

## Examen Marzo 2006

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

Para las siguientes partes conteste, justificando brevemente, cada una de las preguntas.

1. ¿Cuáles son los objetivos principales de un sistema operativo?
2. a) ¿Qué beneficios brinda el concepto de multiprogramación?  
b) ¿Qué beneficios brinda el concepto de memoria virtual?
3. ¿Cuáles son las principales ventajas de un sistema con micronúcleo?
4. En los sistemas multiprogramados y multiusuarios pueden existir problemas por procesos mal programados o mal intencionados.  
Esto puede llevar al sistema a un estado inconsistente.  
Enuncie y comente brevemente distintos mecanismos provistos por hardware que permiten protegerse de este tipo de programas.
5. ¿Qué son los denominados "llamados a sistema" (system calls)?
6. Describa brevemente el conjunto de tareas que implica realizar un cambio de contexto (context switch), mencionando la utilidad del bloque de control del proceso (PCB: Process Control Block).
7. a) ¿Qué soluciona el algoritmo de Dekker?  
b) Escriba el algoritmo de Dekker.
8. Defina los siguientes conceptos:
  - a) Utilización de procesador.
  - b) Tiempo de retorno.
  - c) Tiempo de espera.
9. Describa ventajas y desventajas de manejo de memoria a través de segmentación.
10. Describa dos métodos de acceso a un archivo.
11. a) ¿Qué es un buffer?  
b) ¿Qué ventajas brinda el uso de la técnica de buffering a nivel del sistema operativo?

=====

Solución:

1.

- Proveer un entorno y contexto apropiado para la ejecución de los programas de los usuarios.
- Proveer una interface con el usuario que sea expresiva e intuitiva.
- Eficiencia en la administración de los recursos.

2.

a. Incrementar el aprovechamiento del sistema, incluyendo los recursos de procesador, dispositivos de E/S, memoria , etc.

Varios programas son cargados en memoria y cuando alguno se bloquea, el sistema operativo le asigna la CPU a otro proceso.

b. Memoria Virtual produce los servicios básicos:

- Independiza la dirección de carga y ejecución de la dir. de memoria real.
- Que el tamaño de memoria que un proceso puede utilizar sea mayor a la memoria física del equipo.
- Abstrae el concepto de memoria a un largo y uniforme espacio independiente de la disponibilidad real.
- Separa la vista del usuario de la memoria virtual (o lógica) de la memoria física.

3.

- Aumenta la portabilidad y escalabilidad ya que encapsula las características físicas del sistema
- Para incorporar un nuevo servicio no es necesario modificar el núcleo.
- Es más seguro ya que los servicios corren en modo usuario.
- El diseño simple y funcional típicamente resulta en un sistema más confiable.

4.

- Modo doble de operación (Dual-Mode): El hardware trae por lo menos dos modos de operación: modo usuario y modo monitor.

En modo monitor están activas todo el conjunto de instrucciones que provee el hardware, mientras que en modo usuario solo unas pocas.

Los procesos de los usuarios ejecutan en modo usuario y el sistema operativo ejecuta en modo monitor.

- Protección de E/S: Todas las instrucciones de E/S a nivel de hardware son privilegiadas. O sea, solo pueden ejecutarse en modo monitor.
- Protección de memoria: Todas las referencias hechas a memoria principal son controladas por la MMU (dispositivo de hardware) que verifica el acceso correcto a memoria y su control es por instrucciones privilegiadas.
- Protección de CPU: Permitir que el procesador sea utilizado por todos en forma equitativa. El hardware provee de un timer que genera una interrupción. De esta forma, el sistema operativo puede expropiar el procesador a un proceso y asignar a otro el recurso procesador.

Aumento del nivel de privilegio de ejecución solo es posible mediante la entrega del flujo de ejecución a código protegido del Kernel

5. Son la interface provista por el núcleo del sistema para que los procesos de usuarios accedan a los diferentes servicios que brinda el sistema operativo.

6. Cuando un cambio de contexto ocurre, el núcleo del sistema debe guardar (salvar) el contexto del viejo proceso en el PCB y luego cargar el contexto del nuevo proceso (desde su PCB) en el procesador asignado.

7.

a. Soluciona el problema de la sección crítica para dos procesos.

b.

```
int turn;
bool flag[2];
```

<pre>p0() { while (true) {     flag[0] = true;     while (flag[1]) {         if (turn == 1) {             flag[0] = false;             while (turn == 1) ;             flag[0] = true;         }     }     SECCION CRITICA     turn = 1;     flag[0] = false;     ... }</pre>	<pre>p1() { while (true) {     flag[1] = true;     while (flag[0]) {         if (turn == 0) {             flag[1] = false;             while (turn == 0) ;             flag[1] = true;         }     }     SECCION CRITICA     turn = 0;     flag[1] = false;     ... }</pre>
---	---

8.

a. Es el porcentaje de uso del procesador en procesos de usuario.

b. Desde la perspectiva de un proceso, es el tiempo que transcurre desde que es cargado hasta que finaliza su ejecución.

c. Es la suma de los períodos que un proceso espera en la cola de procesos listos (ready queue).

9.

Ventajas:

- Simple compartir y proteger objetos, linking dinámico, reubicación etc.
- Podrían haber 2 diccionarios (código y datos), permitiendo por ej. reusar código automáticamente.

Desventajas:

- Complejas políticas de ubicación y reemplazo.
- Fragmentación externa.

10.

- Secuencial: se lee o escribe en el archivo en base a un puntero (file pointer).
- Directo: se lee o escribe en el archivo según una dirección que debe ser especificada cada vez.

11.

a. Es un área de memoria que sirve para guardar datos mientras son transferidos entre dispositivos o entre un dispositivo y un proceso.

b.

- Salvar la distancia de velocidades entre diferentes dispositivos.
- Adaptar la diferencia en el tamaño de la transmisión de datos de los dispositivos mediante técnicas de bloqueo.
- Adelantarse en la lectura obteniendo datos antes de que sean requeridos.

=====

### Problema 2

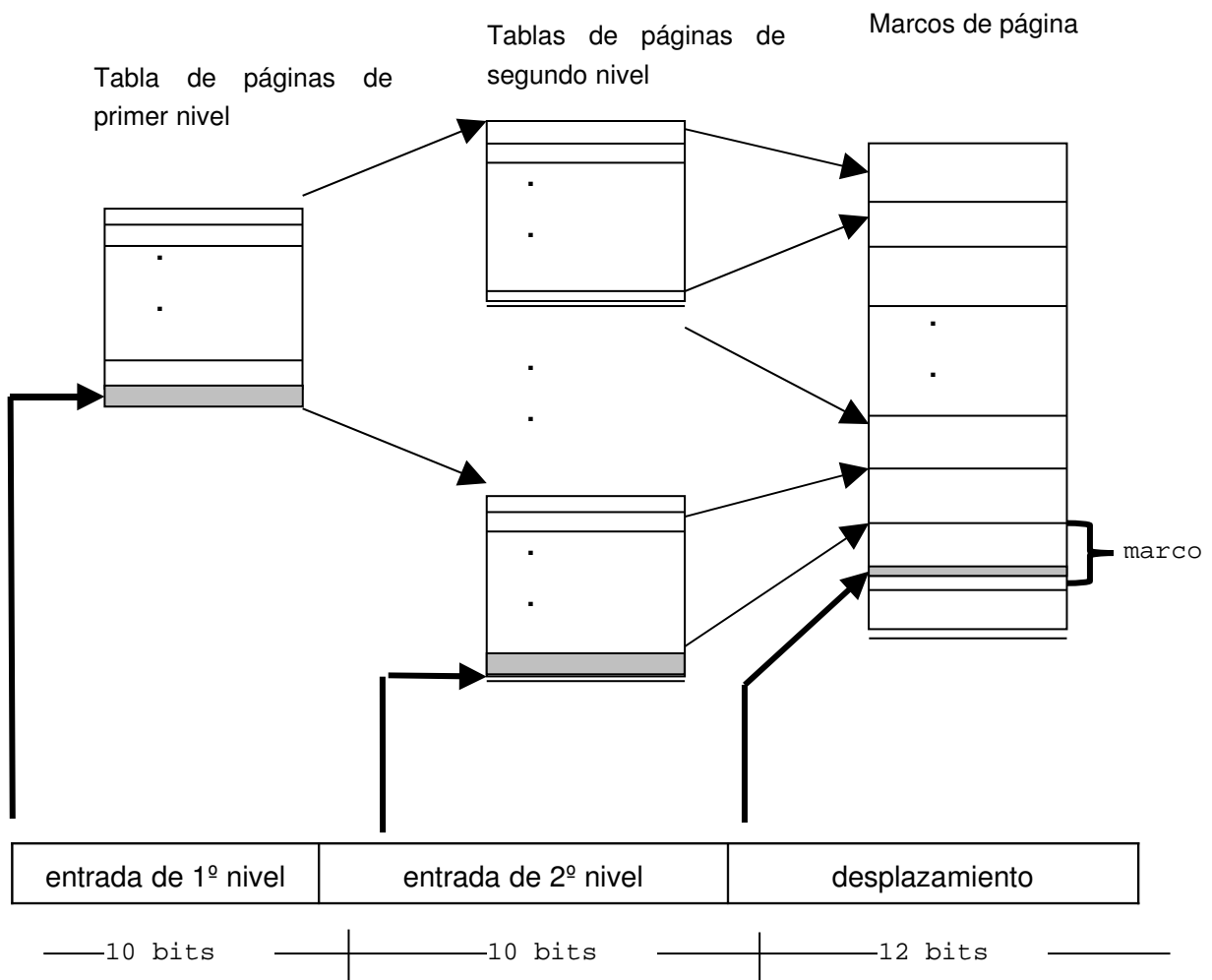
Un sistema operativo usa un sistema de memoria virtual paginada con direcciones virtuales de 32 bits y con páginas de 4KB. La tabla de páginas reside completamente en memoria principal, siendo el tamaño del descriptor de cada entrada de la tabla de páginas de 32 bits.

El espacio virtual es único por proceso, aunque ningún proceso llega a necesitar más de 16 MB. De este espacio, 12 MB se utilizan para almacenar el área de código más el área de datos a partir de la dirección virtual 0. Los 4 MB restantes se destinan al espacio de pila, que se almacena comenzando en la dirección virtual más alta y creciendo hacia direcciones bajas.

Responda justificando adecuadamente las siguientes preguntas:

- a) ¿Cuánto ocupa la tabla de páginas?
- b) ¿Cuál es la ocupación útil (entradas ocupadas) por proceso en % de la tabla de páginas?

Posteriormente, se decide usar un sistema de tablas de dos niveles, como se muestra a continuación.



Una dirección virtual se interpreta de la siguiente manera: Los 10 bits más significativos sirven para seleccionar la entrada de la tabla de primer nivel. Esta entrada contiene el número del marco donde reside la tabla de páginas de segundo nivel. Los 10 bits siguientes se usan para seleccionar una entrada de la tabla de segundo nivel. Esta entrada contiene el número del marco donde está cargada la página. Una vez localizado el marco, solo hace falta sumarle los 12 bits del desplazamiento de la dirección virtual.

Supuesto un proceso de este sistema, que ocupa como máximo 16 MB, y sabiendo que la tabla de primer nivel siempre está cargada en memoria durante la ejecución del proceso, mientras que las tablas de segundo nivel se cargan bajo demanda, responda justificando adecuadamente a las siguientes preguntas:

- c) ¿Qué tamaño tiene cada una de las tablas?
- d) ¿Cuántas tablas de segundo nivel utiliza el proceso? ¿Qué índices se ocupan de la tabla de primer nivel?
- e) ¿Cuánto espacio de memoria se necesita para almacenar simultáneamente todas las tablas necesarias?
- f) ¿Qué ventaja tiene este sistema con respecto al de una sola tabla de páginas?
- g) ¿Qué componente de hardware (en la MMU) puede acelerar los accesos a memoria?

Discuta cuántos accesos se deben realizar por cada acceso a memoria según se disponga o no de dicho componente.

=====

Solución:

- a) Con direcciones virtuales de 32 bits se puede direccionar hasta 4GB. Como las páginas son de 4KB, en 4GB existirán  $4GB/4KB = 1\text{ M}$  páginas. Como cada descriptor ocupa 4 Bytes, entonces la tabla ocupará  $1\text{M páginas} \times 4\text{B} = 4\text{MB}$ .
- b) Un proceso como máximo necesitará 16 MB. Como cada página soporta 4KB, entonces usará  $16\text{MB}/4\text{KB} = 4\text{K}$  páginas, por lo que usará 4K descriptores de la tabla. Si cada descriptor ocupa 4 bytes y la tabla ocupa 4MB, entonces, la ocupación útil de la tabla será:  $(4\text{K descriptores} \times 4\text{ bytes})/4\text{MB} = 2-8$   
0.39 %.
- c) Es decir, se está desperdiciando casi 4MB por proceso.
- d) Tanto la tabla principal como las secundarias tienen  $2^{10}$  (1024) entradas y ambas tienen descriptores de 4 bytes. Por tanto cualquier tabla ocupa  $4 \times 1024 = 4\text{ KB}$ .
- e) Necesita 4 tablas de segundo nivel. Puesto que el área de código y datos empieza en la dirección virtual 0 y ocupa 12 MB, en la tabla de primer nivel se ocuparán los descriptores 0, 1 y 2. El espacio de pila va desde la dirección más alta a direcciones bajas. Por tanto, en la tabla de primer nivel aparece ocupada la última entrada, la 1023.

- f) Se necesitan 5 tablas: la de primer nivel más cuatro secundarias. Cada tabla ocupa 4KB, por tanto en memoria principal se usarán 20 KB.
- g) El ahorro de espacio, puesto que se ha pasado de 4MB de una sola tabla a 20KB, con tablas multinivel.
- h) El componente en cuestión es una TLB (Translation Look-Aside Buffer)

Si no se dispone de una TLB es necesario hacer 3 accesos por cada acceso a memoria: uno a la tabla principal, otro a la tabla secundaria y por último el acceso al marco que contiene la información.

Si se dispone de una TLB (y la dirección en cuestión se encuentra presente en la TLB) ésta retorna el número de marco y solamente debemos accederlo.

=====



### Problema 3

Se considera un puente levadizo de además una única vía, con las características que se mencionan a continuación:

En cuanto al cruce por parte de vehículos:

- Encontrándose el puente vacío, se permitirá el ingreso desde cualquier dirección (sea Norte o Sur).
- Mientras existan vehículos cruzando, solo podrán ingresar aquellos que lleven la misma dirección que los que están cruzando.
- La interacción entre las tareas Autos y el Puente se supondrá de la siguiente forma:

puente.entrada;	Norte o Sur según corresponda.
delay(x);	Demora arbitraria en realizar el cruce.
puente.salida;	Fin de cruce del puente.

En cuanto al cruce por parte de barcos:

- Los barcos tendrán prioridad para el cruce, debiendo poder hacer el cruce tan pronto como se haya vaciado de vehículos y elevado el puente.
- Solamente se admitirá el cruce de hasta dos barcos a la vez.
- En caso de presentarse cola de barcos, se desea evitar que el puente se eleve y descienda para cada uno de ellos.
- Se dispondrá de una procedure `eleva_puente(si: in boolean)`; que elevara o bajara el puente finalizando su ejecución una vez que el puente llega a la posición deseada.
- La interacción entre las tareas Barcos y Puente se supondrá de la siguiente forma:

puente.llegada_barco	Aviso de que un barco desea pasar.
puente.entrada_barco	Comienza el cruce.
delay(y)	Demora arbitraria en realizar el cruce.
puente.salida	Fin de cruce de puente.

Se pide:

Implementar una tarea Ada tal que simule el puente con sus 5 puntos de entrada:

- entrada\_norte
- entrada\_sur
- llegada\_barco
- entrada\_barco
- salida

=====

Solución:

```
Task Puente is
```

```
    entry entrada_norte;  
    entry entrada_sur;  
    entry llegada_barco;  
    entry entrada_barco;  
    entry salida;
```

```
end Puente
```

```
Task Body Puente is
```

```
    autos, barcos, barcos_esperando: integer  
    direccion: {NORTE, SUR}  
    puente_levantado: boolean
```

```
begin
```

```
    autos := 0;  
    barcos := 0;  
    barcos_esperando := 0;  
    direccion := SUR; // da lo mismo  
    puente_levantado := false;  
    elevar_puente(false);  
  
    loop  
    begin  
        select  
            accept llegada_barco;  
                barcos_esperando := barcos_esperando + 1;  
        or
```

```
when autos = 0 and barcos < 2 =>
  accept entrada_barco do
    if not puente_levantado then
      elevar_puente(true);
    end
    puente_levantado := true;
    barcos := barcos + 1;
    barcos_esperando = barcos_esperando - 1;
  or
    when barcos = 0 and barcos_esperando = 0 and
      llegada_barco'count = 0 and (autos = 0 or
        direccion = SUR) =>
      accept entrada_sur do
        if puente_levantado then
          elevar_puente(false);
        end
        puente_levantado = false;
        direccion := SUR;
        autos := autos + 1;
      or
        when barcos = 0 and barcos_esperando = 0 and
          llegada_barcos'count = 0 and (autos = 0 or
            direccion = NORTE) =>
          accept entrada_norte do
            if puente_levantado then
              elevar_puente(false);
            end
            puente_levantado = false;
            direccion := NORTE;
            autos := autos + 1;
          or
            accept salida;
            if barcos > 0 then // sale un barco
              barcos := barcos - 1;
            else // sale un auto
              autos := autos - 1;
            end // select
          end // loop
        end Puente
      =====
```