CS152 Worksheet 4

Q1: Linear vs Hierarchical Page Tables

Consider 4 GiB (32-bit) of addressable virtual memory, 4 KiB pages, 4-byte PTEs

- How many bits in the page offset?
- How many bits in the page number?
- How many pages?

Consider a linear page table for a process with only 1 page mapped to physical memory (paged in)

- How many valid PTEs?
- Total size of page table?

Consider a 2-level page table for a process with only 1 page mapped to physical memory (paged in). Assume that VPN bits are split equally between the two levels.

- How many valid PTEs?
- Total size of page table structures?

Q2. Multi-level Page Table

The table on the next page shows the contents of a portion of physical memory used for page tables. Assume the system uses 32-bit words, 16 byte pages, two-level page tables, and a fully associative four-entry TLB with LRU eviction. At the beginning, the TLB is empty and the free pages list contains 0x9, 0x5, 0xA, 0x7, 0x1, 0x3, 0xB, 0xD, 0xE, and 0xF in that order. For the following virtual memory trace, indicate whether the access results in a TLB hit, a page table hit, or a page fault, and give the translated physical address. Fill out the memory table and the TLB with its final state. Assume that the page table base register is set to 0.

1. How many bytes of virtual memory are addressable?
2. How many bytes of physical memory are addressable?
3. Why might DRAM size > virtual address space size be useful?

Virtual Address	**Index1**	**Index2**	**TLB hit/miss**	**Page hit/ Page fault**	**Physical Address**
0x68					
0x14					
0x6C					
0x90					
0x74					
0xE4					
0x18					
0xD0					

TLB				
VPN				
PPN				

Addr	**Contents**
0x00	0x06
0x04	0x04
0x08	0x02
0x0C	
0x10	
0x14	
0x18	
0x1C	
0x20	0x08
0x24	
0x28	0x0C
0x2C	
0x30	
0x34	
0x38	
0x3C	
0x40	
0x44	
0x48	0x12
0x4C	0x11
0x50	
0x54	
0x58	
0x5C	
0x60	
0x64	0x13
0x68	
0x6C	

Table 1. Initial contents of memory