

Examen Diciembre de 2009

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. (No se corregirá la hoja que tenga el ejercicio compartido, sin excepciones).
- 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 ejercicio entero bien hecho y medio más.

Finalización

- El examen dura 4 horas.

Problema 1

Justifique brevemente cada respuesta:

1. ¿De que forma el sistema operativo se asegura que una aplicación no acceda directamente a un dispositivo?
2. Compare ventajas/desventajas entre un sistema operativo monolítico frente a uno en capas.
3. En un sistema que soporta hilos (*threads*) a nivel del núcleo, describa los pasos a seguir al realizar un cambio de contexto.
4. En un planificador Round-Robin, qué implicancias tiene agrandar o disminuir el tiempo de quantum.
5. ¿De qué se encarga el cargador (*loader*) del sistema operativo?
6. ¿Qué problemas presenta un sistema que no maneja memoria virtual?
7. Sea un sistema con paginación bajo demanda. Realice un diagrama de cómo se podría compartir una página entre dos procesos que residen dentro del mismo sistema operativo.
8. ¿Cuáles son los principales atributos que describen un archivo?
9. ¿Qué ventaja(s) propone implementar un sistema de archivos virtual en un sistema operativo?
10. Describa dos métodos para la asignación de los datos de un archivo en el disco.
11. En general los sistemas operativos brindan un servicio de *Buffering* para el subsistema de entrada/salida, ¿qué motiva brindar este servicio?

Problema 2

Sea un sistema de paginación en donde se utilizan direcciones virtuales de 48 bits, un sistema de direccionamiento multinivel (2 niveles), y las páginas son de 32 KB (kilobytes) de tamaño.

En este sistema las tablas de página de segundo nivel deben ocupar como máximo el tamaño de un solo marco (*frame*). A su vez, se desea que no contengan ninguna otra información como bits para permisos o de validación y, además, su utilización sea óptima.

Se pide:

- Realice un diagrama que muestre como mapear una dirección virtual a una dirección física. Especifique también que registro(s) son necesarios.
- ¿Qué tamaño tiene el espacio virtual?
- Dado que no se cuentan con bits extras (como el bit de validez) en las tablas de páginas de segundo nivel, muestre un mecanismo que garantice que el traductor pueda determinar si una página es válida o no para el proceso.
- ¿Qué tamaño tiene la tabla de páginas de primer nivel?

Nota: Se puede asumir que las direcciones físicas generadas son de 47 bits.

Solución:

- Dado que tenemos páginas de 32Kbytes = 2^{15} el desplazamiento (offset) necesita 15 bits de la dirección virtual. Lo que deja 33 bits para referenciar el marco físico. A su vez, como se pide que las tablas de segundo nivel utilicen solamente un marco (es igual al tamaño de la página 32Kbytes) y, dadas las condiciones de optimizar el espacio y se permite direcciones físicas de 47bits, utilizaremos entradas de largo de 32bits (desperdiciando un bit).

Por lo que tendremos 8192 entradas (32KBytes / 32bits) en las tablas de segundo nivel y no se desperdicia ningún bit en estas tablas. Para referenciar 8192 entradas necesitamos 13 bits.

La dirección física se compondrá entonces de 32 bits para referenciar el marco físico y 15 para el desplazamiento dentro del marco.

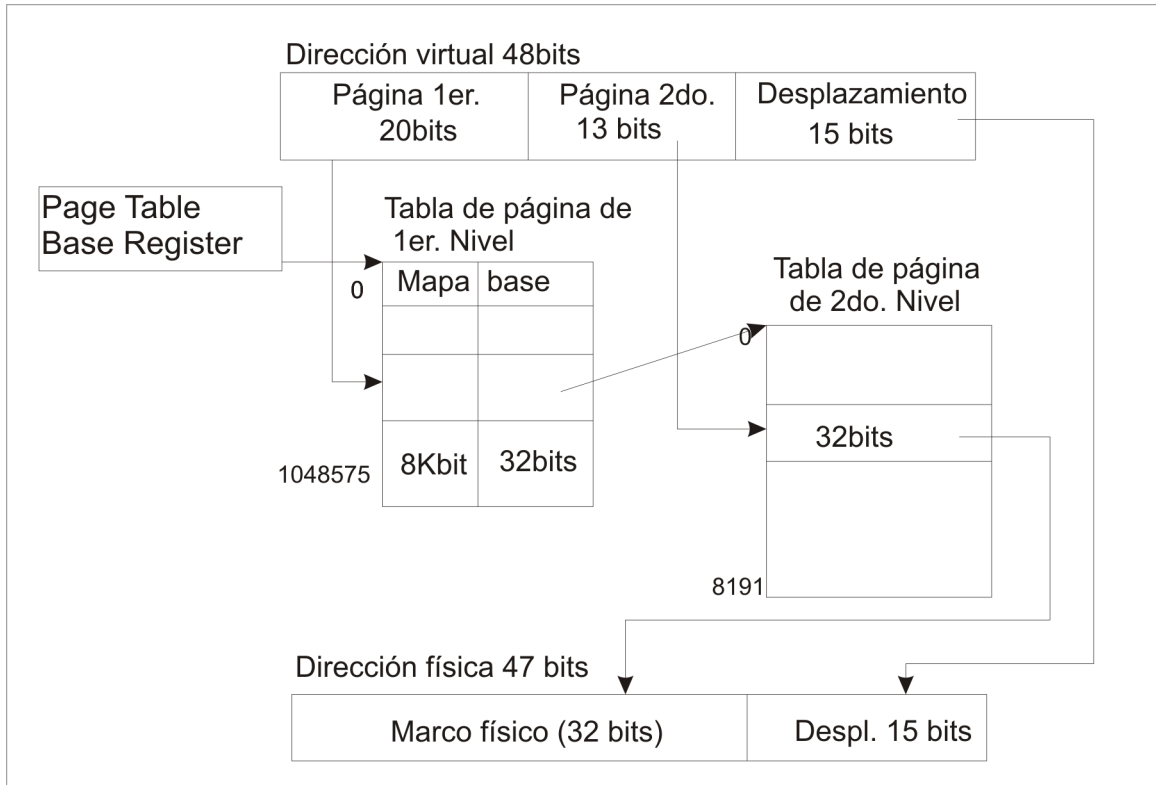
Finalmente, nos quedan los 20 bits restantes para indexar sobre la tabla de páginas de primer nivel. De esa forma, tenemos 1M de entradas posibles en la tabla de primer nivel.

El registro que nos permite indentificar donde comienza la tabla de primer nivel es el Page Table Base Register (PTBR).

- Dado que las direcciones virtuales son de 48 bits, el espacio virtual es de 2^{48} .
- Ya que no podemos poner información en las tablas de segundo nivel debemos agregar información en las entradas de la tabla de primer

nivel. Una solución posible es tener un mapa de bits en cada entrada. Como la tabla de segundo nivel tiene 8192, necesitamos 8192 bits (8Kbit = 1Kbytes).

- d) Dado que cada entrada de la tabla de páginas de primer nivel es la suma de 1Kbytes (mapa de bits) + 32bits (dirección base de la tabla de segundo nivel - 4 bytes) tenemos que el total ocupado por cada entrada es de 1028 bytes. Dado que tenemos 2^{20} entradas en la tabla de primer nivel, el tamaño será entonces: 1028 bytes * 2^{20} .



Problema 3

Se desea modelar usando ADA la atención de clientes en las cajas de un supermercado. El supermercado cuenta con 10 cajas, una cantidad ilimitada de clientes y un supervisor. Los clientes seleccionan una caja y esperan en la misma hasta que se atiendan todos los clientes que estaban en la fila antes que ellos. Los clientes son atendidos por orden de llegada con la excepción de las futuras mamás que tendrán prioridad sobre todos los demás clientes en cualquier caja que hayan elegido.

El cliente le pasará producto a producto de a uno a la caja para que esta los facture. Cuando no hay más productos el cliente enviará un producto FIN para señalar la finalización. Antes de pasar cada producto a la caja revisará si no está vencido. En caso de estar vencido el cliente llamará al supervisor para realizar una queja.

El supervisor recibirá los llamados de quejas y las registrará en una planilla. Cuando recibió 3 llamados de quejas o pasaron más de 5 minutos sin recibir ningún llamado de queja el supervisor atenderá las quejas pendientes en un orden de prioridad que el determinará. La atención del llamado consistirá en recibir el producto vencido y cambiarlo por otro.

El supervisor tiene prohibido comunicarse con las cajas, solo podrá hacerlo con los clientes.

Se dispone de las siguientes funciones:

- **soyFuturaMama(): boolean** Indica si el cliente es una futura mamá.
- **elegirCaja(): integer** Retorna la caja elegida por el usuario.
- **sacarProductoDeCarro(): producto** Retorna uno de los productos del carro del cliente o FIN si no hay más productos.
- **vencido(producto): boolean** Indica si el producto está vencido.
- **registrarQueja(integer)** Registra una queja de una caja que se pasa como parámetro.
- **prioridad(): integer** Retorna la próxima queja a atender.
- **resolverQueja(producto): producto** Recibe el producto vencido y retorna otro en buenas condiciones.

Solución:

```
Task type cliente;
```

```
Task body cliente is
```

```
Var
```

```
    caja: Integer;
```

```
    prod: producto;
```

```
    vencido: Boolean;
```

```
begin
```

```
    caja= elegirCaja();
```

```
    if soyFuturaMama()
```

```
        cajas[caja].ATENDER_MAMA;
```

```
    else
        cajas[caja].ATENDER;
    endif;
loop
    prod = sacarProductoDeCarro();
    if prod <> FIN and vencido(prod)
        supervisor.QUEJA(caja);
        supervisor.DAR_CAMBIO(caja) (prod);
    endif;
    cajas[caja].RECIBIR(prod);
until (prod == FIN)
End cliente

Task type caja is
    entry ATENDER;
    entry ATENDER_MAMA;
    entry RECIBIR(in paquete:producto);
End
Task body caja is
    Var prod: producto;
    var flag:Bool;
begin
    loop
        select
            accept ATENDER_MAMA;
            or when ATENDER_MAMA'Count = 0 =>
                accept ATENDER;
        end
    loop
        accept RECIBIR( prod);
    until (prod == FIN);
end
End caja
Task supervisor is
    entry QUEJA (in numcaja: Integer);
    entry DAR_CAMBIO(1..10) (inout cambio:producto)
End
```

Task body supervisor is

Var

```
    caja: Integer;  
    cant: Integer;  
    timeout: Boolean;
```

begin

```
    cant=0;
```

```
    loop
```

```
        timeout = false;
```

```
        -- Espero primer queja
```

```
        accept QUEJA(numcaja)
```

```
            caja = numcaja;
```

```
        End;
```

```
        registrarQueja(caja);
```

```
        cant = 1;
```

```
        -- Loop siguientes quejas o timeout por 5 minutos
```

```
        loop
```

```
            select
```

```
                accept QUEJA(numcaja)
```

```
                    caja = numcaja;
```

```
            End;
```

```
            registrarQueja(caja);
```

```
            cant = cant + 1;
```

```
            or
```

```
                delay 5*60;
```

```
                timeout = true;
```

```
            end;
```

```
        until (timeout = true or cant = 3);
```

```
        -- Proceso las quejas
```

```
        do
```

```
            accept DAR_CAMBIO(prioridad()) (cambio)
```

```
                cambio=resolverQueja(cambio);
```

```
            End
```

```
            cant=cant-1;
```

```
        until (cant = 0);
```

```
    end
```

```
End
```