



SDEV-385 Homework Week 8

Jarred Glaser

10/25/2020

This week's homework questions come from your book, Silberschatz 8th edition:

1. Consider the page table for a system with 12-bit virtual and physical addresses with 256-byte pages. The list of free page frames is D,E,F (that is, D is at the lead of the list, E is second, and F is last).

Page	Page Frame
0	-
1	2
2	c
3	a
4	-
5	4
6	3
7	-
8	b
9	0

Convert the following virtual addresses to their equivalent physical addresses in hexadecimal. All numbers are given in hexadecimal. (A dash for a page frame indicates that the page is not in memory).

9EF - 0EF

111 - 211

700 - D00

0FF - EFF

2. Assume we have a demand-paged memory. The page table is held in registers. It takes 8 milliseconds to service a page fault if an empty page is available or the replaced page is not modified, and 20 milliseconds if the replaced page is modified. Memory access time is 100 nanoseconds. Assume that the page to be replaced is modified 70 percent of the time. What is the maximum acceptable page fault rate for an effective access time of no more than 200 nanoseconds?

$$\text{effective access time} = (1 - p) \times ma + p \times \text{page fault time}$$
$$ma = 100ns$$

$$\text{Chance for page to be modified} = 0.70$$

$$\text{Chance for page not to be modified} = 1 - 0.70 = 0.30$$

$$((1 - p) \times 100) + (p \times 0.30 \times 8,000,000) + (p \times 0.70 \times 20,000,000) \leq 200$$

$$(100 - 100p) + (2,400,000p) + (14,000,000p) \leq 200$$

$$(2,400,000 + 14,000,000 - 100)p \leq 100$$

$$16,399,900p \leq 100$$

$$p \leq 0.000609759\%$$

3. A certain computer provides its users with a virtual-memory space of 2^{32} bytes. The computer has 2^{18} bytes of physical memory. The virtual memory is implemented by paging, and the page size is 4096 bytes. A user process generates the virtual address 11123456. Explain how the system establishes the corresponding physical location. Distinguish between software and hardware operations.

$$\text{Virtual memory} = 2_{32}$$

$$\text{Physical memory} = 2_{18}$$

$$\text{Page Size} = 4096 = 2_{12}$$

11123456 virtual address in binary format = 0001|0001|0001|0010|0011|0100|0101|0110

Virtual memory = 2_m and page size = 2_n

Thus page table size = $m - n = 32 - 12 = 20$

The lower order 12 bits (0100|0101|0110) are used as an offset into the actual page. The upper order 20 bits (0001|0001|0001|0010|0011) are used as an offset into the page table.

The hardware handles the address translation, page size, and storing the page table in a special register. The operating system / software side handles page faults, reading the new page from a page swap, and repeating processes after a page fault.

References

(2017). Paging and Segmentation 3: Virtual Memory Paging. Retrieved from <https://www.youtube.com/watch?v=KqHNaOrxttM>.

(2020). Operating System Concepts Essentials, 2nd Edition, Silberschatz, Galvin, Gagne, Wiley Press.