριθμό δευτερολέπτων και στη συνέχεια αυξάνει το περιεχόμενο της κοινής *διαμοιραζόμενης μεταβλητή*\* $p$  προσαρτώντας σχετικό κείμενο. Να χρησιμοποιήσετε σημαφόρους έτσι ώστε να προστατέψετε τη κρίσιμη περιοχή κάθε διεργασίας, ώστε τελικά η μεταβλητή \* $p$  να έχει προσαρμοσμένα όλα τα κομμάτια κειμένων με οποιαδήποτε σειρά.

Τι αλλαγές θα πρέπει να κάνετε στον κώδικα, ώστε τα παραπάνω κομμάτια να έχουν συγκεκριμένη σειρά ώστε να βγαίνει νόημα κατά την ανάγνωση?

## Μέρος 2 [30 μονάδες]

### Ερώτημα Α [5]

Δίνεται ο παρακάτω πίνακας τμημάτων μιας διεργασίας:

Τμήμα	Βάση	Όριο
0	1024	1024
1	9896	128
2	512	128
3	3912	1400
4	1536	1024
5	5688	2000

[i] Μπορεί ο παραπάνω πίνακας να είναι όντως πίνακας τμημάτων μιας διεργασίας; Δικαιολογήστε την απάντησή σας. [2]

[ii] Θεωρώντας ότι ο παραπάνω πίνακας είναι σωστός, βρείτε τη φυσική διεύθυνση που αντιστοιχεί σε καθεμία από τις παρακάτω λογικές διευθύνσεις. (Κάθε λογική διεύθυνση είναι της μορφής  $(x, y)$ , όπου  $x$  = τμήμα και  $y$  = λέξη). [3]

(0, 256)	
(1, 40)	
(2, 512)	
(3, 1000)	
(5, 1536)	

### Ερώτημα Β [12]

Θεωρήστε πως διαθέτουμε ένα υπολογιστικό σύστημα με τα παρακάτω χαρακτηριστικά:

- Μια σελίδα στην λογική μνήμη έχει μέγεθος 8 Kbytes.
- Ο χώρος μνήμης μιας διεργασίας είναι 64 Kbytes.
- Η φυσική μνήμη αποτελείται από 32 πλαίσια σελίδων, καθένα μεγέθους 8 Kbytes.

Στο παραπάνω υπολογιστικό σύστημα εκτελούνται 3 διεργασίες. Στο σχήμα που ακολουθεί δίνονται οι σελίδες ή/και τα πλαίσια σελίδων των 3 διεργασιών και οι πίνακες σελίδων τους. Συμπληρώστε στο σχήμα τα στοιχεία που λείπουν με **κόκκινο χρώμα γραμματοσειράς** ως εξής:

[i] Για την Διεργασία 1, συμπληρώστε στην Φυσική μνήμη στα αντίστοιχα πλαίσια σελίδων ένα ζεύγος της μορφής «Δ1, Σ $N$ », όπου  $N$  ένας αριθμός που αναπαριστά την σελίδα στην Λογική μνήμη της Διεργασίας 1. [4]



[ii] Για την Διεργασία 2, συμπληρώστε τον Πίνακα σελίδων, χρησιμοποιώντας την πληροφορία των ζευγών «Δ2, ΣN» από την Φυσική μνήμη, όπου N ένας αριθμός που αναπαριστά την σελίδα στην Λογική μνήμη της Διεργασίας 2. **[4]**

[iii] Για την Διεργασία 3, συμπληρώστε στον Πίνακα σελίδων και στην Φυσική μνήμη τα στοιχεία που λείπουν, χρησιμοποιώντας την κατάλληλη πληροφορία, όπως κάνατε στα παραπάνω 2 υποερωτήματα. **[4]**

Λογική μνήμη Διεργασίας 1	Πίνακας σελίδων Διεργασίας 1	Φυσική μνήμη	Πλαίσιο Σελίδας
Σελίδα 0	4		0
Σελίδα 1	7		1
Σελίδα 2	15		2
Σελίδα 3	12		3
Σελίδα 4	2		4
Σελίδα 5	10		5
Σελίδα 6	13		6
Σελίδα 7	18		7
		Δ2, Σ0	8
			9
			10
		Δ2, Σ3	11
			12
			13
		Δ2, Σ2	14
			15
			16
		Δ2, Σ7	17
			18
		Δ2, Σ6	19
		Δ2, Σ1	20
		Δ3, Σ1	21
		Δ2, Σ5	22
		Δ2, Σ4	23
		Δ3, Σ4	24
			25
			26
			27
			28
		Δ3, Σ6	29
		Δ3, Σ2	30
			31

Λογική μνήμη Διεργασίας 2			
Σελίδα 0			
Σελίδα 1			
Σελίδα 2			
Σελίδα 3			
Σελίδα 4			
Σελίδα 5			
Σελίδα 6			
Σελίδα 7			

Λογική μνήμη Διεργασίας 3			
Σελίδα 0	16		
Σελίδα 1			
Σελίδα 2			
Σελίδα 3	27		
Σελίδα 4			
Σελίδα 5	26		
Σελίδα 6			
Σελίδα 7	25		

### Ερώτημα Γ [5]

Για το παραπάνω σύστημα συμπληρώστε τους πίνακες που ακολουθούν, θεωρώντας πως η Διεργασία 1 προσπελαύνει την λογική θέση μνήμης 30784, με την διεύθυνση μνήμης να είναι εκφρασμένη στο δεκαδικό σύστημα. Στον πρώτο πίνακα υποδείξτε **με πράσινο χρώμα γραμματοσειράς** τα bit που χρησιμοποιούνται για την διευθυνσιοδότηση στον πίνακα σελίδων και **με κόκκινο χρώμα γραμματοσειράς** τα bit που χρησιμοποιούνται για την εύρεση της συγκεκριμένης θέσης μνήμης μέσα σε ένα πλαίσιο σελίδας. Στον δεύτερο πίνακα συμπληρώστε την διεύθυνση της φυσικής μνήμης που αντιστοιχεί στην παραπάνω λογική διεύθυνση. Στους πίνακες χρησιμοποιήστε μόνο όσα bit είναι απαραίτητα για το υπολογιστικό σύστημα που περιγράψαμε νωρίτερα.



## Λογική Διεύθυνση Μνήμης Διεργασίας 1

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

## Φυσική Διεύθυνση Μνήμης

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

**Ερώτημα Δ [8]**

Έστω η παρακάτω ακολουθία αναφοράς μίας διεργασίας:

3 5 8 1 8 7 5 1 4 2 8 2 7 3 6 4 6 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: 7/1/2019**

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>&`