1.Consider the following page reference string:

7,2,3,1,2,5,3,4,3,7,7,3,0,3,4,6,2,3,0,1

Assume demand paging with 3 frames, how many page faults would occur for the following page replacement algorithms?

LRU

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

Faults: 15

FIFO

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

Faults: 15

Optimal

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

Faults: 11

2.A certain computer provides its users with a virtual memory of 232 bytes. The computer has 222 bytes physical memory. The virtual memory is implemented by paging and the page size is 4,096 bytes. A user process generates the virtual address 11123457. Explain how the system establishes the corresponding physical location.

The virtual address in binary is 0001 0001 0001 0010 0011 0100 0101 0111

Since the page size is 4096 bytes, the page table size is 2^20 bytes.

Therefore, the lower 12 bits 0100 0101 0111 are used as the displacement into the page, while the other 20 bits 0001 0001 0001 0010 0011 are used as the displacement in the page table. The offset bits are then concatenated to the resulting physical page number (from the page table), to form the final address.

3. Write a C program(no requirement for threads, mutexes or makefiles) that takes from the command line the page size. It checks to make sure it is valid( a multiple of 2) and if so then prompts the user to enter a decimal address and it prints out the page and the offset.

Code in Ubuntu (already done)

Example

./convert 4092

Invalid page size

./convert 4096

Enter the decimal address 231567

The page number is 56

The offset is 2191