

## Questão ①

a)

2,0

página  $\rightarrow 1\text{KB} \rightarrow 2^{10}$  (10 bits)

memória física  $\rightarrow 4\text{KB} \rightarrow 2^2 \cdot 2^{10}$  (12 bits)

tabela de página possui 8 entradas  $\rightarrow 2^3$  (3 bits)

end. virtual = 13 bits ✓

$0x832 = \underbrace{01000\ 0011\ 0010}_{(12)}$ , portanto o endereço virtual  $0x832$

p/ identificar a página deslocamento =  $50_{(10)}$  mapeia para a página 2, que mapeia para a moldura 3,

segundo a tabela de páginas.

Temos que o endereço físico = (nº moldura  $\cdot$  tam página) + deslocamento,

logo endereço físico =  $(3 \cdot 1024) + 50 \Rightarrow$  end. físico =  $3072 + 50 = 3122_{(10)}$

end. virtual = 13 bits

$0x14AA = \underbrace{0001\ 0100\ 1010\ 1010}_{(12)}$ , portanto o endereço virtual  $0x14AA$

p/ identificar a página deslocamento =  $170_{(10)}$  mapeia para a página 5, que mapeia para a moldura 2,

segundo a tabela de páginas.

Temos que o endereço físico = (nº moldura  $\cdot$  tam página) + deslocamento,

logo o endereço físico =  $(2 \cdot 1024) + 170 = 2048 + 170 = 2218_{(10)}$

① tamanho dos endereços virtuais é de 13 bits, e os dos endereços físicos 12 bits. ✓

A CPU gera os endereços virtuais, repassa para MMU, que os converte em endereços físicos (isso quando a TLB já não é capaz de informar a tradução do endereço virtual para o físico).



1, 0

b)

	PID	V	R	M	#PÁGINA
0	P	1	1	0	4
1	X	1	r	m	p
2	P	1	1	1	5
3	P	1	0	1	2

↑  
tabela de pág.  
invertida

↑  
memória  
física

↑  
nº de  
molduras  
mapeadas

↑  
nº de  
molduras

A tabela de páginas invertida possui número de páginas. Igual ao número de molduras da memória física.

Supondo que a tabela de páginas dada é de um processo arbitrário P, a tabela de páginas invertida é a mostrada acima. Devemos ter um campo para o pid pois a tabela de páginas invertida é única para todos os processos. (compartilhada entre eles)

Como a tabela de páginas dada possui apenas 3 páginas efetivamente mapeadas na memória física (bit V igual a 1) então a moldura 1 (que é e não utilizada pelo processo P) deve ser ocupada (bit V igual a 1) pela página p de um outro processo arbitrário X com bits R=r e M=m arbitrários.

c) Para essa tabela, poderia ser usado o algoritmo de NRU. Vamos considerar apenas as páginas com o bit de validade 1: 2, 4, 5

Page	REF Bit	MODIFY BIT	PAGE FRAME
2	0	1	3 → classe 1
4	1	0	0 → classe 2
5	1	1	2 → classe 3

→ Menores classes possuem maior prioridade para sair da memória, portanto:

Página 2	→	Moldura 3	→	Prioridade	0
4	→	0	→		1
5	→	2	→		2