**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 – 5 bits Physical – 18 bits**

|  |  |
| --- | --- |
| 0 | 32 |
| 1 | 32 |
| 2 | 32 |
| 3 | 32 |
| 4 | 32 |
| 5 | 32 |
| 6 | 32 |
| 7 | 32 |

|  |  |
| --- | --- |
| 0 |  |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |

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

Frames - Size: 64 bits per entry

|  |  |
| --- | --- |
| 0 |  |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| … |  |
| 32 |  |

* 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 |
|  | … |

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?

**26 or 64**

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

**128 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.

**The system will first reference the virtual memory (logical address space) to access the information. This will then be sent to the page tables with physical memory references. The physical address space will then access the page table, grabbing info from the logical address space and translating it to its 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?

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Time* | 1 2 | | 3 4 | | 5 6 | | 7 8 | | 9 10 | |
| *Request* | *a* | *b* | *c* | *d* | *d* | *f* | *b* | *e* | *b* | *e* |
| Frame 1 | A | A | A | A | A | B | B | C | D | D |
| Frame 2 |  | B | B | B | B | C | C | D | F | F |
| Frame 3 |  |  | C | C | C | D | D | F | E | E |
| Frame 4 |  |  |  | D | D | F | F | E | B | B |
| Fault? | \* | \* | \* | \* |  | \* |  | \* | \* |  |

**Total Number of Faults using FIFO: 7**

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

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Time* | 1 2 | | 3 4 | | 5 6 | | 7 8 | | 9 10 | |
| *Request* | *a* | *b* | *c* | *d* | *d* | *f* | *b* | *e* | *b* | *e* |
| Frame 1 | A | A | A | A | B | C | D | D | F | B |
| Frame 2 |  | B | B | B | C | D | D | F | B | E |
| Frame 3 |  |  | C | C | D | D | F | B | E | B |
| Frame 4 |  |  |  | D | D | F | B | E | B | E |
| Fault? | \* | \* | \* | \* |  | \* | \* | \* |  |  |

**Total Number of Faults using LRU: 7**