

## CS 537-Intro to Operating Systems

### Worksheet 3 - Paging & TLBs

July 10<sup>th</sup>, 2017 (Monday)

1. Assume a system with a simple array-based linear page table.

**Parameters:**

- virtual address space size = 16 KB
- physical memory size = 32 KB
- page size = 1 KB
- Size of one Page Table Entry (PTE) = 4 bytes

Assume for a particular process in this system, only 3 pages in its virtual address space are valid. All the other pages are invalid. The Virtual Page Numbers (VPNs) of the three valid pages are 0, 3, and 15.

- a. How much **physical memory** is used for this **process' pages** when this process is running? You may assume that all the valid pages of this process must be in the physical memory (RAM) for this process to be able to run.
- b. How much **physical memory** space is used for storing this **process' page table**? Remember, only 3 pages of this process' address space are currently valid and the rest are invalid.

2. Assume a system with a simple linear page table and a hardware managed TLB.

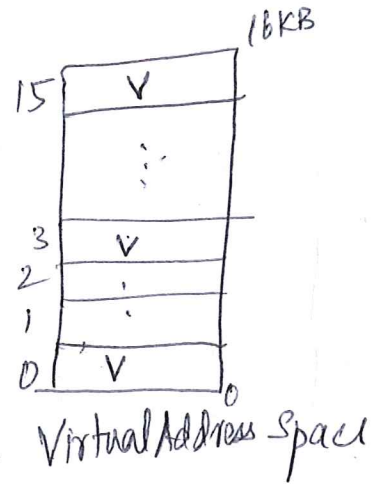
**Parameters:**

- virtual address space size = 64 bytes
- physical memory size = 128 bytes
- page size = 16 bytes

Let's assume we have an array of **ten 4-byte integers** in memory, starting at virtual address 16. Let's further assume that we **access all the elements in the array**, sequentially, starting at array index 0. The references to these array elements are the **only memory references** that we are interested in.

- a. For each element that is accessed in the array, write if it's a **TLB Hit (H)** or a **TLB Miss (M)**. (e.g. HMMHMMHMMH)
- b. What is the **TLB hit ratio** for accessing the entire array?
- c. If the **page size** is **increased** from 16 bytes to **32 bytes**, what is the **TLB hit ratio** for accessing the same array? Assume the TLB was initially empty before accessing the array.

KEY



CS 537-Intro to Operating Systems  
 Worksheet 3 - Paging & TLBs  
 July 10<sup>th</sup>, 2017 (Monday)

1. Assume a system with a simple array-based linear page table.  
 Parameters:

- virtual address space size = 16 KB
- physical memory size = 32 KB
- page size = 1 KB
- Size of one Page Table Entry (PTE) = 4 bytes

Assume for a particular process in this system, only 3 pages in its virtual address space are valid. All the other pages are invalid. The Virtual Page Numbers (VPNs) of the three valid pages are 0, 3, and 15.

- How much **physical memory** is used for this process' pages when this process is running? You may assume that all the valid pages of this process must be in the physical memory (RAM) for this process to be able to run. 3 KB
- How much **physical memory** space is used for storing this process' page table? Remember, only 3 pages of this process' address space are currently valid and the rest are invalid. 64 bytes

2. Assume a system with a simple linear page table and a hardware managed TLB.  
 Parameters:

- virtual address space size = 64 bytes
- physical memory size = 128 bytes
- page size = 16 bytes

Let's assume we have an array of **ten 4-byte integers** in memory, starting at virtual address 16. Let's further assume that we **access all the elements in the array**, sequentially, starting at array index 0. The references to these array elements are the **only memory references** that we are interested in.

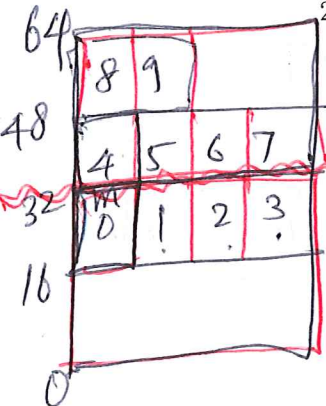
a. For each element that is accessed in the array, write if it's a **TLB Hit (H)** or a **TLB Miss (M)**. (e.g. HMMHMHMMHM)

M H H H M H H H M H

b. What is the TLB hit ratio for accessing the entire array? 70%

c. If the page size is increased from 16 bytes to 32 bytes, what is the TLB hit ratio for accessing the same array? Assume the TLB was initially empty before accessing the array.

M H H H M H H H H H 80%



PT

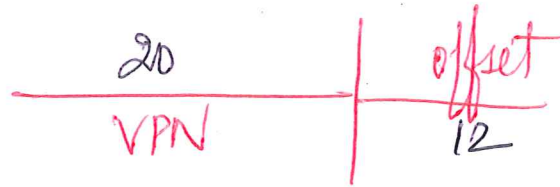
	V	PPN
0	V	7
1	0	—
2	0	—
3	V	4
4	0	—
5	0	—
...	...	...
14	0	—
15	V	3

→ 4 bytes

$$16 \times 4 = 64$$

64  
bytes

32-bit



$$2^{20} \text{ V.P.s}$$

$$\Rightarrow 2^{20} \times 4 \text{ PTEs}$$

$$\Rightarrow 4 \text{ MB for 1 PT.}$$

$$100 \text{ processes} \Rightarrow 400 \text{ MB for Page Tables.}$$