

Câu 1:

a. Number of frame $2^5 = 32 \Rightarrow 5$ bits for storing f

Frame size = 1KB = 2^{10} bytes $\Rightarrow 2^{10}$ offset $\Rightarrow 10$ bits for storing d

- $10 + 5 = 15$ bit for physical address

b. The size of the virtual memory space of the program P: 8 KB

c. The size of the physical memory: 32 KB

Câu 2:

a. The number of frames in the physical memory: $320 \text{ MB} / 8\text{KB} = 40 \cdot 2^{10}$

b. 1 page: 8 KB = 2^{10} byte $\Rightarrow 2^{10}$ offset $\Rightarrow 10$ bit offset $\Rightarrow 48 - 10 = 35$ bit page

The maximum number of pages in logical address spaces: 2^{35}

c. Each page have $2^{13} = 8192$ offset

- $1892 = \langle 0, 1892 \rangle$
 $p = 1892 / 8192 = 0, d = 1892 \% 8192 = 1892$
- $15296 = \langle 1, 7014 \rangle$
 $p = 15296 / 8192 = 1, d = 15296 \% 8192 = 7014$
- $20300 = \langle 2, 3916 \rangle$
 $p = 20300 / 8192 = 2, d = 20300 \% 8192 = 3916$

Câu 3:

a. $d = 32 - 9 - 11 = 12 \Rightarrow 2^{12}$ offsets/page

Word-addressable memory $\Rightarrow 1$ offset point to a word 4 bytes

$$\Rightarrow \text{Size of page} = 2^{12} * 4 = 2^{14} \text{ byte}$$

b. Number of frame = $10 \text{ GB} / 2^{14} \text{ byte} = 10 \cdot (2^{16})$ frames

c. The maximum size of process space supported in this system:

$$2^{32} * 4 = 2^{34} \text{ byte}$$

d. $3\text{GB} / 2^{14} \text{ byte} = 3 \cdot (2^{16}) = 196608 \Rightarrow \text{integer} \Rightarrow \text{no fragmentation}$

e. Internal fragmentation

Câu 4:

$$h = 75 \% ; t_m = 132 \text{ ns} ; t_c = 25 \text{ ns}$$

$$EAT = 0.75 \cdot (25 + 132) + (1 - 0.75) \cdot (25 + 2 \cdot 132) = 190 \text{ ns}$$

Câu 5:

$$t_c = 25\text{ns}; t_m = 150\text{ns}; EAT = 225 \text{ ns}$$

$$225 = h \cdot (25 + 150) + (1 - h) \cdot (25 + 2 \cdot 150) \Rightarrow h = 0.67 = 67\%$$

Câu 6:

$$EAT = h \cdot (t_c + t_m) + (1 - h) \cdot [(1 - p) \cdot (t_c + 2 \cdot t_m) + p \cdot t_p]$$

$$= 0.75 \cdot (15 + 100) + (1 - 0.75) \cdot [(1 - 0.0005) \cdot (15 + 2 \cdot 100) + 0.0005 \cdot 150000]$$

$$= 158.72 \text{ ns}$$

Câu 7:

FIFO

	9	4	8	1	4	9	2	9	4	5	4	5	3	4	6	4	3	7	4	9	4
F1	9	9	9	1	1	1	1	1	4	4	4	4	4	4	6	6	6	6	6	9	9
F2		4	4	4	4	9	9	9	9	5	5	5	5	5	5	4	4	4	4	4	4
F3			8	8	8	8	2	2	2	2	2	2	3	3	3	3	3	7	7	7	7
	*	*	*	*		*	*		*	*			*		*	*		*		*	

No. of page fault: 13, page fault hit ratio = $13/21 = 61.9\%$

$$EAT = (1 - 0.619) \cdot 102 + 0.619 \cdot 192 = 165.9 \text{ ns}$$

Optimal

	9	4	8	1	4	9	2	9	4	5	4	5	3	4	6	4	3	7	4	9	4
F1	9	9	9	9	9	9	9	9	9	9	9	9	3	3	3	3	3	3	3	9	9
F2		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
F3			8	1	1	1	2	2	2	5	5	5	5	5	6	6	6	7	7	7	7
	*	*	*	*			*			*			*		*			*		*	

No. of page fault: 10, page fault hit ratio = $10/21 = 47.6\%$

$$EAT = (1 - 0.476) \cdot 102 + 0.476 \cdot 192 = 144.8 \text{ ns}$$

LRU

	9	4	8	1	4	9	2	9	4	5	4	5	3	4	6	4	3	7	4	9	4
F1	9	9	9	1	1	1	2	2	2	5	5	5	5	5	6	6	6	7	7	7	7
F2		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
F3			8	8	8	9	9	9	9	9	9	9	3	3	3	3	3	3	3	9	9
	*	*	*	*		*	*			*			*		*			*		*	

No. of page fault: 11, page fault hit ratio = $11/21 = 52.4\%$

$$EAT = (1 - 0.524) \cdot 102 + 0.524 \cdot 192 = 149.2 \text{ ns}$$

Second chance:

	9	4	8	1	4	9	2	9	4	5	4	5	3	4	6	4	3	7	4	9	4
F1	9	9	9	1	1	1	1	1	4	4	4	4	4	4	4	4	4	7	7	7	7
F2		4	4	4	4	4	2	2	2	2	2	2	3	3	3	3	3	3	4	4	4
F3			8	8	8	9	9	9	9	5	5	5	5	5	6	6	6	6	6	9	9
FIFO	9	9	9	4	4	1	9	9	2	2	2	2	4	4	3	4	4	3	6	7	7
		4	4	8	8	4	1	1	9	4	4	4	5	3	4	3	3	6	7	4	4
			8	1	1	9	2	2	4	5	5	5	3	5	6	6	6	7	4	9	9
Refer	9	9	9	1	1	1	9	9	2	4	4	4	5	5	6	6	6	7	7	7	7
ence		4	4		4	9	2	2	4	5	5	5	3	3		4	4		4	4	4
d bit			8											4			3			9	9
	*	*	*			*	*		*	*			*		*			*	*	*	

No. of page fault: 13, page fault hit ratio = $13/21 = 61.9\%$

$$EAT = (1 - 0.619) \cdot 102 + 0.619 \cdot 192 = 157.7 \text{ ns}$$

b. best solution

Optimal cho ra con số nhỏ nhất tuy nhiên trong thực tế rất khó để dự đoán tương lai nó có cần truy cập nữa hay không. Nên phương pháp LRU sẽ là phương pháp hiệu quả và thực tế hơn optimal.

Câu 8:

a. The number of frame: $512 \text{ MB} / 4096 \text{ byte} = 2^{17}$

b. $4096 = 2^{12} \Rightarrow d = 12 \text{ offset} \Rightarrow p = 32 - 12 = 20$

The maximum number of pages = 2^{20}

c.

• **P2 – address 13000**

$$\Rightarrow p = 13000 / 4096 = 3$$

$$\Rightarrow d = 13000 \% 4096 = 712$$

trang 3 đã được nạp vào khung trang #400 \Rightarrow truy cập bình thường

- **P1 – address 13000**

$$\Rightarrow p = 13000 / 4096 = 3$$

$$\Rightarrow d = 13000 \% 4096 = 712$$

Trang 3 của P1 chưa được đưa vào bộ nhớ

\Rightarrow Dẫn đến page fault

Còn frame #0 trống nên nạp frame #0 vào Page 3 của P1 đồng thời bật valid bit lên 1.

Đến đây được truy cập bình thường

Page	Frame	Valid bit
0	#300	0
1		1
2	#301	0
3	#0	1

P1

Page	Frame	Valid bit
0	#300	1
1		0
2	#301	1
3	#400	1

P2

Page	Frame	Valid bit
0		0
1		0
2		0
3		0

P3

- **P1 – address 16383**

$$\Rightarrow p = 16383 / 4096 = 3$$

$$\Rightarrow d = 16383 \% 4096 = 4095$$

Trang 3 của P1 đã được nạp vào khung trang #0 ở câu trên \Rightarrow truy cập bình thường.

- **P3 – address 4096**

$$\Rightarrow p = 4096 / 4096 = 1$$

$$\Rightarrow d = 4096 \% 4096 = 0$$

Trang 1 của P3 chưa được nạp vào nhưng hiện tại đang hết frame trống \Rightarrow page fault

Theo đề bài ta sẽ sử dụng LRU và không được phép lấy lại Page 0 và 2 của P1, P2 \Rightarrow ta sẽ lấy frame #400 của P2 lại (theo thuật toán LRU).

Page	Frame	Valid bit
0	#300	0
1		1
2	#301	0
3	#0	1

P1

Page	Frame	Valid bit
0	#300	1
1		0
2	#301	1
3		1

P2

Page	Frame	Valid bit
0		0
1	#400	1
2		0
3		0

P3

Lúc này page 1 của P3 đã được nạp vào frame #400 => Truy cập bình thường.

- **P3 – address 13000**

$$\Rightarrow p = 13000 / 4096 = 3$$

$$\Rightarrow d = 13000 \% 4096 = 712$$

Tương tự như câu trên, trang 3 của P3 chưa được nạp vào nhưng hiện tại đang hết frame trống => page fault

Theo thuật toán LRU ta lấy lại frame #0 ở trang 3 của P1

Page	Frame	Valid bit
0	#300	0
1		1
2	#301	0
3		0

P1

Page	Frame	Valid bit
0	#300	1
1		0
2	#301	1
3		1

P2

Page	Frame	Valid bit
0		0
1	#400	1
2		0
3	#0	1

P3

Lúc này page 3 của P3 đã được nạp vào frame #0 => Truy cập bình thường.

- **P2 – address 16383**

⇒ $p = 16383 / 4096 = 3$

⇒ $d = 16383 \% 4096 = 4095$

Trang 3 của P2 chưa được nạp vào nhưng hiện tại đang hết frame trống => page fault

Theo đề bài ta sẽ sử dụng LRU=> lấy frame #400 của trang 1 P3 để đưa vào trang 3 của P2.

Page	Frame	Valid bit
0	#300	0
1		1
2	#301	0
3		0

P1

Page	Frame	Valid bit
0	#300	1
1		0
2	#301	1
3	#400	1

P2

Page	Frame	Valid bit
0		0
1		0
2		0
3	#0	1

P3

Lúc này page 3 của P2 đã được nạp vào frame #400 => Truy cập bình thường.