Exercise 5.10

As described in Section 5.4, virtual memory uses a page table to track the mapping of virtual addresses to physical addresses. This exercise shows how this table must be updated as addresses are accessed. The following table is a stream of virtual addresses as seen on a system. Assume 4 KB pages, a four-entry fully associative TLB, and true LRU replacement. If pages must be brought in from disk, increment the next largest page number.

4095，31272，15789，15000，7193，4096，8912

TLB

Valid	Tag	Physical Page Number
1	11	12
1	7	4
1	3	6
0	4	9

Page table

Index	Valid	Physical Page or In Disk
0	1	5
1	0	Disk
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12

5.10.1 [10] <5.4> Given the address stream in the table, and the initial state of the TLB and page table, show the final state of the system. List for each reference if it is a hit in the TLB, a hit in the page table, or a page fault.

5.10.2 [15] <5.4> Repeat Exercise 5.10.1, but this time use 16 KB pages instead of 4 KB pages. What would be some of the advantages of having a larger page size? What are some of the disadvantages?

5.10.3 Moved to Backup problem

Exercise 5.10 Part B

There are several parameters that impact the overall size of the page table. Listed below are several key page

table parameters.

Virtual Address Size	Page Size	Page Table Entry Size
32 bits	4 KB	4 bytes

5.10.4 [5] <5.4> Given the parameters in the table above, calculate the total page table size for a system running five applications that utilize half of the memory available.

5.10.5 [10] <5.4> Given the parameters in the table above, calculate the total page table size for a system running five applications that utilize half of the memory available, given a two-level page table approach with 256 entries. Assume each entry of the main page table is 6 bytes. Calculate the minimum and maximum amount of memory required.

5.10.6 Moved to backup problems

Exercise 5.11

In this exercise, we will examine space/time optimizations for page tables. The fo1lowing table shows parameters of a virtual memory system.

Virtual address (bits)	Physical DRAM installed	Page size	PTE size (byte)
32	4 GB	8 KB	4

5.11.1 [10] <5.4> For a single-level page table, how many page table entries (PTEs) are needed? How much physical memory is needed for storing the page table?

5.11.2 [10] <5.4> Using a multilevel page tab1e can reduce the physical memory consumption of page tab1es, by only keeping active PTEs in physical memory. How many levels of page tables will be needed in this case? And how many memory references are needed for address translation if missing in TLB?

5.11.3 Moved to backup problems

Exercise 5.11 Part B

The following tab1e shows the contents of a 4-entry TLB.

Entry-ID	Valid	VA page	Modified	Protection	PA page
1	1	140	1	RW	30
2	0	40	0	RX	34
3	1	200	1	RO	32
4	1	280	0	RW	31

5.11.4 [5] <5.4> Under what scenarios would entry 2's valid bit be set to zero?

5.11.5 [5] <5.4> What happens when an instruction writes to VA page 30?

5.11.6 [5] <5.4> What happens when an instruction writes to VA page xxx?

Exercise 5.12

In this exercise, we will examine how replacement policies impact miss rate. Assume a two-way set-associative cache with four blocks. You may find it helpful to draw a table like those found on page 483 to solve the problems in this exercise, as demonstrated below on the address sequence “0,1,2,3,4”.

Address of memory block accessed	Hit or miss	Evicted block	Contents of cache blocks after reference			
			Set 0	Set 0	Set 1	Set 1
0	Miss		Men[0]			
1	Miss		Men[0]		Men[1]	
2	Miss		Men[0]	Men[2]	Men[1]	
3	Miss		Men[0]	Men[2]	Men[1]	Men[3]
4	Miss	0	Men[4]	Men[2]	Men[1]	Men[3]
…						

The following table shows address sequences.

Address sequence
0,2,4,0,2,4,0,2,4

5.12.1 [5] <5.3, 5.5> assuming an LRU replacement policy, how man hits does this address sequence exhibit?

5.12.2 [5] <5.3, 5.5> assuming an MRU (most recently used) replacement policy, how many hits does this address sequence exhibit?

5.12.3 [5] <5.3, 5.5> Moved to backup problems

5.12.4 [10] <5.3, 5.5> Moved to backup problems

5.12.5 [10] <5.3, 5.5> describe why it is difficult to implement a cache replacement policy that is optimal for all address sequences.

Exercise 5.13 Moved to backup problems