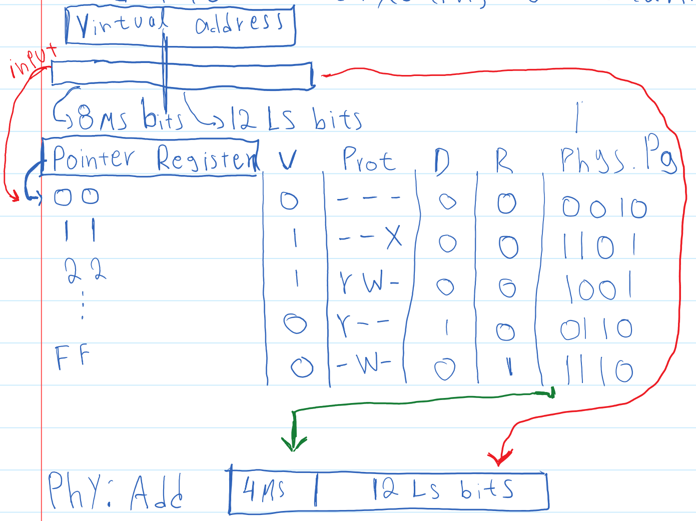
12 Ασκηση

Thursday, May 14, 2020 9:03 PM

CSd 4906 Μιχάλης Μπαστάκης



Pointer Register	V	Prot	D	R	Phys. Pg
00	0	- - -	0	0	0010
11	1	- - X	0	0	1101
22	1	r W -	0	0	1001
⋮	0	r - -	1	0	0110
FF	0	- W -	0	1	1110

Άπο 20 bits του Virt. Add. παμε σε ενα 16 bit Phys Add. Αυτο γινεται μέσω μιας αποκωδικοποιησης απο τον πινακα Μετάφρασης.

β)

Vir.Add	Valid bit	Dirty bit	Prot. bit	Phys P
00	0	0	- - -	-
01	1	0	r - x	4
02	0	0	r - x	on Disk
⋮	0	0	r - x	on Disk
09	-//-	-//-	-//-	-//-
0A	1	1	r w -	F
0B	0	0	- - -	-
...	-//-	-//-	-//-	-//-
BF	-//-	-//-	-//-	-//-
C0	1	1	rw-	9
C1	1	0	r - -	0
C2	0	0	rw -	on Disk
C3	0	0	rw -	on Disk
C4	0	6	- - -	-
⋮	0	0	- - -	-
FD	0	0	- - -	-
FE	1	1	r w -	D
FF	1	0	r w -	1

γ)

01|038(fetch): Translated to Ph. Add: 4038

0B/0F4(read): Unallocated Page: Shuting Program down

C0/01C(write): Translated to Ph.Add: 901C

02/92C(fetch): Missing Page (its on disk): The OS brings it from disk

90/000(read): NULL Page : Shuting Program down

99/F88(read): Unallocated Page: Shutting down

FE/5D8(write): Translated To Ph Add: D5D8

FF/100 (fetch): No execute Permission: Shuting down

C2/0CC(Write): Missing Page (its on Disk): The OS brings it from disk

CD/0CC(write): Unallocated Page: Shuting Down

C0/444(read): Tranlated To Ph Add: 9444

01/FF4(fetch): Translated to Ph Add: 4FF4

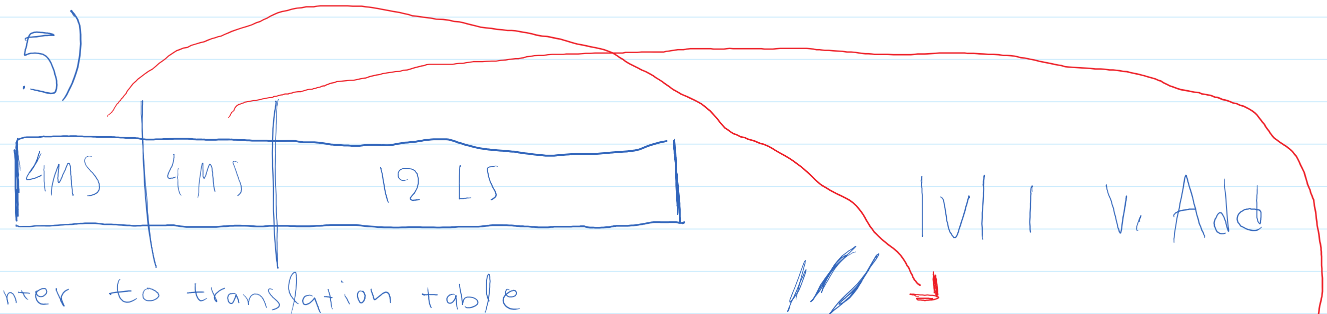
C1/FFC(write): No Write Permission: shuting down

00/BE4(write): NULL Page: Shuting down

C7/700(read): Unallocated Page: Shuting down

01/E40(write): No write Permission -> Shuting down

12.5)



Pointer to translation table

Valid Bit	Pointer to lvl2 table
1	pointer to lvl2 table 0
0	NULL
0	NULL
0	NULL
0	NULL
0	NULL
0	NULL
0	NULL

0	NULL
0	NULL
0	NULL
0	NULL
0	NULL
0	NULL
1	Pointer to lvl2 table C
0	NULL
0	NULL
1	Pointer to lvl2 table F

lvl 2 V. Add.

Translate Table 0

V. Addr	Valid	Dirty	Prot	Ph. Addr
0	0	0	- - -	-
1	1	0	r - x	4
2 - 9	0	0	r - x	-
A	1	1	r w -	F
B - F	0	0	- - -	-

Translate Table C

V. Addr	Valid	Dirty	Prot	Ph. Addr
0	1	1	r w -	9
1	1	0	r - -	0
2 - 3	0	0	r w -	on Disk
4 - F	0	0	- - -	-

Translate Table F

V. Addr	Valid	Dirty	Prot	Ph. Add.
0 - D	0	0	- - -	-
E	1	1	rw-	D
F	1		rw-	1

d) Για το συγκεκριμένο παράδειγμα έχουμε 1 πινακα 1ου επιπέδου και 3 2ου επιπέδου. Άρα έχουμε 4 πινακες οι οποίοι ο καθε πινακας περιέχει 16 θέσεις μνήμης. Άρα 16·4=64 θέσεις μνήμης συνολικα σε σχεση με τον μονοδιάστατο πινακα που εχει 256. Άρα με τον δισδιάστατο έχουμε 256-64=192 θέσεις λιγότερες Οπότε υπάρχι οικονομία.

12.6

Μια V. Add 64 bits θα είναι ως εξής:

10 MS Unused	12 MS bits lvl 1	12 MS bits lvl 2	12 MS bits lvl 3	14 LS bits V. Add. Not Transl.

Ptr to lvl 1

lvl 1

V. Addr	Ptr. Addr.
0	
⋮	
Addr	Ptr to lvl 2 table
⋮	
4096	

lvl 2

V. Addr.	Ptr. Addr.
0	
⋮	
Addr	Ptr to lvl 3 table
⋮	
4096	

lvl 3

V. Addr	Valid	Dirty	Prot	Ph. Page
0				
⋮				
Addr				
⋮				
4096				

12.7)

a)

Χρησιμοποιώ τα μεγέθη του 12.4 άρα:

- Χρειάζομαι 8 bits PID
- και 1 bit Dirty
- και Potect 3 bits
- Οπότε 8 bits Virtual Page
- Οπότε 4 bits Physical Page

β)

PID	Dirty	Prot.	V.Pg	Ph.Pg
3B	0	r-x	03	0
B4	0	rw-	03	1
3C	0	r-x	03	0
3B	0	rw-	FF	2
3B	0	rw-	FF	3
A2	1	-w-	C2	4
A3	0	r--	C3	4

γ)

Στην περιπτώση του 3B και 3C που έχουν access στο Ph.Pg 0 και οι δυο δεν δημιουργείται κανοιο προβλημα αφου κάνενα απο τα δυο Processes δεν έχει Write Permission. Οπότε μπορουν να κάνουν read κ' execute αρα δεν μπορουν να αλλοιώσουν την τιμη του Ph.Pg.

Στην περιπτωση που τα 3B έχουν Write Permission παρατηρούμε ότι δεν μπορουν να κάνουν Write στο ίδιο Ph.Pg αφου το ενα 3B κάνει Write στο 3Ph.Pg ενώ το αλλο στο 2Ph.Pg

Άρα δεν δημιουργείται κάποιο θεμα.