Answer Key

Name:_____

CSE410 Quiz June 15th 2018

Answer Bank:
Associative || Software || Direct || Indexed || Hardware || Page Table || Inverted Page Table || Page Table Entry(ies) || Physical Address(es) || Virtual Address(es) || TLB || Cache || Locality || Working|Thrashing || Multiprogramming || Page Fault || Offset || Present || Fetching || Second Level Page Table || Resident || Demand Paging || Prepaging || Root/First Level Page Table || Replacement || Modified || Answer Not in List

Question 1 (.5 pts): Locality is the principle/concept that is fundamental to the success of virtual memory systems and explains that memory references tend to cluster.

Question 2 (1 pts): The resident set is the pages of a process currently in memory. The working set is the pages of a process that SHOULD be in memory.

Question 3 (.5 pts): A higher level of multiprogramming is an advantage of the partial loading of processes, in addition to being able to offer processes more memory than all of RAM.

Question 4 (.5 pts): Thrashing is a state in which the CPU spends more time swapping process pages in and out of memory than actually executing instructions.

Question 5 (.5 pts): A(n) page fault occurs when there is a request to a process page that is not currently in main memory.

Question 6 (1 pts): These two bits are added to the page table entry in a virtual memory system: present, and modified.

Question 7 (.5 pts): If there is no free frame in memory a(n) replacement algorithm is needed to decide which page is swapped out.

Question 8 (.5 pts): Recently used page table entries is/are stored in the TLB.

Question 9 (.5 pts): The TLB is a hardware construct. (Hardware or Software)

Question 10 (.5 pts): The TLB uses Associative mapping, a system that searches all entries at the same time.

Question 11 (1 pts): In a virtual address using a two level page table the first set of bits are used to index the root page table, the second set of bits are used to index the 2nd level page table, and the last bits are used for the offset.

******Questions on the next page do not use the answer bank and all work must be shown*******

Name:_____

Question 12 (4 pts): 1 2 3 2 1 5 2 1 6 2 7 Use the clock algorithm, with 3 frames per process. Show all page faults, the use bit, and the location of the pointer after each memory reference.

1	2	3	2	1	5	2	1	6	2	7
1*	1*	→1*	→1*	→1*	5*	5*	→5*	5	2*	→2*
→	2*	2*	2*	2*	→2	→2*	2	6*	→6*	6
	→	3*	3*	3*	3	3	1*	→1*	1	7*
F	F	F			F		F	F	F	F

Question 13 (3 pts): Use the buddy system to show how memory will be partitioned after each operation given the following request stream. Assume main memory is of size 2^{21} bytes.

1 - Request: A - 250 KB
2 - Request: B - 129 KB
3 - Request: C - 50 KB
4 - Request: D - 400 KB
5 - Release: B
6 - Release: D
7 - Release: A
8 - Release: ~~D~~ C

2 MB							
A	256KB	512KB			1MB		
A	B	512KB			1MB		
A	B	256	128	64 C	1MB		
A	B	256	128	64 C	512	D	
A	256	256	128	64 C	512	D	
A	256	256	128	64 C	~~D~~	1MB	
512		256	128	64 C	1MB		
2 MB							

Question 14 (3 pts): Assume 40 bit virtual addresses, a page size of 8 KB, and a page table entry size of 8 bytes. How many pages can each process have?

40 bits - 13 bits offset = 27 bits for page# ⇒ 2^{27} pages

Question 15 (3 pts): Make the same assumptions as question 14, how many pages are needed to store the page table for a process?

2^{27} pages per process · 8 bytes per PTE = 2^{30} bytes per page table

2^{30} total bytes / 2^{13} page size = 2^{17} pages to store page table