Review Questions – Main Memory (Chapter 9)

Operating Systems SFWRENG 3SH3 Term 2, Winter 2023

Prof. Neerja Mhaskar

Questions

1. Consider a system in which memory protection is achieved using the **Base and Limit registers**. Suppose the value in base register = 1200 and value in limit register = 1000.
    a. If the logical address generated by the CPU = 32. To what physical address is this logical address mapped to?
    b. If the logical address generated by the CPU = 1500. To what physical address is this logical address mapped to?
2. Consider a system in which the logical addresses are mapped to physical addresses using relocation registers. Suppose the value of the relocatable register =14000 and the limit register = 1200. If the CPU generated a logical address = 1300. To what physical address is the logical address mapped to?
3. Explain the difference between internal and external fragmentation.
4. Given six memory partitions of 300 KB, 600 KB, 350 KB, 200 KB, 750 KB, and 125 KB (in order), how would the first-fit, best-fit, and worst-fit algorithms place processes of size 115 KB, 500 KB, 358 KB, 200 KB, and 375 KB (in order)? Rank the algorithms in terms of how efficiently they use memory.
5. Consider a physical memory of size 64MB. It is partitioned using the contiguous dynamic partitioning scheme. The operating system uses 8MB of memory space. At a given instance in time, below is the snapshot of memory:

OS = 8MB	P1=16MB	P2=20MB	P3=18MB	2MB

Process P2 terminates and releases 20MB space and P4 is brought into main memory.

OS = 8MB	P1=16MB	P_4=14MB	6MB	P3=18MB	2MB

a) Can Process P5 = 8MB be brought into memory? If not, why?
b) Can Process P6 = 10MB be brought into memory? If not, why?

6. Compare the memory organization schemes of contiguous memory allocation (fixed and dynamic partitioning), and pure paging with respect to the following issues: a) External fragmentation b) Internal fragmentation.

7. Assuming a 1-KB page size, what are the page numbers and offsets for the following address references (provided as decimal numbers):
   a. 3085
   b. 42095
   c. 215201
   d. 650000
   e. 2000001

8. Consider a logical address space of 256 pages with a 4-KB page size, mapped onto a physical memory of 64 frames.
   a. How many bits are required in the logical address?
   b. How many bits are required in the physical address?

9. The BTV operating system has a 21-bit virtual address, yet on certain embedded devices, it has only a 16-bit physical address. It also has a 2-KB page size. How many entries are there in each of the following?
   a. A conventional, single-level page table.
   b. An inverted page table.

10. What is the maximum amount of physical memory in the BTV operating system?

11. Consider a computer system with a 32-bit logical address and 4-KB page size. The system supports up to 512 MB of physical memory. How many entries are there in each of the following?
   c. A conventional single-level page table
   d. An inverted page tables.

12. Consider a paging system with the page table stored in memory.
   e. If a memory reference takes 200 nanoseconds, how long does a paged memory reference take?
   f. If we add TLBs, and 75 percent of all page-table references are found in the TLBs, what is the effective memory reference time? (Assume that finding a page-table entry in the TLBs takes 2 nanoseconds, if the entry is present.)

13. Assume a program has just referenced an address in virtual memory. Describe a scenario how each of the following can occur: (If a scenario cannot occur, explain why.)
   g. TLB miss with no page fault
   h. TLB miss and page fault
   i. TLB hit and no page fault
   j. TLB hit and page fault

14. Consider Hierarchical paging, with logical address space of 2^{48} bytes, page size = 1024 = 2^{10} bytes and each page table entry takes 8 bytes. A multi-level page table is used because each table must be contained within a page.
   a. How many levels of page table are required?

b. What is the distribution of no. of bits to represent each level of the multi-level page table, and the page offset?

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

Page	Page Frame
0	–
1	2
2	C
3	A
4	–
5	4
6	3
7	–
8	B
9	0

a. 9EF
b. 111
c. 700
d. 0FF