

ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

2019-2020

2η εργαστηριακή Άσκηση

Μάριος-Αλέξανδρος Μορφόπουλος 1058102

Παναγιώτης Χριστόπουλος 1054409

Χρήστος Στεμιτσιώτης 1054375

Κωνσταντίνος Μελισσουργός 1054318

Μέρος 1

Ερώτημα Α:

```
panos@panos-D15D:~/Desktop/Λειτουργικα2$ ./er1
To pid μου είναι 14157 ο πατέρας μου είναι 14133
To pid μου είναι 14159 ο πατέρας μου είναι 14157
To pid μου είναι 14158 ο πατέρας μου είναι 14157
To pid μου είναι 14160 ο πατέρας μου είναι 14158
```

Ο κώδικας που μας δόθηκε εκτελείται ως εξής: Δηλώνονται δύο διεργασίες με pid_1 και pid_2 και αρχικά επιχειρείται `fork()` της p_1 . Αν το `fork` επιτύχει το πρόγραμμα επιχειρεί να δημιουργήσει μια διεργασία-αντίγραφο της p_2 ξανά με το `fork()`. Γίνεται ξανά έλεγχος επιτυχίας για το 2 και στη συνέχεια γίνεται για την επιτυχία και των 2 `fork`. Ο κώδικας μπαίνει σε κατάσταση `sleep` για 20 - δευτερόλεπτα και στη συνέχεια σταματάει να εκτελείται επιστρέφοντας 0. Συνεπώς, όπως παρατηρείται μετά την εκτέλεση του κατάλληλου κώδικα, μετά από 10 λεπτά υπάρχουν 4 διεργασίες σε κατάσταση `sleeping` (pid_1 , pid_2 , $forkedpid_1$, $forkedpid_2$)

ΕΡΩΤΗΜΑ Aii – B – Γ – Δ :

Βρίσκονται συνημμένα μαζί με την αναφορά.

Μέρος 2

Ερώτημα Α:

α) Από την εκφώνηση παρατηρούμε ότι έχουμε λογικές διευθύνσεις των 32 bits όπου τα πρώτα (πιο σημαντικά) 18 bits αναπαριστούν τον αριθμό σελίδα ($page = 18 \text{ bits}$) οπότε τα υπόλοιπα 14 bits είναι το `offset`.

Το `offset` έχει διπλό ρόλο:

- A) Δείχνει τη μετατόπιση της λογικής διεύθυνσης από την αρχή της σελίδας που βρίσκεται
- B) Προσδιορίζει ταυτόχρονα και το μέγεθος της σελίδας

Συνεπώς το μέγεθος της σελίδας είναι $2^{14} \text{ bytes} = 16384$

$39600_{16} = 234752 \text{ bytes} = \text{μέγεθος διεργασίας} = \text{μέγεθος πίνακα σελίδων}$

Για να βρούμε πόσα πλαίσια σελίδων καταλαμβάνει (πλήθος σελίδων της διεργασίας) διαιρούμε τα δύο μεγέθη : $234752/16384 = 14,32$. Άρα ο πίνακας σελίδων καταλαμβάνει 15 σελίδες. Οι 14 σελίδες της διεργασίας γεμίζουν πλήρως ενώ η 15η σελίδα γεμίζει μόνο κατά 0,32 άρα η εσωτερική κλασματοποίηση δηλαδή ο κενός χώρος εντός της σελίδας είναι 0,68 του μεγέθους της ή $0,68*16384 = 11141,12$ bits.

β) Σύμφωνα με την εκφώνηση έχουμε την ακόλουθη αντιστοίχιση:

Σελίδα	Φυσικό Πλαίσιο
0	x
1	x
2	x
...	x
10	16
11	225
12	170
13	35
14	51

Το 00031958_{16} σε δυαδική μορφή είναι:

0000 0000 0000 0011 0001 1001 0101 1000

Από την εκφώνηση γνωρίζουμε ότι τα πρώτα 18 bits είναι το page. Άρα $1100_2 = 12$ η σελίδα. Συνεπώς βρισκόμαστε στο φυσικό πλαίσιο 170

Από την εκφώνηση γνωρίζουμε ότι τα υπόλοιπα 14 bits είναι το offset. Άρα $01100101011000_2 = 6488$

Επομένως $170*16384$ (μέγεθος της σελίδας στο φυσικό πλαίσιο) + $6488 = 2791768_{10} = 2A9958_{16}$ (φυσική διεύθυνση)

β) Το $001E800_{16}$ σε δυαδική μορφή είναι:

0000 0000 0000 0001 1110 1000 0000 0000

Όπως και πριν, τα πρώτα 16 bits είναι το page.
Άρα $0111_2 = 7$ η σελίδα

14 bit offset

Όμως, η σελίδα 7 δεν απεικονίζεται σε φυσικό πλαίσιο μνήμης, άρα έχουμε page fault.

Ερώτημα Β:

α) Από την εκφώνηση παρατηρούμε ότι κάθε λογική (εικονική) διεύθυνση αποτελείται από 32 bit, συνεπώς το μέγεθος του λογικού (εικονικού) χώρου διευθύνσεων είναι 2^{32} bytes. Επίσης, ξέρουμε ότι το μέγιστο υποστηριζόμενο μέγεθος κάθε τμήματος είναι 16 MegaBytes, οπότε προκύπτει ότι $16 * 2^{20} = 2^4 * 2^{20} = 2^{24}$ bytes. Για να βρούμε το μέγιστο υποστηριζόμενο αριθμό τμημάτων για μία διεργασία, θα διαιρέσουμε τα δύο παραπάνω μεγέθη, οπότε προκύπτει $2^{32} / 2^{24} = 2^8 = 256$ τμήματα/

βi) Το $0B00042A_{16}$ σε δυαδική μορφή είναι:

0000 1011 0000 0000 0000 0100 0010 1010

Γενικά, γνωρίζουμε ότι στην τμηματοποίηση η μορφή των λογικών διευθύνσεων είναι (s,d) όπου s=segment (τμήμα) και d=offset (μετατόπιση). Από πριν ξέρουμε ότι τα πρώτα 8 bit είναι το segment, οπότε $1011_2 = 11$ τμήμα, ενώ τα τελευταία 24 bit είναι το offset, οπότε $10000101010_2 = 1066$

Σύμφωνα με τον πίνακα της Άσκησης:

Αριθμός Τμήματος	Διεύθυνση Βάσης	Μήκος Τμήματος
0	1650	1110
1	3200	2380
2	10310	1290
...
10	5950	2265
11	9050	1230
12	12270	5535
...

Για να μετατρέψουμε τη λογική (εικονική) διεύθυνση σε φυσική εξετάζουμε πρώτα αν το $d < \text{limit}$ για το συγκεκριμένο τμήμα που βρισκόμαστε. Επειδή $d=1066 < \text{limit}=1230$, η λογική διεύθυνση είναι έγκυρη δηλαδή έχει φορτωθεί σε κάποιο τμήμα (segment). Επομένως, η φυσική διεύθυνση είναι $\text{base address} + \text{offset} = 9050 + 1066 = 10116$ και η μετατροπή της σε δεκαεξαδική μορφή είναι 2784_{16} .

βii) Το $0200B6D_{16}$ σε δυαδική μορφή είναι:

000000100000000000000101101101101

Πάλι απο πριν τα πρώτα 8 bit είναι το segment (τμήμα) οπότε $0010_2 = 2$ τμήμα, ενώ τα τελευταία 24 bit είναι το offset, οπότε $101101101101_2 = 2926$. Για να μετατρέψουμε την εικονική διεύθυνση σε φυσική εξετάζουμε πρώτα αν το $d < \text{limit}$ για το συγκεκριμένο πλαίσιο. Επειδή $d=2915 > \text{limit}=1290$ (από τον πίνακα της εκφώνησης), η λογική διεύθυνση δεν απεικονίζεται σε φυσική διεύθυνση, συνεπώς έχουμε σφάλμα τμήματος (segment fault).

Ερώτημα Γ:

α) Γενικά γνωρίζουμε ότι στην σελιδοποιημένη τμηματοποίηση (paged segmentation) η μορφή των λογικών διευθύνσεων είναι (s,p,d) όπου s είναι το τμήμα(segment), p είναι η σελίδα (page) και d το offset (μετατόπιση). Από πριν (θέμα Β) γνωρίζουμε ότι έχουμε 256 τμήματα, άρα τα bits για το segment είναι τα πρώτα 8 bits. Από την εκφώνηση γνωρίζουμε ότι το μέγεθος σελίδας είναι 512 bytes (2^9), άρα το offset είναι τα τελευταία 9 bits. Άρα τα bits που απομένουν για το πλήθος σελίδων είναι 15 ($32-17=15$) δηλαδή υπάρχουν 2^{15} σελίδες σε κάθε πίνακα σελίδων. Συνεπώς, η μορφή των λογικών διευθύνσεων είναι (s,p,d) \Rightarrow (8,15,9) ή

S	P	D
8	5	19

β) Γενικά, γνωρίζουμε ότι το κάθε τμήμα (segment) έχει έναν πίνακα σελίδων και ο κάθε πίνακας σελίδων έχει 2^{15} σελίδες. Συνεπώς, ο μέγιστος αριθμός σελίδων μίας διεργασίας είναι:

$$2^8 \text{ τμήματα (segment)} * 2^{15} \text{ σελίδες (pages)/segment} = 2^{23} \text{ σελίδες}$$

γi) Το 010004C₁₆ σε δυαδική μορφή είναι:

0000 0001 0000 0000 0000 0100 1100 1111

segment

page

offset

Παρατηρούμε ότι τα πρώτα 8 bit που είναι το segment είναι το $0001_2 = 1$, άρα είμαστε στο τμήμα 1, επίσης από το page παρατηρούμε ότι είναι $0010_2 = 2$, άρα είμαστε στη σελίδα 2, που από τον πίνακα της εκφώνησης έχει φορτωθεί στο φυσικό πλαίσιο OBOB₁₆=2827.

Πίνακας Σελίδων Τμήματος 0	
Αριθμός Σελίδων	Αριθμός Πλαισίου
0	151F
1	?
2	34FE
3	7E11
4	2345
5	-
...	...

Πίνακας Σελίδων Τμήματος 1	
Αριθμός Σελίδων	Αριθμός Πλαισίου
0	-
1	2EE1
2	0B0B
3	0C11
4	?
5	18A2
...	...

Αυτό το φυσικό πλαίσιο αρχίζει στη διεύθυνση $2827 * 512 = 1447424$. Σε αυτή την αρχική διεύθυνση προσθέτουμε το offset ($1101111_2 = 207$) και προκύπτει η τελική φυσική διεύθυνση που είναι:

$2827 * 512 + 207 = 1447631$ και σε δεκαεξαδική μορφή είναι $161CF_{16}$

γii) Το $01009FF_{16}$ σε δυαδική μορφή είναι:

0000 0001 0000 0000 0000 1001 11111111

segment

page

offset

Παρατηρούμε ότι τα πρώτα 8 bit που είναι το segment είναι το $0001_2 = 1$, άρα είμαστε στο τμήμα 1, επίσης από το page παρατηρούμε ότι είναι $0100_2 = 4$, άρα είμαστε στη σελίδα 4. Επίσης το offset προκύπτει $11111111_2 = 511$. Η σελίδα 5 δεν έχει φορτωθεί σε φυσικό πλαίσιο μνήμης. Άρα βάζουμε – στο αντίστοιχο ?

Πίνακας Σελίδων Τμήματος 1	
Αριθμός Σελίδων	Αριθμός Πλαισίου
0	-
1	2EE1
2	0B0B
3	0C11
4	-
5	18A2
...	...

Το $E0E1F0_{16}$ σε δυαδική μορφή είναι:

1100 0000 1110 0001 1111 0000

frame

offset

Παρατηρούμε ότι τα πρώτα 15 bits προσδιορίζουν το φυσικό πλαίσιο μνήμης και τα τελευταία 9 bits προσδιορίζουν το offset. Συνεπώς, προκύπτει ότι ο αριθμός φυσικού πλαισίου είναι: $11100000 - 1110000 = 7070_{16}$, άρα θέτουμε 7070 στο αντίστοιχο ? Για τη σελίδα 1 του πίνακα σελίδων τμήματος 0.

Πίνακας Σελίδων Τμήματος 0	
Αριθμός Σελίδων	Αριθμός Πλαισίου
0	151F
1	7070
2	34FE
3	7E11
4	2345
5	-
...	...

Ερώτημα Δ:

	3	5	8	1	8	7	5	1	8	2	4	2	7	3	6	4	7	5	3	7
0	<u>3</u>	3	3	3	3	<u>7</u>	7	7	7	<u>2</u>	2	2	2	2	2	<u>4</u>	4	4	4	4
1		<u>5</u>	5	5	5	5	5	5	5	5	<u>4</u>	4	4	4	<u>6</u>	6	6	6	<u>3</u>	3
2			<u>8</u>	8	8	8	8	8	8	8	8	8	8	<u>3</u>	3	3	3	<u>5</u>	5	5
3				<u>1</u>	1	1	1	1	1	1	1	1	1	<u>7</u>	7	7	7	7	7	7
	F	F	F	F	H	F	H	H	H	F	F	H	F	F	F	F	H	F	F	H