HW Chapter 8

8.2 A process consisting of n pages has been allocated m empty frames (assume m < n). The process will access all of its pages at some point during execution. Its page reference string is of length p.

a) What is a lower bound on the number of page faults?

**n is the lower bound of page faults.**

b) What is an upper bound on the number of page faults?

**n \* m is the upper bound on the number of page faults**

8.8 Consider the following page reference string:

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

How many page faults would occur for the following replacement algorithms, assuming four frames are allocated to the process?

1. FIFO

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 2 | 1 | 5 | 6 | 2 | 1 | 2 | 3 | 7 | 6 | 3 | 2 | 1 | 2 | 3 | 6 |
| 1 | 1 | 1 | 1 | 1 | 1 | **5** | 5 | 5 | 5 | 5 | **3** | 3 | 3 | 3 | 3 | **1** | 1 | 1 | 1 |
|  | 2 | 2 | 2 | 2 | 2 | 2 | **6** | 6 | 6 | 6 | 6 | **7** | 7 | 7 | 7 | 7 | 7 | **3** | 3 |
|  |  | 3 | 3 | 3 | 3 | 3 | 3 | **2** | 2 | 2 | 2 | 2 | **6** | 6 | 6 | 6 | 6 | 6 | 6 |
|  |  |  | 4 | 4 | 4 | 4 | 4 | 4 | **1** | 1 | 1 | 1 | 1 | 1 | **2** | 2 | 2 | 2 | 2 |
| F | F | F | F |  |  | F | F | F | F |  | F | F | F |  | F | F |  | F |  |

14 faults

1. Optimal

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 2 | 1 | 5 | 6 | 2 | 1 | 2 | 3 | 7 | 6 | 3 | 2 | 1 | 2 | 3 | 6 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | **7** | 7 | 7 | 7 | **1** | 1 | 1 | 1 |
|  | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
|  |  | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
|  |  |  | 4 | 4 | 4 | **5** | **6** | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| F | F | F | F |  |  | F | F |  |  |  |  | F |  |  |  | F |  |  |  |

8 faults

1. LRU

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 2 | 1 | 5 | 6 | 2 | 1 | 2 | 3 | 7 | 6 | 3 | 2 | 1 | 2 | 3 | 6 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | **6** | 6 | 6 | 6 | 6 | 6 | 6 |
|  | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
|  |  | 3 | 3 | 3 | 3 | **5** | 5 | 5 | 5 | 5 | **3** | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
|  |  |  | 4 | 4 | 4 | 4 | **6** | 6 | 6 | 6 | 6 | **7** | 7 | 7 | 7 | **1** | 1 | 1 | 1 |
| F | F | F | F |  |  | F | F |  |  |  | F | F | F | F |  | F |  |  |  |

11 faults

8.18 A certain computer provides its users with a virtual-memory space of 232 bytes. The computer has 222 bytes of physical memory, with a frame size of 4096 bytes. A user process generates the hexadecimal virtual address 0x11123456.

1. Give the logical address format

0b 0001 0001 0001 0010 0011 0100 0101 0110

1. Give the physical address format
2. Give the displacement

8.31 Consider a demand-paging system with a paging disk that has an average access and transfer time of 20 msecs. Addresses are translated through a page table in main memory, with an access time of 1 microsecond per memory access. Thus, each memory reference through the page table takes two accesses. To improve this time, we have added an associative memory that reduces access time to one memory reference, if the page-table entry is in the associative memory. Assume that 80% of the accesses are in the associative memory and that, of those remaining, 10% (or 2 % of the total) cause page faults. What is the effective memory access time?

The effective memory access time is:

Practical: Find the page\_check\_references()function (in the memory management routines for scanning through virtual memory). Locate the code that deals with pages that have been referenced (i.e. their page table entry has been accessed). Explain the comment, and relate the described process to a mechanism discussed in class.



So this is just like the second chance algorithm that we talked about in class, where it makes a pass once, and set it to deactivated.