

exam review

last time

multi-level page tables

- very fat tree-like data structure

- multiple tables \approx tree nodes

- split virtual page number into parts

- first (most sig.) part is index into first-level table

- page table entry at first level gives location of second-level table

- second part is index into second-level table, etc.

- last-level table entry gives location in memory

allocate-on-demand/swapping

- OS deliberately doesn't setup page tables completely

- on exception, detect what program was accessing

- if accessing something that was promised to work, set it up, retry

- otherwise, crash the program or similar

quiz Q1

	virtual page #	page offset
$0x1432 =$	0001 0	100 0011 0010

from table: virtual page number 2 is physical page number 0x35

$$0x35 = 0011\ 0101$$

physical page #	page offset	
001 1010 1	100 0011 0010	$= 0x1ac32$

quiz Q2

page table lookup

physical address 0x402000 is 0x2000 bytes from beginning of table

0x2000 bytes = 0x800 page table entries

so virtual page number 0x800

final physical address access

physical address 0x201234

physical page number 0x201

page offset 0x234

so original address was 0x800234

quiz Q3

0x00099003

lower bits: (most sig) 0011 (least sig)

least significant bit: valid = 1

second least: user-mode accessible = 1

third least: writeable = 0

fourth least: executable = 0

quiz Q4

	VPN pt 1	VPN pt 2	VPN pt 3	page offset
$0x123456789A =$	00 0100 1000 (0x48)	1 1010 0010 (0x1A2)	1 0110 0111 (0x167)	0x89A

second level page table starts at physical page number 0x456

PPN	page offset	
0x456	0x000	= 0x456000

going to look at index 0x1A2 of second level PT

which is $0x1A2 \times 8$ bytes into it

$$0x456d10 = 0x456000 + 0x1A2 \times 8$$

backup slides