# 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? 216+10 = 226 bytes = 67,108,864 bytes  b. How many bits are in a logical address? 216 pages + 210 bytes/page = 16 + 10 = 26  c. How many pages in a physical address space? 232-10 = 222 = 4,194,304 pages  d. How many bytes are in a physical address? 232 bytes = 4,294,967,296  e. How many bits are in a physical address? 32  f. How many bits in the physical address specify the frame? 32 – 10 = 22  g. How many bits in each page table entry? 32 – 10 + 3 = 25  (Include valid, dirty, and pin bits.) | |
| 2. Using a clock replacement algorithm, how many page faults would there be? 27   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 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** | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 5 | 5 | 5 | 5 | 0 | 0 | 0 |  |  |  |  |  |  |  |  |  |  |  | 0 | 7 | 7 | | **1** |  | 1 | 1 | 1 | 1 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 6 | 6 | 6 | 6 | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  | 1 | 1 | 2 | | **2** |  |  | 7 | 7 | 7 | 7 | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 3 | 7 | 7 | 7 | 7 | 5 |  |  |  |  |  |  |  |  |  |  |  | 5 | 5 | 5 | | **3** |  |  |  | 8 | 8 | 8 | 8 | 0 | 0 | 0 | 0 | 4 | 4 | 4 | 4 | 8 | 8 | 8 | 8 |  |  |  |  |  |  |  |  |  |  |  | 6 | 6 | 6 | |  | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |  |  |  |  |  |  |  |  |  |  |  | x | x | x | | |
| 3. Using a clock replacement algorithm, how many page faults would there be? 17   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 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** | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 5 | 5 | 5 | 5 | 0 | 0 | 0 |  |  |  |  |  |  |  |  |  |  |  | 0 | 7 | 7 | | **1** |  | 1 | 1 | 1 | 1 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 6 | 6 | 6 | 6 | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  | 1 | 1 | 2 | | **2** |  |  | 7 | 7 | 7 | 7 | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 3 | 7 | 7 | 7 | 7 | 5 |  |  |  |  |  |  |  |  |  |  |  | 5 | 5 | 5 | | **3** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | **4** |  |  |  | 8 | 8 | 8 | 8 | 0 | 0 | 0 | 0 | 4 | 4 | 4 | 4 | 8 | 8 | 8 | 8 |  |  |  |  |  |  |  |  |  |  |  | 6 | 6 | 6 | | |
| 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. | # pages = 264 / 212 = 252 pages# pte/page = 212 / 22 = 210 pte/page  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | o | o | o | o | o | o | o | o | o | o | o | o |  6 |
| 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? | a. Root page table and memory allocation tableb. Root page tables are NOT in swap space. User page tables may be swapped to disk, hence swap space. User data frames are always in swap space. |
| 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? | # pages = 232 / 210 = 222 pages# pte/page = 210 / 22 = 28 pte/page  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 6 bits | | | | | | 8 bits | | | | | | | | 8 bits | | | | | | | | 10 bits | | | | | | | | | | | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | o | o | o | o | o | o | o | o | o | o |  a. 3 levelsb. 256 (64×4), 1024, 1024 bytesc. toptop: 1 + 64 + (64 × 256) + ((64 × 256) × 256) = 4,210,753bottom: 1 + 256 + (256 × 256) + ((256 × 256) × 64) = 4,407,553 |
| 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 | a. pnum = the page number of swap space for read/write operations. Not used to new write.b. frame = physical frame number to be read/written to swap space.c. rwnFlg= intent of function call: 0=init, 1=return address of pnum, 2=write new swap page, 3=write old swap page, 4=read swap page, 5=free swap paged. return value= address of swap page when requesting page address, otherwise, the swap page number. |
| 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? | Logical = 32 pages × 2 Kb = 64 Kb = 16 bitsPhysical = 1 Mb 220 / 211 = 29 framesa) 5 bit index, 11 bit offset  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1 | 1 | 1 | 1 | 1 | o | o | o | o | o | o | o | o | o | o | o |  b) 32 entries × 2 bytes (29 frames) = 64 bytesc) One less frame bit needed, thus 32 bytes. |
| 9. (4 points) Using the display frame table output,a. How many frames are available in the LC-3 frame bit table? 93b. What are the beginning and ending LC-3 memory addresses of the available frames? x3000 - x477f | dft |