**Homework 6: Virtual Memory**

**Due date: 11:59PM Sunday April 25**

1. Consider a computer with a paged logical address space with 8 pages and each page is 4 Kbytes. The logical address space is mapped into a 256-Kbyte of physical memory space. (30pts)
   1. Draw the fields in the logical and physical addresses and show the number of bits of each field.

Logical:

|  |  |
| --- | --- |
| 16 Bits – Virtual Address | 8 Bits – Page Offset |

Physical:

|  |  |
| --- | --- |
| 8 Bits – Physical Page Number | 8 Bits – Page Offset |

* 1. Draw the page table of a process and show the number of entries in the table and number of bits per entry.

|  |  |
| --- | --- |
| Valid Bit (1) | Physical Page Number (8 Bits) |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

* 1. Populate the page table for process, namely A, which is currently running on the CPU. Several pages of process A is in the physical memory as follows:

|  |  |
| --- | --- |
|  | … |
| #frame 10 | Page 5 of Process A |
| #frame 11 | Page 4 of Process A |
| #frame 12 | Page 0 of Process A |
| #frame 13 | Page 7 of Process A |
|  | … |

|  |  |
| --- | --- |
| Valid Bit (1) | Physical Page Number (8 Bits) |
| 1 | 00001100 |
| 0 | 00000000 |
| 0 | 00000000 |
| 0 | 00000000 |
| 1 | 00001010 |
| 1 | 00001011 |
| 0 | 00000000 |
| 1 | 00001101 |

1. Consider paged virtual memory systems. Assume a page size of 256 bytes (28), and that processes in this system can have a maximum virtual address space of 16K bytes (214). The system is currently configured with 8K (213) bytes of physical memory. (30pts)
   1. How many pages are in the virtual address space?

(214/28) = (26 = 64 pages)

* 1. How many page frames are in the physical address space?

(213/28) = (25 = 32 page frames)

* 1. A user process generates the virtual address 12,345 (0011000000111001 in binary). Explain how the system establishes the corresponding physical address assuming that the hardware memory management unit and transfer lookaside buffer (TLB) is used.

First, the system looks for the virtual address 12345 in the TLB using a linear search. If the virtual address appears in a TLB entry, then it is a TLB hit. The system then retrieves the physical address from the TLB entry.

If the virtual address is not in the TLB, then it is a TLB miss. The system uses the virtual address as an array index, and searches through the page table. If there is an entry at that page table location, then the system retrieves the physical address from that entry.

If there is no entry, then it is a page fault, and the TLB, Page Table, and Core Map must be updated to point to a new physical address.

1. Consider a paged virtual memory system with a physical memory that can only contain 4 pages. Assume the execution of a program generates the following address trace

*a b c d d f b e b e*

where *a*, *b*, *c*, *d*, *e*, and *f* are the pages referenced and the page frames are initially empty. (40pts)

* 1. How many page faults occur with first-in-first-out Page Replacement? 4

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Time* | 1 2 | | 3 4 | | 5 6 | | 7 8 | | 9 10 | |
| *Request* | *a* | *b* | *c* | *d* | *d* | *f* | *b* | *e* | *b* | *e* |
|  | a |  |  |  |  | f |  |  |  |  |
|  |  | b |  |  |  |  | b | e |  |  |
|  |  |  | c |  |  |  |  |  | b |  |
|  |  |  |  | d | d |  |  |  |  | e |
| Fault? |  |  |  |  |  | Y |  | Y | Y | Y |

* 1. How many page faults occur with LRU Page Replacement? 2

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Time* | 1 2 | | 3 4 | | 5 6 | | 7 8 | | 9 10 | |
| *Request* | *a* | *b* | *c* | *d* | *d* | *f* | *b* | *e* | *b* | *e* |
|  | a |  |  |  |  | f |  |  |  |  |
|  |  | b |  |  |  |  | b |  | b |  |
|  |  |  | c |  |  |  |  | e |  | e |
|  |  |  |  | d | d |  |  |  |  |  |
| Fault? |  |  |  |  |  | Y |  | Y |  |  |