# Homework #4 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_

•Grading: 3 = correct

2 = almost

1 = an attempt

0 = nothing

•Score: Points / Possible

# (53 pts) (Name) (Section)

**Memory Management (Chapter 7)**

**Virtual Memory (Chapter 8)**

|  |  |
| --- | --- |
| Questions: | Answers: |
| 1. (7.12) (10 points) Consider a simple (1 level) byte addressable paging system with the following parameters: 232 bytes of physical memory; page/frame size of 210 bytes; 216 pages of logical address space.  a. How many bytes in the logical address space?  b. How many bits are in a logical address?  c. How many pages in a physical address space?  d. How many bytes are in a physical address?  e. How many bits are in a physical address?  f. How many bits in the physical address specify the frame?  g. How many bits in each page table entry?  (Include valid, dirty, and pin bits.) | |
| 2. Using a clock replacement algorithm, how many page faults would there be?   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Frame | **0** | **1** | **2** | **3** | **4** | **3** | **4** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **7** | **8** | **0** | **1** | **3** | **7** | **8** | **0** | **1** | **2** | **3** | **7** | **8** | **7** | **8** | **0** | **1** | | **0** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | **1** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | **2** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | **3** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | |
| 3. Using a clock replacement algorithm, how many page faults would there be?   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Frame | **0** | **1** | **2** | **3** | **4** | **3** | **4** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **7** | **8** | **0** | **1** | **3** | **7** | **8** | **0** | **1** | **2** | **3** | **7** | **8** | **7** | **8** | **0** | **1** | | **0** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | **1** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | **2** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | **3** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | **4** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | |
| 4. (8.10) Assuming a page size of 4 Kbytes and that a page table entry takes 4 bytes, how many levels of page tables would be required to map a 64-bit address space if the top-level page table is the smallest? It is desired to limit the page table size to one page. |  |
| 5. (6 points) Answer the following questions:a. What part(s) of main memory must be initialized before enabling virtual memory?b. In which memory space (swappable or non-swappable) would you find root page tables? User page tables? User data frames? |  |
| 6. (6 points) Consider a paged virtual memory system with 32-bit virtual addresses and 1K-byte pages. Each page table entry requires 32 bits. It is desired to limit the page table size to one page.a. How many levels of page tables are required?b. What is the size of the page table at each level? Hint: One page table size is smaller.c. The smaller page size could be used at the top level or the bottom level of the page table hierarchy. Which strategy consumes the least number of pages? |  |
| 7. (12 points) Document the function arguments and return values for the os345mmu.c function:int accessPage(int pnum, int frame, int rwnFlg)a. pnumb. framec. rwnFlgd. return value |  |
| 8. (6 points) Consider a paged logical address space (composed of 32 pages of 2 Kbytes each) mapped into a 1-Mbyte physical memory space.a. What is the format of the processor’s logical address?b. What is the size (length and width) of the page table? Put entries on byte boundaries and disregard any “access rights” bits.c. What is the effect on the page table if the physical memory space is reduced by half? |  |
| 9. (4 points) Using the display frame table output,a. How many frames are available in the LC-3 frame bit table?b. What are the beginning and ending LC-3 memory addresses of the available frames? | dft |