

Ασκηση 12

HY225-Οργάνωση Υπολογιστων

csd4569-Χρηστος Παπασταμος

Ασκηση 12.4

α) 00000000000000000000

Τα bits της εικονικής διεύθυνσης αποτελούνται από τα 8MS bits που περιγράφουν την σελίδα και τα 12LS τα οποία περιγράφουν το byte μέσα στην σελίδα.

Το index του πίνακα μεταφράσης προέρχεται από τα 8MS bits της διεύθυνσης

TableRegister
Pointer to table

V.Addr	V	Prot	D	R	P. Addr
00	0	---			0
01	0	---			1
02	0	---			2
03	0	---			3
04	0	---			4
...
FF	0	---			F

Από την θέση αυτή του πίνακα διαβάζουμε την αντιστοιχή φυσική διεύθυνση της σελίδας έτσι ώστε να βρούμε τα data στην φυσική μνήμη

Για να συνθέσουμε την φυσική διεύθυνση, κρατάμε τα 12LS bits (τα οποία είναι το offset του byte μέσα στην λέξη) και αντικαθιστούμε τα 8MS με τα αντιστοιχά 4MS bits της φυσικής διεύθυνσης (σύμφωνα με το Page table)

β)

V.Addr	V	Prot	D	R	P.Addr
00	0	---			
01	1	r-x	0	0	4
02	0	r-x			Disk
...	0	r-x			Disk
09	0	r-x			Disk
0A	1	rw-	1	0	F
0B	0	---			
...	0	---			
BF	0	---			
C0	1	rw-	1	0	9
C1	1	r--	0	0	0
C2	0	rw-			Disk
C3	0	rw-			Disk
C4	0	---			
...	0	---			
FD	0	---			
FE	1	rw-	1	0	D
FF	1	rw-	0	0	1

γ)

01038 (fetch)	Fetch απο την φυσικη διευθυνση 4038
0B0F4 (read)	Read απο unallocated εικονικη σελιδα (τερματισμος προγραμματος)
C001C (write)	Write στην φυσικη διευθυνση 901C
0292C (fetch)	Fetch απο σελιδα που βρισκεται στον δισκο (αναμονη για μεταφορα στην φ. μνημη)
00000 (read)	Read απο την παρανομη εικονικη διευθυνση 00000 (τερματισμος προγραμματος)
99F88 (read)	Read απο unallocated εικονικη σελιδα (τερματισμος προγραμματος)
FE5D8 (write)	Write στην φυσικη διευθυνση D5D8
FF100 (fetch)	Fetch απο διευθυνση χωρις δικαιομα execute (τερματισμος προγραμματος)
C20CC (write)	Write σε σελιδα που βρισκεται στον δισκο (αναμονη για μεταφορα στην φ. μνημη)
CD0CC (write)	Write σε unallocated εικονικη σελιδα (τερματισμος προγραμματος)
C0444 (read)	Read απο την φυσικη διευθυνση 9444
01FF4 (fetch)	Fetch απο την φυσικη διευθυνση 4FF4
C1FFC (write)	Write σε διευθυνση χωρις δικαιομα write (τερματισμος προγραμματος)
008E4 (write)	Read απο την παρανομη εικονικη διευθυνση 008E4 (τερματισμος προγραμματος)
C7700 (read)	Read απο unallocated εικονικη σελιδα (τερματισμος προγραμματος)
01E40 (write)	Write σε διευθυνση χωρις δικαιομα write (τερματισμος προγραμματος)

Ασκηση 12.5

LEVEL 1 TABLE

Table 0

αβγ)

L1 V. Address	V	L2 V. Address	L2 V. Addr	V	Prot	D	R	P.Addr
0	1	Table 0	0	0	---			
1	0	-	1	1	r-x	0	0	4
2	0	-	2	0	r-x			Disk
3	0	-	3	0	r-x			Disk
4	0	-	4	0	r-x			Disk
5	0	-	5	0	r-x			Disk
6	0	-	6	0	r-x			Disk
7	0	-	7	0	r-x			Disk
8	0	-	8	0	r-x			Disk
9	0	-	9	0	r-x			Disk
A	0	-	A	1	rw-	1	0	F
B	0	-	B	0	---			
C	1	Table C	C	0	---			
D	0	-	D	0	---			
E	0	-	E	0	---			
F	1	Table F	F	0	---			

L2 V. Addr	V	Prot	D	R	P.Addr
0	1	rw-	1	0	9
1	1	r--	0	0	0
2	0	rw-			Disk
3	0	rw-			Disk
4	0	---			
5	0	---			
6	0	---			
7	0	---			
8	0	---			
9	0	---			
A	0	---			
B	0	---			
C	0	---			
D	0	---			
E	0	---			
F	0	---			

L2 V. Addr	V	Prot	D	R	P.Addr
0	0	---			
1	0	---			
2	0	---			
3	0	---			
4	0	---			
5	0	---			
6	0	---			
7	0	---			
8	0	---			
9	0	---			
A	0	---			
B	0	---			
C	0	---			
D	0	---			
E	1	rw-	1	0	D
F	1	rw-	0	0	1

Ασκηση 12.6

-τα 10 κοκκίνα bits δεν χρησιμοποιούνται

-τα 12 μπλε bits χαρακτηρίζουν το πρώτο επίπεδο

-τα 12 μωβ bits χαρακτηρίζουν το δεύτερο επίπεδο

-τα 12 μωβ bits χαρακτηρίζουν το τρίτο επιπεδο

-τα 14 πράσινα χαρακτηρίζουν το πρώτο bite μέσα στην σελίδα

L1 V. Address	V	L2 V. Address	L2 V. Address	V	L2 V. Address	L3 V. Address	V	Prot	P. Address
000 _{hex}	0		000 _{hex}	0		000 _{hex}	0	---	
...					
XXX	1	XXX _{hex}	XXX	1	XXX	XXX	1	rwX	XXX
...					
FFF	0		FFF	0		FFF	0	---	

Ασκηση 12.7

α) Το TBL θα πρέπει να έχει τα εξής πεδία:

-8bits Virtual address

-8bits PID

-1bit Dirty page

-3bit Page protection

-4bit Physical address

β)

V. Address	PID	Prot	D	P. Address
03	3B	r-x	0	0
03	B4	rw-	0	1
03	3C	r-x	0	0
FF	3B	rw-	0	2
FF	3C	rw-	0	3
C2	A2	-w-	1	4
E3	A3	r--	0	4

γ) Οι διεργασίες 3B και 3C έχουν κοινή φυσική μνήμη μόνο στον κωδικά του προγράμματος, όπου καμία από τις δύο δεν έχει δικαίωμα να γράψει σε αυτήν την σελίδα (Prot:r-x). Όσο για τα δεδομένα (data) των δύο διεργασιών, είναι σε διαφορετικές θέσεις φυσικής μνήμης οπότε δεν έχουν δικαίωμα να αλληλεπιδράσουν (με κανέναν τρόπο) ή μια με τα data της άλλης. Με αυτόν τον τρόπο εξασφαλίζεται η επιθυμητή προστασία για διεργασίες με διαφορετικό PID