SISTEMAS OPERATIVOS
Guia de Ejercicios

Wenceslao Palma, Sebastián Rodriguez

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

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

   (b) En un sistema que administra su memoria mediante paginación se puede direccionar como máximo $1GB$ de memoria y el tamaño de página es de $16KB$. Determine el formato de la dirección lógica.

   (c) Considere un sistema con memoria paginada donde cada página tiene un tamaño de $2KB$, la tabla de páginas tiene un tamaño de $28KB$ y cada entrada ocupa 16 bits. Determine el máximo espacio de direccionamiento lógico.

   (d) 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 $8MB$. 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.

   (e) 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?

   (f) Qué es la hiperpaginación?.

   (g) Cómo determina un estado inseguro el Algoritmo del Banquero?

   (h) Un computador tiene una memoria caché, una memoria principal y un disco usado para memoria virtual. Si una palabra referenciada está en memoria caché, se necesitan 20ns para acceder a ella. Si está en memoria principal pero no en memoria caché, se necesitan 60ns para cargarla allí, y la referencia comienza de nuevo. Si la palabra no está en memoria principal se necesitan 12ms para cargarla desde disco, seguidos de 60ns para copiarla en memoria caché , y la referencia comienza de nuevo. La tasa de aciertos en la memoria caché es de 0.9 y la tasa de aciertos de la memoria principal es de 0.6 . Cuál es el tiempo, en ns, necesario para acceder a una palabra referenciada en este sistema?

   (i) Cuál es el número mínimo de procesos y recursos necesarios para que se produzca deadlock?

2. Algoritmos de reemplazo de páginas.

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

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

   Muestre si es aconsejable aumentar a 4 el tamaño del conjunto residente. Justifique.

   (b) 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.

   (c) 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:

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

   Comente.

3. Creación de procesos, semáforos.

   (a) Considere 2 procesos, uno de los cuales ejecuta **escribirA** y el otro **escribirB**. Indique los valores iniciales de los semáforos **sA** y **sB** para que la salida generada sea **BABABABABA**. Fundamente su respuesta.

```
void *escribirA () {
  int i; i
  for (i= 0; i< 5; i++) {
    wait(sA);
    printf ("A");
    sleep(random()%2);
    signal(sB);
  }
}
```

```
void *escribirB () {
 int i;
 for (i= 0;i< 5; i++) {
   wait(sB);
   printf ("B");
   sleep(random()%2);
   signal(sA);
 }
}
```

   (b) 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{
          c(5 ptos)hild = fork();
          c += 10;
          if(child)
             c += 5;
       }

       return 0;
 }
```

   (c) Considere la siguiente situación: *Cuando un alumno llega al comedor de la facultad, el cual cuenta con S sillas y B bandejas, busca una silla libre. Si no encuentra una silla libre, se va. En caso contrario, pone su mochila en la silla y va a buscar una bandeja. Si no hay bandejas libres, espera haciendo una fila hasta que se libere una. Si hay una bandeja libre, se pone en la cola para que le sirvan la comida.*
   Realice un análisis de la situación incluyendo una descripción de las situaciones a considerar, los procesos a implementar y los semáforos propuestos para controlar los problemas de sincronización.

SOLUCIONES

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

   (a) 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 paginación es necesario conocer el inicio del segmento, verificar que el despalzamiento sea consistente con la logitud del desplazamiento y finalmente la dirección real será la suma de la dirección real del inicio del segmento más el desplazamiento.

   (b) se necesitan 30 bits para direccionar 1GB de memoria, de los cuales 16 bits se utilizan para direccionar una página ya que un proceso puede direccionar como máximo $2^{30}/2^{14} = 2^{16} pags$. Los 14 bits restantes (16KB=2^{14}) se utilizan para direccionar el contenido de cada página.

   (c) $\#entradas = 28KB/2Bytes = \frac{28 \times 2^{10}}{2} = 14 \times 2^{10}$
   espacio de direccionamiento lógico= $14 \times 2^{10} \times 2KB = 28 \times 2^{20} = 28MB$

   (d)
   ```
   #páginas de la memoria =  8M/2KB = (1024x1024x8)/(2x1024) = 4096 págs

   PROCESO 1
   #páginas del proceso 1 = 31566 (bytes)/2048 (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=(16x2048)(bytes)-31566 (bytes)
                                       = 1202 (bytes)

   PROCESO 2
   #páginas del proceso 2 = 18432 (bytes)/2048 (bytes) = 9 págs
                            ----> No hay fragmentación interna


   Cuando la memoria es paginada no existe fragmentación externa.
   ```

   (e) 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.

   (f) Es una situación que se manifiesta cuando se administra la memoria virtual y en la cual el procesador pasa la mayor parte de su tiempo intercambiando páginas.

(g) El algoritmo del banquero, utilizando las matrices de demanda y asignación, y los vectores de recursos y disponibilidad, determina un estado inseguro examinando una posible asignación de recursos que permita que al menos uno de los procesos termine. Con esto, los recursos del proceso que terminará quedan libres, aumentando la disponibilidad de recursos y la posibilidad que los otros procesos también terminen.

(h) (1) palabra en caché: $(0.9 \times 20)ns$

(2) palabra en memoria pero no en chaché: $(0.6 \times 0.1) \times (60 + 20)$

(3) palabra ni en memoria ni en chaché: $(0.4 \times 0.1) \times (12 \times 10^6 + 60 + 20)$

tiempo = (1)+(2)+(3)

(i) Para que se produzca interbloqueo se necesitan como mínimo dos procesos y dos recursos. De este modo se puede configurar un ciclo de retención y espera que lleva a la aparición de deadlock.

2. Algoritmos de reemplazo de páginas.

(a) LRU

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

CLOCK

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

LRU

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

CLOCK

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

Conviene incrementar a 4 el tamaño del conjunto residente ya que se obtiene en ambos algoritmos un menor número de fallos de página.

(b) OPT

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

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	

LRU

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	

CLOCK

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	

(c) OPT

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

FIFO

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

LRU

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

CLOCK

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

3. Creación de procesos, semáforos.

   (a) Los valores iniciales de los semáforos son sA = 0 y sB = 1. Justificar con ruteo.

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

   (c) **Procesos**: alumno y cocinero. El procesos alumno se encargará de verificar si hay sillas libre, tomar una bandeja y esperar su turno para ser atendido por el cocinero. El procesos cocinero debe verificar que existe un alumno que espera por su comida. Ambos procesos deben utilizar semáforos para controlar el acceso a los recursos compartidos.
   Semáforos:

   - sillas: valor inicial S. También es correcto si en vez de un semáforo se considera un contador.
   - bandejas: si no hay bandejas disponibles el alumno se bloquea. Valor inicial: B.
   - cocinero: para controlar que sólo se sirva comida a un alumno por vez. Un alumno debe esperar si llega y el cocinero se encuentra ocupado.
   - alumno: par que el cocinero no sirva comida a un alumno que no lo pida.