**CHAPTER 4 - Memory**

**Questions 1**

Consider the following page reference string: e, c, b, e, a, g, d, c, e, g, d, a

With 4 frames, how many page faults would occur with the following page replacement algorithms? Fill in the tables accordingly.

RS: reference strings; F0: frame 0, F1: frame 1, etc.

*Hint: all frames are initially empty, so your first unique pages will all cost one fault each.*

1. Optimal

| Time | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RS | e | c | B | e | a | g | d | c | e | g | d | a |
| F0 | e | e | e | e | e | e | e | e | e | e | e | e |
| F1 |  | c | c | c | c | c | c | c | e | e | e | e |
| F2 |  |  | B | B | B | B | B | B | B | B | B | B |
| F3 |  |  |  | e | e | e | e | e | e | g | g | g |
| Page fault? | x | x | x | x | - | - | x | - | x | - | - | - |

Total page fault:6

2. LRU

| Time | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RS | e | c | b | e | a | g | d | c | e | g | d | a |
| F0 | e | e | e | e | e | e | e | c | c | c | c | c |
| F1 |  | c | c | c | c | c | c | c | e | e | e | e |
| F2 |  |  | b | b | b | b | b | b | e | e | e | e |
| F3 |  |  |  | e | e | e | e | e | e | e | e | a |
| Page fault? | x | x | x | x | - | - | x | x | x | - | - | x |

Total page fault:8

3.Second chance

| Time | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RS | e | c | b | e | a | g | d | c | e | g | d | a |
| F0 | e | e | e | e | e | e | d | d | e | e | e | e |
| F1 |  | c | c | c | c | c | c | c | c | g | g | g |
| F2 |  |  | b | b | b | b | b | c | c | c | c | c |
| F3 |  |  |  | e | e | e | e | e | e | e | e | a |
| Page fault? |  |  |  |  |  |  |  |  |  |  |  |  |

Total page fault: 9

**Questions 2**

**1.** True or False? A program does not need to be stored in memory in its entirety.

=>> true

**2.** True or False? A physical address space is at least as large as a virtual address space.

=>>false

3. When does a page fault occur?

=>>A page fault occurs when a process tries to access a page that has not been brought into

memory

=>>A page fault occurs when a process tries to access a page that has not been brought into

memory

=>> a page fault occurs when a process tries access a page that has not been brought into memory.

**3.** True or False? In a pure demand paged system a page is never brought into memory until it is needed.

=>>true

1. A machine has 48 bit virtual addresses and 32 bit physical addresses. Pages are 8 KB. How many entries are needed for the page table?

2^35 pages

=>> With **8-KB** pages and a **48-bit** virtual address space, the number of virtual pages is 2^48/2^13, which is 2^35 (about 34 billion).

1. For each of the following decimal virtual addresses, compute the virtual page number and offset for a 4-KB page and for an 8 KB page: 20000, 32768, 60000.

2.1 Page size = 4KB

20000 = page number \* 4 \* 1024 + offset

| Virtual address | Page number | Offset |
| --- | --- | --- |
| 20000 | 4 | 3616 |
| 32768 | 8 | 0 |
| 60000 | 14 | 2656 |

2.2 Page size = 8KB

20000 = page number \* 8 \* 1024 + offset

| Virtual address | Page number | Offset |
| --- | --- | --- |
| 20000 | 2 | 3616 |
| 32768 | 4 | 0 |
| 60000 | 7 | 2656 |

3. The figure below shows a virtual address space from 0 to 64K and 32K of physical memory. There are 16 pages and 8 frames and transfers between memory and disk are in pages. Give the physical address corresponding to the following virtual addresses, explain how did you get the answer?:

a) 20 b) 4100 c) 8300

=>>a=8212

=>>b=4100

=>>c=24684

4. A memory free in 3 frames. How many page fault occur after running as the following page 7, 0 , 1, 2 , 0, 3, 0 , 4, 2 , 3 , 0 , 3 , 2 , 1, 2, 0, 1, 0 , 7 using FIFO

|  | **7** | **0** | **1** | **2** | **0** | **3** | **0** | **4** | **2** | **3** | **0** | **3** | **2** | **1** | **2** | **0** | **1** | **0** | **7** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| F1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| F2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| F3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| PF |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

2. With given page table and 16 bit virtual address, that is split 4 bit page numbers and 12 bit offset. If user references the address 15016, which physical address is in memory?

Answer: 2728

3. A memory free in 4 frames. How many page faults do occur after running as the following page 2 3 2 0 1 5 2 4 5 3 2 5 2 using LRU

Answer: 8

|  | **2** | **3** | **2** | **0** | **1** | **5** | **2** | **4** | **5** | **3** | **2** | **5** | **2** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| F1 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| F2 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| F3 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| F4 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| PF |  |  |  |  |  |  |  |  |  |  |  |  |  |

4. A memory free in 4 frames. Which state of the memory after the page 4 is accessed when the requested page as 2 3 2 0 1 5 2 4 5 3 2 5 2 using LRU

=>> 7

6. Assume that the Page Table below is in effect. The number of lines per page is 400. The actual memory location for line 1634 is \_\_\_\_\_34\_\_ .

| Page Number | Page Frame Number |
| --- | --- |
| 0  1  2  3  4 | 8  10  5  11  0 |

7. A computer has four page frames. The time of loading, time of last access, and the R and M bits for each page are as shown below (the times are in clock ticks):

| Page | Loaded | Last ref | R | M |
| --- | --- | --- | --- | --- |
| 0 | 226 | 280 | 0 | 0 |
| 1 | 160 | 265 | 0 | 1 |
| 2 | 110 | 270 | 1 | 0 |
| 3 | 120 | 285 | 1 | 1 |

Which page will LRU replace?

=>> answer:1

8. A computer has four page frames. The time of loading, time of last access, and the R and M bits for each page are as shown below (the times are in clock ticks):

Page Loaded Last ref. R M

------------------------------------------------------------------------------

0 226 280 0 0

1 160 265 0 1

2 110 270 0 0

3 120 285 1 1

Which page will Second Chance replace? (NRU)

=>> answer: Page 0, because it’s RM = 0.

9. A memory free in 3 frames. How many page hits do? Assume that the running as the following page 7, 0 , 1, 2 , 0, 3, 0 , 4, 2 , 3 , 0 , 3 , 2 , 1, 2, 0, 1, 0, 7 using LRU

|  | **7** | **0** | **1** | **2** | **0** | **3** | **0** | **4** | **2** | **3** | **0** | **3** | **2** | **1** | **2** | **0** | **1** | **0** | **7** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| F1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| F2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| F3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| PF |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

9. A computer provides the user with virtual address space of 2^32

(2 to the power 32) bytes. Pages of size 4096 (4K or 2^12) bytes

are used for implementing virtual memory where the total physical

memory is equal to 2^18 bytes. If the hexadecimal virtual

address is 23456111, the page number in hexadecimal would be?

=>>answer: 23456.

10. A computer has four page frames. The time of loading, time of last

access, and the R and M bits for each page are as shown below

(the times are in clock ticks):

Page Loaded Last Ref. R M

0 230 285 1 0

1 120 265 0 0

2 140 270 0 1

3 110 280 1 1

(a) Which page will NRU replace? (1)

=>> answer: Page 0, because it’s RM = 0.

(b) Which page will FIFO replace? (3)

=>>answer: Page 2, because the page 2 is the oldest loaded page.(in the FIFO queue)

(c) Which page will LRU replace? (1)

=>> answer: Page 1, because the page 1’s last reference time is 260, which is the oldest referenced time.

(d) Which page will second chance replace? (1)

=>> answer: Page 0, because the page 0 R bit is 0, and the second-earliest page loaded. The page 2’s R bit is 1, will put at the end of the list by FIFO replace scheduler.

11. Consider a logical address space of 64 pages of 2048 words each, mapped

unto a physical memory of 32 frames.

a) How many bits are there in logical address?

=>>> answer: As a logical address space of 64 pages of 1,024 words each, mapped onto a physical memory of 32 frames.

So the bits that are there in the logical address are -

Logical address space has 8 pages, size of each page, offset = 1024 words

Number of bits in page# field of the logical address = bits = 3 bits

Offset bits = = 10 bits

=> Logical address = 3 + 10 = 13 bits

17

b) How many bits are there in physical address?

=>> answer: Physical memory has 32 frames, offset = 1024 words

Number of bits in frame# field of the Physical address = bits = 5 bits

=>Physical address = 5 + 10 = 15 bits

16

12. A system with 32 bit virtual address. If the page size is 4 KB and each table entry occupies 4 bytes, what is the size of the page table?

=> answer: 1MB

Consider a swapping system in which the memory consists of the following hole sizes: 10K, 4K, 20K, 15K, 9K. Assume worst fit algorithm is used. Which holes are taken for successive segment requests of 8K,

=>>answer: 20K, 15K, 10K

13. If there are 64 pages and the page size is 2048 words, what is the length of logical address?

=>> answer: 17bits

14. A system with 32 bit virtual address. If the page size is 4 KB and each table entry occupies 4 bytes, what is the size of the page table?

total pages = 2^20

size of page table = 4B \* 2 ^ 20 = 4MB

size of virtual memory = 2^ 20 \* 4KB = 4GB