

**Sistemas Operativos,  
Pauta Certamen #3, 02/2020  
Santiago, 11.01.2020**

1. Conteste brevemente las siguientes preguntas:

a) ¿Cuál es la diferencia entre evadir y prevenir un deadlock?

**Respuesta:** La prevención bloquea una o más de las condiciones necesarias para la ocurrencia del deadlock mientras que la evasión no lo hace pero analiza todas las posibles asignaciones de recursos de tal forma de que nunca se llegue a un deadlock.

b) Indique las tareas que se deben ejecutar ante una falla de página

**Respuesta:**

- Trap al SO.
- Buscar página en disco.
- Cargar página en disco.
- Actualizar valores en tabla de páginas.
- Reiniciar ejecución de instrucción.

c) ¿Cómo se puede detectar si un sistema está en Thrashing? ¿Cómo se puede solucionar?

**Respuesta:** Se puede detectar cuando disminuye la utilización de CPU y aumenta la cantidad de fallos de página. Se puede disminuir el problema bajando el grado de multiprogramación y/o utilizando políticas de reemplazo local.

d) Indique 4 requisitos que debe tener un SO para implementar paginación bajo demanda.

**Respuesta:**

- Reiniciar una instrucción después de un fallo de página.
- Mantener una lista con los frames disponibles.
- Agregar un bit de Vo I a la tabla de páginas.
- Contar con memoria secundaria

e) Considere un SO que tiene  $n$  instancias de un recurso compartidas entre  $m$  ( $n-1$ ) procesos. Cada proceso necesita con máximo dos recursos. ¿El sistema está en deadlock?

**Respuesta:** Si el SO estuviese en deadlock, cada proceso tendría asignada una instancia y estaría a la espera de otra. Ya que hay una instancia libre, cualquiera de los procesos puede obtenerla, finalizar y liberar, por lo que el sistema no está en deadlock.

f) ¿Qué es una TLB? ¿Por qué se requiere en paginación?

**Respuesta:** Es una memoria caché implementada en el procesador que almacena las últimas referencias a páginas realizadas con el objetivo de evitar acceder dos veces a memoria principal por cada referencia.

2. Considere la siguiente instantánea de un sistema:

Procesos	Allocation	Request	Available
	ABCD	ABCD	ABCD
P1	0010	2001	2100
P2	2001	1010	
P3	0120	2100	

Determine si el sistema no está en deadlock. Si no lo está, indique la secuencia de ejecución correspondiente.

**Respuesta:** Corremos el algoritmo:

1. Finish = (F,F,F)  
Work = (2,1,0,0)
  2. R3 = (2,1,0,0) <= (2,1,0,0) OK → P3  
Finish = (F,F,T) y W = (2,1,0,0) + (0,1,2,0) = (2,2,2,0)  
R2 = (1,0,1,0) <= (2,2,2,0) OK → P2  
Finish = (F,T,T) y W = (2,2,2,0) + (2,0,0,1) = (4,2,2,1)  
R1 = (2,0,0,1) <= (4,2,2,1) OK → P1  
Finish = (T,T,T)...
- El sistema no está en deadlock. El orden de ejecución de los procesos <P3,P2,P1>.

3. Un SO gestiona su memoria principal utilizando paginación. Las direcciones lógicas y físicas utilizan 32 bits. Si el tamaño de la página es de 4KB y se sabe que se tienen 4096 frames, se pide:

- a) ¿Cuál es el tamaño de la memoria principal?  
**Respuesta:**  $4096 * 2^{12}B$
- b) ¿Cuál es el tamaño del espacio virtual?  
**Respuesta:**  $2^{32}$ . Esto debido a que se tienen 32 bits para direcciones lógicas
- c) ¿Número máximo de páginas por proceso?  
**Respuesta:** Para el offset se requieren 12 bits, por lo que se tienen 20 bits para las páginas. El número máximo es de  $2^{20}$ .
- d) ¿Cuántas páginas requiere un proceso de 45KB?  
**Respuesta:**  $45KB/4Kb = 12$  páginas.
- e) En la pregunta (d) ¿Se produce fragmentación? ¿De cuánto?  
**Respuesta:** Se producen 3KB de fragmentación interna.
- f) Si se agregan 5 bits a cada entrada de la tabla de páginas. ¿Cuál es el tamaño máximo que puede tener una tabla de páginas?  
**Respuesta:**  $2^{20} * (20+5)$  bits.

4. Un SO dispone de 4 MB en memoria principal y los gestiona a través de **paginación bajo demanda**. Las direcciones lógicas tienen un tamaño de 20 bits y se tiene un total de 1024 frames, de los cuales solo se asignan 3 a un proceso que genera la siguiente secuencia de direcciones: 0x1A001, 0x2BB02, 0x3C5BA, 0x4F000, 0x2B644, 0x1A4BC, 0x5800B, 0x6ACCC, 0x2B440, 0x1A4CC, 0x2BC04, 0x3C4C0, 0x7A5BC, 0x6A5F0, 0x3CFFC, 0x2BCC0, 0x1A004, **u, v y w**.

a) Para un algoritmo de reemplazo de páginas FIFO, determine los posibles valores que podrían tomar las referencias u, v y w para qué se generen:

- Más de 10 fallas.
- Exactamente 15 fallas, en donde la última es generada por v.
- Más de 15 fallas.

b) Asuma que u, v y w son eliminadas. Para LRU calcular M (3,11).

c) Determine el número de fallas que se generan para el algoritmo Óptimo ¿Tiene algún efecto haber eliminado u, v y w?

**Respuesta:**  $4MB/1024 = 2^{22}/2^{10} = 2^{12}$  por lo que el offset utiliza 12 bits. Dado esto, se requieren 8 bits (2 dígitos hexadecimales) para direccionar páginas. La cadena se convierte en:

**1A, 2B, 3C, 4F, 2B, 1A, 58, 6A, 2B, 1A, 2B, 3C, 7A, 6A, 3C, 2B, 1A, u, v, w**

**FIFO**

1A	2B	3C	4F	2B	1A	58	6A	2B	1A	2B	3C	7A	6A	3C	2B	1A	U	V	W
1A	1A	1A	4F		4F	4F	6A	6A	6A		3C	3C	3C		2B	2B			
	2B	2B	2B		1A	1A	1A	2B	2B		2B	7A	7A		7A	1A			
		3C	3C		3C	58	58	58	1A		1A	1A	6A		6A	6A			
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>		<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>		<b>10</b>	<b>11</b>	<b>12</b>		<b>13</b>	<b>14</b>			

- Sin importar los valores de u, v y w siempre habrán más de 10 fallas.
- v genera la quinceava falla.  $u = \{2B, 1A, 6A\}$ ,  $v = \{3C, 4F, 58, 7A\}$  y  $w = \{v, 1A, 2B\}$
- Como no hay restricción la quinceava falla la puede generar u, por lo que  $u = \{3C, 4F, 58, 7A\}$ , luego v y w pueden tomar cualquier otro valor que no tenga frames asignados para seguir generando fallas.

**LRU**

1A	2B	3C	4F	2B	1A	58	6A	2B	1A	2B	3C	7A	6A	3C	2B	1A
1A	1A	1A	4F		4F	58	58	58	1A		1A					
	2B	2B	2B		2B	2B	6A	6A	6A		3C					
		3C	3C		1A	1A	1A	2B	2B		2B					
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>		<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>		<b>10</b>	...				

$M(3,11)=\{1A, 6A, 2B\}$

## OPTIMO

1A	2B	3C	4F	2B	1A	58	6A	2B	1A	2B	3C	7A	6A	3C	2B	1A
1A	1A	1A	1A			1A	1A				3C	3C			2B*	2B*
	2B	2B	2B			2B	2B				2B	7A			7A	1A*
		3C	4F			58	6A				6A	6A			6A	6A
1	2	3	4			5	6				7	8			9	10

Se generan 10 fallas. Haber eliminado u, v y w si genera un efecto ya que al mirar hacia el futuro estas referencias podrían haber definido el comportamiento del algoritmo al momento de reemplazar las páginas marcadas con \*.