Λειτουργικά Συστήματα 2η Άσκηση

Χειμερινό Εξάμηνο 2019-2020

Σταθόπουλος Ιωάννης | 1043823 (6353) 6ο
Τσαρκάτογλου Τριαντάφυλλος | 1043836 (6367) 6ο
Σπανού Άννα-Μαρία | 1041884 (236207) 6ο

Μέρος 1 [70 μονάδες]

Ερώτημα Α [5]:

```c
#include <stdio.h>
#include <stdlib.h>
int main()
{
int pid1;
int pid2; //Orismos metavlitwn
pid1 = fork(); //fork sto pid1 ara ena paidi enas pateras
if (pid1 < 0)
 printf("Could not create any child\n"); //error check
else
{
 pid2 = fork(); //allios proxora se fork gia to pid2 ara ena paidi enas pateras
 if (pid2 < 0)
 printf("Could not create any child\n"); //error check
 else
 if ( (pid1 < 0) && (pid2 < 0) kill(pid1,9); //error check
}
sleep(20); //sleep gia 20 defterolepta
return (0);
}
```

ι) Πόσες διεργασίες υπάρχουν σε κατάσταση sleeping 10 δευτερόλεπτα μετά την έναρξή του;

Έπειτα απο 10 δευτερόλεπτα σε κατάσταση sleeping βρίσκονται πέντε διεργασίες.

ii) Τροποποιήστε κατάλληλα το κώδικα έτσι ώστε κάθε διεργασία να τυπώνει το id της και το id του γονέα της πριν βρεθεί σε κατάσταση sleeping.

2erA.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <sys/wait.h>
#include <unistd.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)){
printf("to lathos sto else if , %i \n",getpid());
printf("to lathos sto else if , %i \n",getppid());
kill(pid1,9);
printf("meta to kil paidia, %i \n",getpid());
printf("meta to kil goneis, %i \n",getpid());
}
printf("Child ID:, %i \n",getpid());
printf("Parent ID:, %i \n",getppid());
sleep(20);

return (0);
}

```

Έξοδος:

```
Child ID:, 6923
Parent ID:, 6918
Child ID:, 6924
Parent ID:, 6923
Child ID:, 6926
Parent ID:, 6924
Child ID:, 6925
Parent ID:, 6923
```

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.

Το πρόγραμμα περιέχεται στο αρχείο **2erG.c** και δεν συμπεριλήφθηκε εδώ στην αναφορά λόγο όγκου.

Ερώτημα Δ [20]: (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 Δi εκτελείται πριν από τη διαδικασία Δj» σημαίνει ότι η εκτέλεση της Δi πρέπει να ολοκληρωθεί πριν αρχίσει η εκτέλεση της Δj. Αναλόγως η έκφραση «η διαδικασία Δi εκτελείται μετά από τη διαδικασία Δj» σημαίνει ότι η εκτέλεση της Δi μπορεί να αρχίσει αφού ολοκληρωθεί η εκτέλεση της Δj.

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.

Απάντηση:

Χρησιμοποιούμε σημαφόρων Σij για τον έλεγχο των προτεραιοτήτων Δi → Δj (χρησιμοποιούμε μία σημαφόρο Σij – με αρχική τιμή ‘0’ – για τον έλεγχο κάθε σχέσης προτεραιότητας Δi → Δj):

```
var Σ12, Σ13, Σ24, Σ25, Σ35, Σ36, Σ46: semaphores;
Σ12:= Σ13:= Σ24:= Σ25:= Σ35:= Σ36:= Σ46:= 0;

cobegin

begin Δ1; up(Σ12); up(Σ13); end;
begin down(Σ12); Δ2; up(Σ24); up(Σ25); end;
begin down(Σ13); Δ3; up(Σ35); up(Σ36);end;
begin down(Σ24); Δ4; up(Σ46); end;
begin down(Σ25); down(Σ35); Δ5; end;
begin down(Σ46); down(Σ36); Δ6; end;
coend
```

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 3950016 bytes. Αν θεωρήσουμε ότι ή 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 κοκ). 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_{16} (ii) $0001E800_{16}$

Απάντηση:

(α) Σύμφωνα με την εκφώνηση:

- Τα 18 bits αφορούν τον αριθμό σελίδας
- Τα υπόλοιπα 14 bits αφορούν τη μετατόπιση (γιατί 32-18 =14 bits)

Ακόμα: $39500_{16}=234752_{10}$ και $2^{14}=16384$

Η εν λόγω διεργασία είναι ολόκληρη φορτωμένη στη μνήμη.
Καταλαμβάνει: 234752/16384= 15 σελίδες και η εσωτερική κλασματοποίηση που προκαλεί είναι ίση με: 15* 16384- 234752= 11002

(β) Υποθέτω ότι η εν λόγω διεργασία, κατά το τρέχον χρονικό διάστημα έχει φορτωμένες στη μνήμη μόνο τις τελευταίες πέντε σελίδες της στα ακόλουθα πλαίσια:

Αριθμός σελίδας	Αριθμός πλαισίου
11	16
12	225
13	170
14	35
15	51

i) Αρχικά μετατρέπω την λογική διεύθυνση 00031958_{16} στο δυαδικό σύστημα:
00031958_{16}= 0000 0000 0000 0011 0001 1001 0101 1000_2

Διαχωρίζω τον αριθμό σελίδας και τη μετατόπιση σύμφωνα με τα παραπάνω:

- Αριθμός σελίδας: 0000 0000 0000 0011 00_2= C_{16}= 12_{10}
- Μετατόπιση: 01 1001 0101 1000_2= 1958_{16}

Με βάση τον πίνακα σελίδων, η λογική σελίδα 12 αντιστοιχεί στη φυσική σελίδα(πλαίσιο): $225_{10.}$
Άρα η τελική φυσική διεύθυνση είναι η: 0000 0000 1110 0001 1001 0101 1000_2= $E11958_{16}$

ii) Αρχικά μετατρέπω την λογική διεύθυνση $0001E800_{16}$ στο δυαδικό σύστημα:
$0001E800_{16}$= 0000 0000 0000 0001 1110 1000 0000 0000_2

Διαχωρίζω τον αριθμό σελίδας και τη μετατόπιση:

- Αριθμός σελίδας: 0000 0000 0000 0001 11_2= 19_{16}= 7_{10}
- Μετατόπιση: 10 1000 0000 0000_2

Παρατηρώ πως ο αριθμός σελίδας είναι 7. Όμως το τρέχον χρονικό διάστημα η σελίδα αυτή δεν είναι φορτωμένη στη μνήμη(page fault).

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	1100
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) $0B00042A_{16}$ (ii) $02000B6D_{16}$

Απάντηση:

(α) 16MB= $2^4 * 2^{20}$ bytes= 2^{24} bytes

Επειδή το σύστημα υποστηρίζει τμηματοποίηση, η μετατόπιση θα είναι 24 bits και ο αριθμός τμήματος 8 bits(32-24= 8 bits).

Άρα ο μέγιστος υποστηριζόμενος αριθμός τημάτων είναι 2^8= 256 τμήματα.

(β) i) Αρχικά μετατρέπω τη λογική διεύθυνση $0B00042A_{16}$ στο δυαδικό σύστημα:

$0B00042A_{16}$= 0000 1011 0000 0000 0000 0100 0010 1010_2

Διαχωρίζω τον αριθμό τμήματος και τη μετατόπιση:

- Αριθμός τμήματος: 0000 1011_2= 11_{10}= B_{16}
- Μετατόπιση: 0000 0000 0000 0100 0010 1010_2= 1066_{10}= $42A_{16}$

Προσπελαύνω το τμήμα 11 και βρίσκω τη διεύθυνση βάσης 9050_{10} (10001101011010_2), με μήκος τμήματος 1230. Για να έχω έγκυρη διεύθυνση, ελέγχω αν η μετατόπιση είναι <= απο το μήκος τμήματος(1066_{10}<= 1230_{10}), που ισχύει.

Προσθέτοντας τη βάση και τη μετατόπιση προκύπτει η αντίστοιχη φυσική διεύθυνση, που ειναι: 10011110000100_2= 2784_{16}

ii) Μετατρέπω τη λογική διεύθυνση $02000B6D_{16}$ δυαδικό σύστημα:

$02000B6D_{16}$= 0000 0010 0000 0000 0000 1011 0110 1101_2

Διαχωρίζω τον αριθμό τμήματος και τη μετατόπιση:

- Αριθμός τμήματος: = 0000 0010_2= 2_{10}
- Μετατόπιση: 0000 0000 0000 1011 0110 1101_2= 2925_{10}= $B6D_{16}$

Προσπελαύνω το τμήμα 2 και βρίσκω τη διεύθυνση βάσης 10310 με μήκος τμήματος 1290. Για να έχω έγκυρη διεύθυνση, ελέγχω αν η μετατόπιση είναι <= απο το μήκος τμήματος(2925_{10}<= 129_{10}), που δεν ισχύει. Αρα η διεύθυνση δεν είναι έγκυρη.

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.

(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 010004CF16 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 $010009FF_{16}$ προκαλεί σφάλμα σελίδας (page fault), και
- η λογική διεύθυνση $000003F0_{16}$ αντιστοιχεί στη φυσική διεύθυνση $E0E1F0_{16}$

Απάντηση:

Από το ερώτημα Β, η ιδεατή/λογική διεύθυνση είναι των 32 bits και μέγιστο υποστηριζόμενο μέγεθος ενός τμήματος είναι 2^{24} bytes.

(α) μέγεθος σελίδας= 512 bytes= 2^9 bytes

Άρα 9 bits αποτελούν τη μετατόπιση, 15 bits είναι ο αριθμός της σελίδας και 8 bits ο αριθμός τμήματος.

Αριθμός τμήματος	Μετατόπιση	
8 bits	Αριθμός σελίδας	Μετ/ση σελίδας
	15 bits	9 bits

(β) Μια διεργασία στο νέο σύστημα μνήμης αποτελείται κατα μέγιστο από:
$2^8 * 2^{16} = 2^{24}$ σελίδες.

(γ) i) Αρχικά μετατρέπω τη λογική διεύθυνση $010004CF_{16}$ στο δυαδικό σύστημα:

$010004CF_{16}$= 0000 0001 0000 0000 0000 0100 1100 1111_2

Διαχωρίζω τα bits:

- Αριθμός τμήματος: 0000 $0001_2 = 1_{10}$
- Αριθμός σελίδας: 0000 0000 0000 $010_2 = 2_{10}$
- Μετατόπιση σελίδας: 01100 $1111_2 = 207_{10} = CF_{10}$

Προσπελαύνω το τμήμα 1, τη σελίδα 2 με αριθμό πλαισίου $0B0B_{16}$

Άρα η νέα φυσική διεύθυνση είναι: 0000 1011 0000 1011 0000 1100 $1111_2 = 0B0BCF_{16}$

ii)

Ως δεδομένο θεωρώ ότι η λογική διεύθυνση $010009FF_{16}$ προκαλεί σφάλμα σελίδας(page fault), άρα θα βάλουμε "-" ,γιατί δεν είναι φορτωμένη στη μνήμη.

Μετατρέπω τη διεύθυνση 000003F0 στο δυαδικό και διαχωρίζω τα bits:

$000003F0_{16}$= 0000 0000 0000 0000 0000 0011 1111 0000_2

- Αριθμός τμήματος: 0000 $0000_2 = 0_{10}$
- Αριθμός σελίδας: 0000 0000 0000 $001_2 = 1_{10}$

Άρα προσπελαύνω το τμήμα 0, τη σελίδα 1, όπου μέσα έχει ερωτηματικό.

Η φυσική διεύθυνση στην οποία καταλήγω είναι η E0E1F016= 1110 0000 1110 0001 1111 0000_2= 0000 0000 1110 0000 1110 0001 1111 0000 (32 bits διεύθυνση για να συγκρίνω με την δοσμένη)
Επομένως στον πίνακα σελίδων του τμήματος 0, με σελίδα 1, στον αριθμό πλαισίου θα βάλω το 7070.

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)

Απάντηση:

ακολουθία αναφορας	3	5	8	1	8	7	5	1	8	2	4	2	7	3	6	7	5	3	7
0	3	3	3	3	3	7	7	7	7	7	7	7	7	7	6	6	6	6	6
1		5	5	5	5	5	5	5	5	2	2	2	2	2	2	7	7	7	7
2			8	8	8	8	8	8	8	8	4	4	4	4	4	4	5	5	5
3				1	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3
ΣΦΑΛΜΑ	X	X	X	X		X				X	X			X	X	X	X		

Με την πολιτική αντικατάστασης LRU αντικαθίσταται η σελίδα που δεν έχει χρησιμοποιηθεί για το μεγαλύτερο διάστημα.
Προκύπτουν 11 σφάλματα.