

## Examen 25 de julio de 2016

Lea detenidamente las siguientes instrucciones. No cumplir los requerimientos puede implicar la pérdida del examen.

### Formato

- Indique su nombre completo y número de cédula en cada hoja (No se corregirán las hojas sin nombre, sin excepciones). Numere todas las hojas e indique la cantidad total de hojas que entrega en la primera.
- Escriba las hojas de un solo lado y empiece cada problema en una hoja nueva.
- Si se entregan varias versiones de un problema solo se corregirá el primero de ellos.

### Dudas

- Sólo se contestarán dudas de letra.
- No se aceptarán dudas en los últimos 30 minutos del examen.

### Material

- El examen es SIN material (no puede utilizarse ningún apunte, libro ni calculadora). Sólo puede tenerse las hojas del examen, lápiz, goma y lapicera en su banco. Todas sus demás pertenencias debe colocarlas en el piso debajo de su asiento.

### Aprobación

- Para aprobar el examen se debe tener un mínimo de 60 puntos.

### Finalización

- El examen dura 4 horas.
- Al momento de finalizar el examen no se podrá escribir absolutamente nada en las hojas, debiéndose parar e ir a la fila de entrega. Identificar cada una de las hojas con nombre, cédula y numeración forma parte de la duración del examen.

**Problema 1 ( 32 puntos)**

1. Describa y compare dos métodos para la asignación de datos en un sistema de archivos.
2.
  - a) Indique las ventajas del uso de DMA (*Direct Memory Access*).
  - b) Describa los pasos necesarios para realizar una transferencia mediante DMA.
3.
  - a) ¿Para que son útiles los algoritmos de reemplazo de marcos (*frames*)?
  - b) Describa el algoritmo de reemplazo de óptimo (*Optimal Page Replacement*) indicando beneficios y limitaciones.
4.
  - a) Describa brevemente la estructura de tablas de página jerárquica.
  - b) Describa el funcionamiento de la TLB (*Translation Lookaside Buffer*).
5. Describa el nivel 4 de RAID.
6.
  - a) Explique el concepto de emulación de plataforma e indique ventajas y desventajas.
  - b) Describa qué son las instrucciones sensibles y qué condición debe cumplir el hardware para soportar hipervisores de tipo 1.
7. Describa brevemente el proceso de invocar un llamado al sistema (*system call*) e indique tres formas posibles de pasaje de parámetros entre el proceso de usuario y el núcleo.
8. Describa 2 servicios que brinda el núcleo del sistema operativo para el manejo de Entrada/Salida

**Problema 2 (34 puntos)**

Se tiene un sistema con un único procesador con un planificador round robin con cuanto de 3 unidades de tiempo. Este sistema utiliza memoria virtual con direcciones virtuales de 12 bits. Usa un sistema de paginación para el cual dispone solamente de 8 marcos de memoria con un mecanismo de asignación global, cada uno de ellos con un tamaño de 512 bytes, y administra estos marcos por medio de una estructura de tabla de página simple. Además, utiliza un algoritmo de segunda chance para el reemplazo de páginas en memoria.

El sistema dispone de instrucciones de: escritura a memoria (E), lectura de memoria (L), operaciones que de entrada/salida (ES), y operaciones sobre registros (R) (p.ej. operaciones aritméticas). Cada operación E, L, y R ocupa una unidad de tiempo de ejecución. La operación ES requiere una unidad de tiempo de ejecución al comienzo, y luego una cantidad de unidades de tiempo variable que dependerá del tiempo que requiera el dispositivo para completar la solicitud. El tiempo requerido por el dispositivo está especificado, entre paréntesis, junto a cada operación de tipo ES (p.ej. ES(5)). Las operaciones de memoria van seguidas, entre signos de menor y mayor, de la página de memoria accedida (p.ej. L<1> y E<2>).

Se tienen tres procesos con ID 1, 2 y 3 con la siguiente secuencia de instrucciones:

<b>ID1</b>	L<1>	ES(3)	R	E<1>	E<2>	E<3>	-
<b>ID2</b>	R	E<1>	L<3>	R	ES(2)	R	E<1>
<b>ID3</b>	L<1>	L<2>	L<3>	R	E<1>	-	-

En estos tres procesos la secuencia de instrucciones ocupa la página 0 de memoria. Esta página puede no estar siempre en memoria.

Notas:

1. Los procesos 1, 2 y 3 comienzan su ejecución en el tiempo t=0, t=1 y t=2 respectivamente.
2. Al ingresar a la cola de listos, el sistema debe favorecer a los procesos que se encuentran esperando por una operación de entrada/salida.
3. Se utiliza un sistema de paginación bajo demanda puro en el que los procesos comienzan sin ninguna página cargada en memoria.
4. Los cambios de contexto y fallos de página no consumen tiempo de procesador.
5. Las páginas asociadas a las operaciones L y E dadas en la letra son válidas y forman parte del espacio de direccionamiento lógico de su respectivo proceso.

Se pide:

1. Realice un esquema que muestre el uso del procesador, la cola de procesos listos, y el estado de los procesos para cada unidad de tiempo.
2. Defina y calcule el tiempo de espera de ID 2 y el tiempo de retorno de ID 3.
3. Realice un esquema que muestre el estado de cada marco de memoria y la cola de segunda chance cada vez que ocurre un cambio de contexto y cambia el estado de la memoria.
4. Determine cuántos fallos de página se producen en total, y cuantas operaciones de escritura al espacio swap son necesarias.

**Problema 3 (34 puntos)**

En la casa de repuestos NOMACO los clientes llevan su pedido en una hoja y son atendidos por uno de los 20 vendedores los cuales arman el pedido y lo dejan en Expedición para que los clientes los retiren una vez le hayan pagado al vendedor.

Los clientes esperan en una fila por cada vendedor y deberán elegir la más corta después de haber escrito la hoja de pedido.

Se desea modelar este problema utilizando mailboxes las tareas Cliente, Vendedor y Expedición. No se permiten tareas auxiliares. Se debe especificar la semántica de los mailboxes utilizados.

Se dispone de las siguientes funciones auxiliares:

- escribir\_hoja(): pedido
  - Ejecutada por el cliente para escribir la hoja de pedido
- armar\_pedido(pedido): factura
  - Ejecutada por el vendedor para armar el pedido y dejarlo directamente en expedición
- pagar(factura): recibo
  - Ejecutada por el cliente para pagar la factura al vendedor
- armar\_paquete(recibo): paquete
  - Ejecutada por expedición para empaquetar el pedido correspondiente al recibo.
- irse(paquete)
  - Ejecutada por el cliente para irse del local con su paquete.

**Solución del problema 2**

1)

Tiempo	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
ID1	L<1>	ES(3)	B	B	B	L/1	L/1	L/1	R	E<1>	E<2>	L/2	L/2	L/1	L/1	E<3>	T	T
ID2		L/1	R	E<1>	L<3>	L/2	L/2	L/2	L/1	L/1	L/1	R	ES(2)	B	B	L/1	R	E<1>
ID3			L/1	L/1	L/1	L<1>	L<2>	L<3>	L/2	L/2	L/2	L/1	L/1	R	E<1>	T	T	T

B indica un proceso bloqueado

L/X indica un proceso esperando en la cola de listo en la posición X

T indica un proceso terminado

2)

El tiempo de espera de un proceso es el tiempo que el proceso pasa en la cola de listos.

Para el proceso ID 2 es en T=1, desde t=4 a t=9 y en t=14. Es decir  $1+9-4+1+1 = 8$  unidades de tiempo.

El tiempo de retorno de un proceso es el intervalo de tiempo desde que un proceso es cargado hasta que este finaliza su ejecución.

Para el proceso ID 3 es desde t=2 hasta t=14. Es decir  $14-2+1 = 13$  unidades de tiempo

3)

En total ocurren 8 cambios de contexto en los instantes de tiempo T=2, 5, 8, 11, 13, 15, 16 y 18.

Mostraremos el estado de la memoria en cada cambio de contexto.

- "f" indica el número de marco
- "id" el número de proceso
- "pag." el número de página
- "m" si la página fue modificada en memoria
- "sc" el bit de segunda chance
- "o" el orden en la cola de reemplazo.
- "\*" indica un fallo de página

T= 2

f	id	pag.	m	sc	o
0	1	0		1	1
1	1	1		1	2
2					
3					
4					
5					
6					
7					

T= 5

f	id	pag.	m	sc	o
0	1	0		1	1
1	1	1		1	2
2	2	0		1	3
3	2	1	1	1	4
4	2	3		1	5
5					
6					
7					

T= 8

f	id	pag.	m	sc	o
0	3	3		1	8
1	1	1			1
2	2	0			2
3	2	1	1		3
4	2	3			4
5	3	0			5
6	3	1			6
7	3	2			7

T= 11

f	id	pag.	m	sc	o
0	3	3		1	5
1	1	0		1	6
2	1	1	1	1	7
3	1	2	1	1	8
4	2	3			1
5	3	0			2
6	3	1			3
7	3	2			4

T= 13

f	id	pag.	m	sc	o
0	3	3		1	4
1	1	0		1	5
2	1	1	1	1	6
3	1	2	1	1	7
4	2	0		1	8
5	3	0			1
6	3	1			2
7	3	2			3

T= 15

f	id	pag.	m	sc	o
0	3	3		1	4
1	1	0		1	5
2	1	1	1	1	6
3	1	2	1	1	7
4	2	0		1	8
5	3	0		1	1
6	3	1	1	1	2
7	3	2			3

T= 16

f	id	pag.	m	sc	o
0					
1	1	0		1	1
2	1	1	1	1	2
3	1	2	1	1	3
4	2	0		1	4
5	1	3	1	1	5
6					
7					

T= 18

f	id	pag.	m	sc	o
0					
1					
2					
3					
4	2	0		1	1
5	2	1	1	1	2
6					
7					

4)

La cantidad de fallos de página son  $2+3+4+3+1+0+1+1=15$

Ocurrió solamente un acceso de escritura al espacio swap en T=10

**Solución del problema 3**

Se usan mailboxes infinitos con resolución FIFO, con receive bloqueante.

```
var
    mbx_colas: mailbox of array [1..20] of integer;
    mbx_espera: array [1..20] of mailbox of null;
    mbx_vendedor: array [1..20] of mailbox of pedido;
    mbx_cliente: array [1..20] of mailbox of factura;
    mbx_expedicion: mailbox of null;
    mbx_exp_recibo: mailbox of recibo;
    mbx_exp_factura: mailbox of factura;
```

```
procedure Cliente
```

```
var
    ped: pedido;
    fact: factura;
    rec: recibo;
    paq: paquete;
    colas: array[1..20] of integer;
    mas_corta, largo_min: integer;
```

```
begin
```

```
    ped := escribir_hoja();

    colas := mbx_colas.receive();
    mas_corta := 1;
    largo_min := colas[1];
    for i := 2 to 20 do
        if largo_min > colas[i] then
            largo_min = colas[i];
            mas_corta := i;
        end
    end
    colas[mas_corta]++;
    mbx_colas.send(colas);

    mbx_espera[mas_corta].receive();
    mbx_vendedor[mas_corta].send(ped);
    fact := mbx_cliente[mas_corta].receive();
    rec := pagar(fact);
    mbx_vendedor[mas_corta].send(null);

    colas := mbx_colas.receive();
    colas[mas_corta]--;
    mbx_colas.send(colas);

    mbx_expedicion.receive();
```

```
        mbx_exp_recibo.send(rec);
        paq := mbx_exp_paquete.receive();
        mbx_expedicion.send(null);
        irse(paq);
end

procedure Vendedor (id: integer)
var
    ped: pedido;
    fact: factura;

begin
    while true do
        mbx_espera[id].send(null);
        ped := mbx_vendedor[id].receive;
        fact := armar_pedido(ped);
        mbx_cliente[id].send(fact);
        mbx_vendedor[id].receive();
    end
end

procedure Expedicion
var
    rec: recibo;
    paq: paquete;

begin
    mbx_expedicion.send(null);
    while true do
        rec := mbx_exp_recibo.receive();
        paq := armar_paquete(recibo);
        mbx_exp_paquete.send(paq);
    end
end

// principal
begin
    var colas: array [1..20] of integer;
    for i := 1 to 20 do
        colas[i] := 0;
    end
    mbx_colas.send(colas);

    cobegin
        Vendedor(1);
        Vendedor(2);
```

```
Vendedor(3);  
Vendedor(4);  
Vendedor(5);  
Vendedor(6);  
Vendedor(7);  
Vendedor(8);  
Vendedor(9);  
Vendedor(10);  
Vendedor(11);  
Vendedor(12);  
Vendedor(13);  
Vendedor(14);  
Vendedor(15);  
Vendedor(16);  
Vendedor(17);  
Vendedor(18);  
Vendedor(19);  
Vendedor(20);  
Expedicion();  
    coend  
end
```