

# SISTEMAS OPERATIVOS

## CERTAMEN #2

### Pauta de corrección

Wenceslao Palma <wenceslao.palma@pucv.cl>

1. Responda las sgtes preguntas. Justifique cada una de sus respuestas.

- (a) (10 ptos.) En un sistema operativo que utiliza gestión de memoria basada en paginación cada página tiene un tamaño de 2048 bytes. La memoria física disponible para los procesos es de 8 MB. Si llega un proceso que requiere 31566 bytes y después llega otro proceso que requiere 18432 bytes. Cuantifique la fragmentación interna/externa que provoca cada proceso. R.:

#páginas de la memoria =  $8M/2KB = (1024 \times 1024 \times 8) / (2 \times 1024) = 4096$  págs

PROCESO 1

#páginas del proceso 1 =  $31566 \text{ (bytes)} / 2048 \text{ (bytes)} = 15,4$  págs

----> el proceso 1 necesita 16 págs

el proceso 1 puede ser almacenado en la memoria ya que  $16 < 4096$ .

fragmentación interna del proceso 1 =  $(16 \times 2048) \text{ (bytes)} - 31566 \text{ (bytes)}$   
= 1202 (bytes) (4 ptos)

PROCESO 2

#páginas del proceso 2 =  $18432 \text{ (bytes)} / 2048 \text{ (bytes)} = 9$  págs

----> No hay fragmentación interna (4 ptos)

Cuando la memoria es paginada no existe fragmentación externa. (2 ptos)

- (b) (5 ptos.) Considere un espacio de direcciones lógicas paginado, compuesto de 32 páginas de 2KB, correspondiente a un espacio de memoria física de 1MB. Cuál es el formato de las direcciones lógicas de los procesos? Cuál es el efecto sobre la tabla de páginas si el espacio de memoria física se reduce a la mitad?

R.: Si tenemos 32 páginas se necesitan 5 bits para representarlas. Luego el máximo desplazamiento dentro de una página equivale a  $2^{11}$  bytes (2KB). Por lo tanto la dirección lógica es de 16 bits compuesta de 5 bits para la página y 11 bits para el desplazamiento. Ya que se tiene un espacio de 1MB para direccionar, cada entrada en la tabla de páginas necesita 9 bits para determinar la dirección del frame. Si la memoria se reduce a la mitad se necesitarán 8 bits para la dirección del frame.

(c) (5 pts.) Por qué el cálculo de direcciones reales a partir de direcciones lógicas es diferente en la paginación y la segmentación?

R.: Es diferente debido a la forma en la cual se divide la memoria. En la paginación, donde la memoria se divide en partes iguales la obtención de la dirección real se obtiene usando una simple concatenación. En cambio, en la segmentación es necesario conocer el inicio del segmento, verificar que el desplazamiento sea consistente con la longitud del desplazamiento y finalmente la dirección real será la suma de la dirección real del inicio del segmento más el desplazamiento.

2. (25 pts.) Considere un sistema con memoria virtual donde el conjunto residente es de tamaño 3. Muestre el funcionamiento de los algoritmos de reemplazo de página OPT, FIFO, LRU y CLOCK para la siguiente secuencia de referencias a páginas:

4 2 4 1 6 3 2 5 6 4 1 3 5 3

Comente.

OPT (6 pts.)

4	2	4	1	6	3	2	5	6	4	1	3	5	3
4	4	4	4	4	3	3	3	3	3	3	3	3	3
	2	2	2	2	2	2	5	5	5	5	5	5	5
			1	6	6	6	6	6	4	1	1	1	1
				F	F		F		F	F			

FIFO (6 pts.)

4	2	4	1	6	3	2	5	6	4	1	3	5	3
4	4	4	4	6	6	6	5	5	5	1	1	1	1
	2	2	2	2	3	3	3	6	6	6	3	3	3
			1	1	1	2	2	2	4	4	4	5	5
				F	F	F	F	F	F	F	F	F	F

LRU (6 pts.)

4	2	4	1	6	3	2	5	6	4	1	3	5	3
4	4	4	4	4	3	3	3	6	6	6	3	3	3
	2	2	2	6	6	6	5	5	5	1	1	1	1
			1	1	1	2	2	2	4	4	4	5	5
				F	F	F	F	F	F	F	F	F	F

CLOCK (7 pts.)

4	2	4	1	6	3	2	5	6	4	1	3	5	3
*4-1	*4-1	*4-1	*4-1	6-1	6-1	*6-1	5-1	5-1	*5-1	1-1	1-1	*1-1	*1-1
	2-1	2-1	2-1	*2-0	3-1	3-1	*3-0	6-1	6-1	*6-0	3-1	3-1	3-1
			1-1	1-0	*1-0	2-1	2-0	*2-0	4-1	4-0	*4-0	5-1	5-1
				F	F	F	F	F	F	F	F	F	F

3. (15 ptos.) En el siguiente programa C, cuantas copias de la variable c existen? Cuáles son los valores de cada copia antes del fin del programa? justifique.

```
int main(int argc, char ** argv){
    int child = fork();
    int c = 10;

    if(child == 0){
        c += 5;
    }else{
        child = fork();
        c += 10;
        if(child)
            c += 5;
    }

    return 0;
}
```

En el programa se tiene 3 procesos, un padre y dos hijos, por lo tanto existen 3 copias de la variable c (3 ptos). En el padre el valor de c es 25 (4 ptos). En el 1er hijo, creado con el 1er fork(), el valor de c es 15 (4 ptos). Y en el 2do hijo el valor de c es 20 (4 ptos).