

Examen Julio de 2014

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 y cada parte del problema de teórico 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.

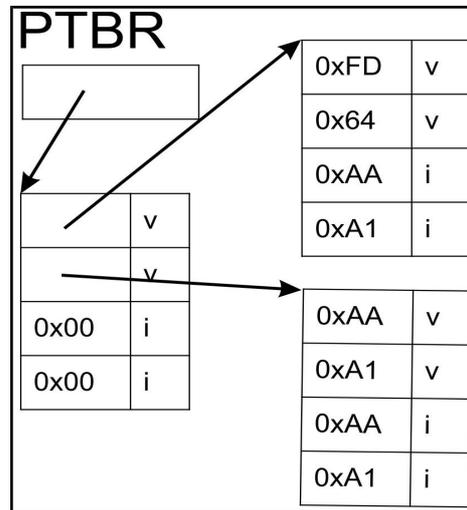
Problema 1 (32 puntos)

1. Describa dos mecanismos implementados a nivel de hardware para que proteger al sistema operativo.
2. a) ¿Qué es un buffer?
b) ¿Qué ventajas brinda el uso de la técnica de buffering a nivel del sistema operativo?
3. Describa el planificador (scheduler) de colas multinivel con retroalimentación (multi-level feed-back queue).
4. Defina las estrategias de reemplazo de marcos (frames) local y global.
5. Describa las condiciones necesarias que un sistema operativo debe garantizar para prevenirse de que ocurra deadlock.
6. Describa el método de asignación FAT (File Allocation Table) en un sistema de archivos.
7. Describa como funciona el método DMA (Direct Memory Access).
8. Describa los cuatro servicios básicos que debe brindar un sistema operativo

Problema 2 (35 puntos)

Sea un sistema operativo en un sistema computacional mono-procesador. El sistema utiliza memoria virtual con direcciones virtuales de 16 bits, paginación por demanda con asignación de marcos local de 4 marcos por proceso, algoritmo de reemplazo LRU (*Least Recently Used*) y un esquema de traducción de dos niveles.

Sean la siguientes tablas de páginas para algún proceso:



Notas:

- La cantidad de entradas en las tablas de página de primer y segundo nivel es de 4 y se numeran como 0, 1, 2 y 3.
- Las direcciones 0x00 representan memoria virtual no utilizada por el proceso.

Se pide:

1. Determine el tamaño en bytes de las páginas y marcos.
2. Determine el espacio virtual utilizado y la memoria residente en bytes para el proceso según las tablas de páginas presentadas en la figura.
3. Dada la siguiente secuencia de accesos, donde (X, Y, Z) representa: X referencia sobre la tabla de primer nivel, Y referencia sobre la tabla de segundo nivel, Z desplazamiento.

(1,1,FD)	(1,0,FD)	(0,0,0)	(0,1,10)	(0,0,01)	(1,3,11)	(1,1,F0)	(1,2,40)	(0,0,20)	(0,0,39)	(0,1,3)
----------	----------	---------	----------	----------	----------	----------	----------	----------	----------	---------

- a) Muestre un esquema que para cada instante de tiempo indique el estado de las tablas de página luego de la ejecución del acceso a memoria.
 - b) Suponga que se cuenta con un algoritmo de reemplazo óptimo. Muestre el mismo esquema que en la parte anterior pero utilizando este nuevo algoritmo de reemplazo.
 - c) Determine cuantos fallos de página se producen en la parte a) y cuantos en la parte b).
4. Indique que sucede si el proceso realiza la operación M(3, 3, 33).

5. Muestre el estado de la tabla de páginas luego de que el proceso solicite una nueva página e inmediatamente acceda a ella. Nota: la primer dirección de la nueva página corresponde a la siguiente dirección disponible del espacio de direccionamiento del proceso. Se asume el estado de la memoria virtual luego de la secuencia de accesos definidos en la parte 3 y utilizando el algoritmo de reemplazo LRU.

Solución:

1) Como las direcciones son de 16 bits, en total un proceso puede direccionar hasta 2^{16} bytes de memoria. Cómo se tiene una estructura de tabla de páginas de 2 niveles, con 4 entradas por nivel, en total un proceso puede referenciar hasta $4 \times 4 = 16 = 2^4$ marcos de memoria. Por lo tanto, para que cada proceso pueda referenciar toda la memoria disponible cada marco debería tener un tamaño de $2^{16}/2^4 = 2^{12} = 4 \text{ KB}$

Los marcos y las páginas tienen el mismo tamaño. Por lo tanto el tamaño de cada página es de 4KB.

2) El proceso tiene en total 8 páginas de memoria asignadas, por lo tanto ocupa $8 \times 4 \text{ KB} = 32 \text{ KB}$ de memoria virtual.

De esas 8 páginas solo 4 tienen marcos asignados, por lo tanto el proceso ocupa $4 \times 4 \text{ KB} = 16 \text{ KB}$ de memoria física.

3) a)

Se accede sin fallos de página a las páginas: (1,1), (1,0), (0,0), (0,1), (0,0)

Lista LRU: (1,1), (1,0), (0,1), (0,0)

Al acceder a (1,3) se produce un fallo de página. Se descarga la página (1,1) y se libera el marco 0xA1. El estado de las páginas queda como sigue:

Segundo nivel - Tabla 0		Segundo nivel - Tabla 1	
0xFD	v	0xAA	v
0x64	v	0xA1	i
0xAA	i	0xAA	i
0xA1	i	0xA1	v

Lista LRU: (1,0), (0,1), (0,0), (1,3)

Se produce un fallo de página accediendo a (1,1). Se descarga la página (1,0).

Segundo nivel - Tabla 0		Segundo nivel - Tabla 1	
0xFD	v	0xAA	i
0x64	v	0xAA	v
0xAA	i	0xAA	i
0xA1	i	0xA1	v

Lista LRU: (0,1), (0,0), (1,3), (1,1)

Se produce un fallo de página al acceder a (1,2). Se descarga la página (0,1).

Segundo nivel - Tabla 0		Segundo nivel - Tabla 1	
0xFD	v	0xAA	i
0x64	i	0xAA	v
0xAA	i	0x64	v
0xA1	i	0xA1	v

Lista LRU: (0,0), (1,3), (1,1), (1,2)

Se acceden sin fallos de página a las páginas (0,0), (0,0)

Lista LRU: (1,3), (1,1), (1,2), (0,0)

Finalmente se produce un fallo de página al acceder a (0,1). Se descarga la página (1,3).

Segundo nivel - Tabla 0		Segundo nivel - Tabla 1	
0xFD	v	0xAA	i
0xA1	v	0xAA	v
0xAA	i	0x64	v
0xA1	i	0xA1	i

Lista LRU: (1,1), (1,2), (0,0), (0,1)

b) Se accede sin fallos de página a las páginas: (1,1), (1,0), (0,0), (0,1), (0,0)

Al acceder a (1,3) se produce un fallo de página. Se descarga la página (1,0) y se libera el marco 0xA1. El estado de las páginas queda como sigue:

Segundo nivel - Tabla 0		Segundo nivel - Tabla 1	
0xFD	v	0xAA	i
0x64	v	0xA1	v
0xAA	i	0xAA	i
0xA1	i	0xAA	v

Se accede a la página (1,1).

Se produce un fallo de página al acceder a (1,2). Aquí se podría descargar la página (1,1) o (1,3). Se elige descargar la página (1,1).

Segundo nivel - Tabla 0		Segundo nivel - Tabla 1	
0xFD	v	0xAA	i
0x64	v	0xA1	i
0xAA	i	0xA1	v
0xA1	i	0xAA	v

Se acceden sin fallos de página a las páginas (0,0), (0,0), (0,1).

b) En total se producen 4 fallos de página en la parte a) y 2 fallos de página en la parte b).

4) El proceso intenta acceder a un espacio de memoria no asignado. El sistema operativo debe terminar el proceso con un error de acceso a memoria.

5)

Se inicializa una nueva tabla de 2do nivel, se produce un fallo de página al acceder a (2,0) y se invalida la página (1,1).

Segundo nivel - Tabla 0		Segundo nivel - Tabla 1		Segundo nivel - Tabla 2	
0xFD	v	0xAA	i	0xAA	v
0xA1	v	0xAA	i	0x00	i
0xAA	i	0x64	v	0x00	i
0xA1	i	0xA1	i	0x00	i

Problema 3 (33 puntos)

En la fábrica HighS3 trabajan 8 personas. En la misma hay una maquinaria especial que requiere de por lo menos 3 personas para su encendido por lo que si la maquinaria estaba apagada deberán esperar a que haya 3 personas para comenzar a trabajar.

Una vez que la maquinaria esta en funcionamiento pueden sumarse o restarse personas. Si en algún momento no hay personas en la maquinaria, la misma se apaga automáticamente por lo que volverá a necesitarse 3 personas para su encendido.

Dado que el trabajo en esta maquinaria genera alto stress, los trabajadores pasan buena parte de su tiempo descansando.

La fábrica cuenta con un único baño con dos gabinetes. Debido a las reglamentaciones de la fábrica no se permite que haya dos personas de distinto sexo dentro del mismo. Cuando hay al menos una persona en el baño de un sexo y llega otra persona del otro sexo ésta deberá esperar para entrar. Si hay una persona en el baño y llega otra persona del mismo sexo y hay un gabinete disponible podrá entrar al baño.

Se dispone de los siguientes procedimientos:

- `que_soy():{hombre, mujer}` que llamado por una persona indica su sexo.
- `que_hacer():{trabajar, descansar, baño}` que llamado por una persona indica que es lo próximo que va a hacer.
- `Trabajar()` que es llamado por una persona para trabajar.
- `descansar()` que es llamado por una persona para descansar
- `uso_gabinete()` que es llamado por una persona para utilizar un gabinete

Se deberá modelar utilizando **monitores** la realidad planteada.

Solución:

```

Procedure trabajador ()
begin
    while(true) do
        case que_hacer() of
            trabajar:
                Maquinaria.trabajar()
                Trabajar();
                Maquinaria.salir();
            descansar:
                descansar();
            baño:
                Baño.entrar(que_soy());
                usar_gabinete();
                Baño.salir();
        end switch
    end while
end

Monitor Maquinaria
var
    prendida: boolean;
    trabajando: integer;
    esperando: integer;
    espera: condition

Procedure trabajar()
begin
    if (not prendida and esperando < 2)) then

```

```
        esperando := esperando + 1;
        espera.wait;
        esperando := esperando - 1;
    end if
    trabajando := trabajando + 1;
    if (trabajando = 3) then
        prendida := true;
    end if
    espera.signal();
end
```

```
Procedire salir()
begin
    trabajando := trabajando - 1;
    if (trabajando = 0) then
        prendida := false;
    end if
end
```

```
begin
    trabajando := 0;
    prendida := false;
    esperando := 0;
end
end Maquinaria
```

```
Monitor Baño
var
    libres: integer;
    sexoEnUso: sexo;
    hombres, mujeres: condition;
```

```
Procedure entrar(s: sexo)
begin
    if(libres = 2)
        sexoEnUso = s;
    endif;
    if(s <> sexoEnUso or libres = 0)
        if(s = hombre)
            hombres.wait();
        else
            mujeres.wait();
        endif
        sexoEnUso = s;
    endif;
    libres := libres - 1;
    if(libres > 0)
        if(sexoEnUso = hombre)
            hombres.signal();
        else
            mujeres.signal();
        endif;
    endif;
end
```

```
Procedure salir(g: integer)
begin
    libres := libres + 1;
    if(libres = 1)
        if(sexoEnUso = hombre)
            hombres.signal();
        end if;
    end if;
end
```

```
        else
            mujeres.signal();
        endif
    else
        if(sexoEnUso = hombre)
            mujeres.signal();
        else
            hombres.signal();
        endif
    endif
end
end

begin
    libres := 2;
end
end Baño

begin
    cobegin
        Trabajador();
        ...
        Trabajador();
    coend
end
```