CSC 4100 Exam #2 Name:_____
10/07/2020

A system uses **segmentation** with four segments, one unused, to map a 128KB virtual address space to physical memory. If 00 is the code segment, 01 is the heap segment, and 11 is the stack segment, translate the virtual address **0x089DE** to the physical address with the given settings (the translation should be a valid address). **Answer in hex**. (**20 points**)

Segment	Base	Size	Grows
00:	0x343A	8K	1
01:	0x49A2	8K	1
11:	0xFF99	8K	0

Suppose a system can only hold **four** pages in physical memory. Assume that some process accesses six pages in the given order (**show your work**):
0, 4, 1, 0, 2, 1, 5, 3, 0, 5, 2, 4, 2, 0, 5, 2

Compute the hit rate (omit compulsory misses)
using (actual, not approx) **LRU** (**show your work**). (**20 points**)
Why are **approx** LRU algorithms necessary? (**5 points**)

Explain how the hardware knows that a particular page has been **swapped out** to disk and how does the OS know where to find that page as it is now needed in main memory again.
(**10 points**)

Suppose each process has 2MB of virtual address space and that each page is 2K in size. Show how the virtual address is divided up in the rectangle below between bits for the **VPN** and bits for the **offset** (in a **single level** page table). Indicate the number of bits for each. (**5 points**)

How many pages does the page table described above use up given that the PTE is 8 bytes? Why is a page table of more than one page a problem in this case? Explain. (**5 points**)

If the system in the previous question uses a multi-level (2-level) page table, show the number of bits for the page directory index, page table index, and offset. (**5 points**)

In the multi-level page table of the previous question, assume the page directory has three entries at index 0 (64), at index 2 (39), and at index 3 (119). At PFN 64, there is a single entry at index 114 (41). At PFN 39, there is a single entry at index 222 (97). At PFN 119, there is a single entry at index 154 (61). Translate the virtual address **0x1CD75C**. Show the **intermediate address** (using the PTI) in hex that you need to complete the translation. Assume base 10 above unless specified as hex. (**20 points**)

Assume the following translation (unrelated to the previous question) is stored in the TLB.
OxA99C7 (virtual address) translates to **0xD6147** (physical address)
Fill out the TLB entry below. You should be able to deduce what TlbEntry.PFN is.
The number of **offset bits** is **7**. (**10 points**)

Use hex in the table. VPN PFN

Offset = Virtual Address & OFFSET_MASK
PhysAddr = (TlbEntry.PFN << SHIFT) **|** Offset //note: | is bitwise or
Write down in binary (TlbEntry.PFN << SHIFT). Write down in binary Offset.
Show the **bitwise or** between these two (producing PhysAddr) (**10 points**)