

# **TSN2101/TOS2111 OPERATING SYSTEMS**

## **Tutorial 8 (Virtual-Memory Management) - Solutions**

1. Consider the page table for a system with 12-bit virtual and physical addresses with 256-byte pages. The list of free page frames is D, E, F (D is at the head of the list, E is second, and F is last)

Page	Frame
0	-
1	2
2	C
3	A
4	-
5	4
6	3
7	-
8	B
9	0

Convert the following virtual addresses to their equivalent physical addresses in hexadecimal. All numbers are given in hexadecimal. A dash for frame indicates that the page is not in memory. If the frame is not in memory, it will be brought to one of the free page frames in order.

- (i) 9EF
- (ii) 111
- (iii) 700
- (iv) OFF

### **Solution:**

- (i) 0EF
- (ii) 211
- (iii) D00
- (iv) EFF

2. Consider the following page table in a demand paging system. Assume the page size of 2000 bytes. Assume the usage of decimal values.

Page	Frame	Valid-Invalid bit
page0	5	v
page1	22	i
page2	300	v
page3	150	v
page4	30	i
page5	50	i
page6	120	v
page7	101	v

Check whether the virtual addresses given below would generate a page fault? If it does not generate page fault, identify the physical address.

- (i) 10230
- (ii) 5213
- (iii) 100
- (iv) 8125

**Solution:**

- (i) Virtual address 10230:  
Offset 230 in page 5 (10230/2000).  
Page 5 contains invalid bit (i). This results in page fault
- (ii) Virtual address 5213:  
Offset 1213 in page 2 (5213/2000).  
Page 2 contains valid bit (v) and the corresponding frame no. is 300.  
Physical address = Starting address of the frame + Offset  
 $= (300 \times 2000) + 1213 = 601213$
- (iii) Virtual address 100:  
Offset 100 in page 0 (100/2000).  
Page 0 contains valid bit (v) and the corresponding frame no. is 5.  
Physical address = Starting address of the frame + Offset  
 $= (5 \times 2000) + 100 = 10100$
- (iv) Virtual address 8125:  
Offset 125 in page 4 (8125/2000).  
Page 4 contains invalid bit (i). This results in page fault.

3. On a system using demand-paged memory, it takes 200ns to satisfy a memory request if the page is in memory. If the page is not in memory, the request takes 7ms if a free frame is available or the page to be swapped out has not been modified. It takes 15ms if the page to be swapped out has been modified. What is the effective access time if the page fault rate is 5%, and 60% of the time the page to be replaced has been modified/free frame is not available?

**Solution:**

Given: Page fault rate is 5%  
Hence the page is in memory 95% of time.  
It requires 200ns or 0.2microseconds 95% of time.  
Given: Of the remaining 5% that result in page fault, 60% of the time the page to be replaced has been modified/free frame is not available.  
Hence  $60\% \text{ of } 5 = 3\%$  of total access takes 15milliseconds or 15000 microseconds.  
 $40\% \text{ of } 5 = 2\%$  of total access takes 7milliseconds or 7000 microseconds.  
Effective access time =  $(0.95 \times 0.2) + (0.02 \times 7000) + (0.03 \times 15000) = 590.19$  microseconds.

4. Consider a demand-paging system with a paging disk that has an average access and transfer time of 20 milliseconds. 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, an associative memory is added that reduces access time to one memory reference if the page-table entry is in the associative memory.

Assume that 80 percent of the accesses are in the associative memory and 10 percent cause page faults. What is the effective memory access time?

**Solution:**

Effective Memory Access Time

$$\begin{aligned} &= (0.8) \times (1 \text{ microsecond}) + (0.1) \times (2 \text{ microseconds}) + (0.1) \times (20003 \text{ microseconds}) \\ &= 0.8 + 0.2 + 2000.3 = 2001.3 \text{ microseconds or } 2.0013 \text{ milliseconds.} \end{aligned}$$

5. 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

Assume the availability of FOUR frames and all the frames are initially empty. How many page faults would occur for First in First out (FIFO) page replacement algorithm? Identify the final pages in the frames.

**Solution:**

- (i) First In First Out (FIFO) page replacement algorithm:

Number of available frames: 4

Before	Page	Fault	After
-,-,-,-	1	Yes (1)	1,-,-,-
1,-,-,-	2	Yes (2)	2,1,-,-
2,1,-,-	3	Yes(3)	3,2,1,-
3,2,1,-	4	Yes (4)	4,3,2,1
4,3,2,1	2	No	4,3,2,1
4,3,2,1	1	No	4,3,2,1
4,3,2,1	5	Yes(5)	5,4,3,2
5,4,3,2	6	Yes (6)	6,5,4,3
6,5,4,3	2	Yes(7)	2,6,5,4
2,6,5,4	1	Yes(8)	1,2,6,5
1,2,6,5	2	No	1,2,6,5
1,2,6,5	3	Yes (9)	3,1,2,6
3,1,2,6	7	Yes(10)	7,3,1,2
7,3,1,2	6	Yes(11)	6,7,3,1
6,7,3,1	3	No	6,7,3,1
6,7,3,1	2	Yes(12)	2,6,7,3
2,6,7,3	1	Yes(13)	1,2,6,7
1,2,6,7	2	No	1,2,6,7
1,2,6,7	3	Yes (14)	3,1,2,6
3,1,2,6	6	No	3,1,2,6

Number of page faults: 14

Final Pages in the frames: 3,1,2,6

6. 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  
 Assume the availability of FOUR frames and all the frames are initially empty. How many page faults would occur for Least Recently Used (LRU) page replacement algorithm? Identify the final pages in the frames.

**LRU page replacement algorithm:**

Number of available frames: 4

Reference string:

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

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	6			6			
2	2	2	2			2	2					2	2	2			2			
	3	3	3			5	5					3	3	3			3			
		4				4	6					6	7	7			1			
y	y	y	y	n	n	y	y	n	n	n	y	y	y	n	n	y	n	n	n	

y- Page fault

n- No page fault

Number of Page Faults: 10

Final Pages in the frames: 6,2,3,1

7. 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

Assume the availability of FOUR frames and all the frames are initially empty. How many page faults would occur for optimal page replacement algorithm? Identify the final pages in the frames.

**Optimal Page replacement algorithm:**

Number of available frames: 4

Reference string:

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

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					7				1			
2	2	2	2			2	2					2				2			
	3	3				3	3					3				3			
		4				5	6					6				6			
y	y	y	y	n	n	y	y	n	n	n	y	n	n	n	y	n	n	n	

y- Page fault

n- No page fault

Number of Page faults: 8

Final Pages in the frames: 1,2,3,6

8. Given a system with four page frames, the following table indicates page number, load time, last reference time, dirty bit, and reference bit.

Page number	Load Time	Last Reference Time	Dirty Bit	Reference bit
0	167	374	1	1
1	321	321	0	0
2	254	306	1	0
3	154	331	0	1

Identify the victim (page to be replaced) for the following algorithms.

- (i) First-In-First-Out (FIFO)
- (ii) Least-Recently-Used (LRU)
- (iii) Reference bit – Dirty bit combination
- (iv) Second Chance

#### Solution:

- (i) Page 3 (Selects the page that is loaded first)
- (ii) Page 2 (Selects the page that is referenced least recently)
- (iii) Page 1 (class with no modification (Dirty bit:0) and not referenced recently (Reference bit:0))
- (iv) Page 2 (Second chance first checks the oldest page, page 3. Since its reference bit is 1, it resets the bit to 0 and sets the entry's load time to the current time. Next it checks the next oldest page, which is page 0. Since its reference bit is 1, it resets the bit to 0 and sets the entry's load time to the current time. Then it checks the next oldest page, which is page 2. Since its reference bit is 0, it is selected)

9. Assume that a computer system has four frames and uses six reference bits to implement additional-reference bits (aging) page-replacement algorithm. At the first clock tick, the reference bits for pages 0 through 3 are 1,0,1, and 1 respectively. Values at subsequent ticks are 0011,1001,0110,1100, and 0001.

- (i) After the last tick, what is the value of the reference bits for all four pages?
- (ii) Identify the victim page for page replacement.

#### Solution:

- (i)

Page number	Value of reference bits after Tick 6	
0	010101	(21)
1	011000	(24)
2	001011	(11)
3	100111	(39)

- (ii) Victim page – Page with lowest unsigned integer value – Page 2

### **References:**

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, "Operating System Concepts", 9/E, John Wiley & Sons, 2013.
2. William Stallings, "Operating Systems: Internals and Design Principles", 7<sup>th</sup> Edition, Prentice Hall, 2011
3. J. Archer Harris, "Operating Systems", McGraw-Hill, 2002.