

## Examen Febrero de 2003

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.
- Escriba el nombre en la hoja de letra, la cual deberá entregar conjuntamente con la solución.

### 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 (el 2 o el 3) bien hecho y medio más (de cualquiera de los 3).

### Finalización

- El examen dura 4 horas.

## Problema 1

Se debe responder a las cuestiones aquí planteadas de manera concisa (no más de 15 renglones por pregunta. En todos los casos debe justificar su respuesta.

### Pregunta 1

a) Si en un S.O. la implementación de los threads (hebras) se realizó a nivel de usuario, ¿qué sucede con las restantes hebras de un proceso cuando una se bloquea como consecuencia de la invocación de un servicio (system call) del S.O.?

=====  
Se bloquean también el resto de las hebras del mismo proceso.  
=====

b) Se analiza un sistema de paginación por demanda y se observa que, con cierta carga de trabajo, de la CPU se emplea un 15% y el disco de intercambio (swap) está ocupado un 92% del tiempo. ¿Qué acción tomaría para aumentar el rendimiento del sistema para esta misma carga de trabajo?

=====  
La situación descrita se ajusta a la de sobrecarga del sistema de memoria virtual o "Thrashing", resultado de que la suma de los "Working Set" de los procesos activos es sensiblemente mayor que la memoria real. La única acción posible sin variar la carga de trabajo es agregar más memoria real.  
=====

c) Describa brevemente en que consiste la técnica de swapping así como su oportunidad de uso.

=====  
Consiste en retirar a almacenamiento secundario casi la totalidad de las páginas de un proceso de forma de generar una gran disponibilidad de memoria real. Esto es necesario, en situaciones de 'Thrashing' o sobrecarga del sistema de memoria virtual.  
=====

### Pregunta 2

a) El semáforo es un buen mecanismo para obtener la exclusión mutua entre dos procesos. Explique brevemente porque si las operaciones P y V no se ejecutan en forma atómica puede no proporcionarse la exclusión mutua.

=====  
**wait(s)**=    si (s > 0) entonces  
              s := s - 1;  
              sino  
              detener\_tarea;  
              fin si;  
  
**signal(s)**=  si (hay\_tarea\_detenida) entonces  
              reanudar\_tarea;  
              sino  
              s := s + 1;  
              fin si;

#### Semáforos no binarios:

- 1) Tenemos una variable común **s**, en signal un incremento de **s** y en wait un decremento de **s**. Hay problemas cuando una tarea incrementa concurrentemente con otra tarea una variable común. Entonces si el incremento o decremento son divisibles hay problemas a que esa instrucción puede consistir de varias instrucciones de máquina.
- 2) Si suponemos que el incremento y el decremento de **s** son indivisibles, pero wait y signal divisibles también hay problemas pues puede darse a la vez la

pregunta del valor de **s** y el decremento/incremento de la variable. Entonces puede darle que **s** es mayor que cero y a la vez estarse decrementando y quedar menor que cero cuando va a ejecutar la opción del if.

Otra forma de verlo es haciendo el entrelazado de wait con wait, wait con signal o signal con signal.

b) Un proceso A recibe datos por la red y lo escribe en una zona de memoria. Un proceso B debe imprimir por pantalla cada uno de los datos recibidos por A. Explique si el siguiente código resuelve el problema.

```

Semaphore ready = 0;          While(TRUE) {
Semaphore go     = 0;          P(go);
While(TRUE) {                Copiar_datos();
  Leer_red();                 V(ready);
  v(go);                       Imprimir();
  poner_datos();              }
  P(ready); }

```

No lo resuelve

Podría ocurrir que Copiar\_datos se ejecutara a la vez e incluso antes que poner\_datos

Por ejemplo la primera vez podría darse esta secuencia de ejecución:

```

V(go)
      P(go)
      Copiar_datos()
poner_datos

```

Pregunta 3

a) En la gestión de memoria, ¿qué problemas plantea el algoritmo de búsqueda de bloques libres Best Fit?

Incrementa el consumo de CPU en ocasión de la búsqueda y tiende a generar bloques libres extremadamente pequeños y escasamente reusables.

b) Dado un sistema con paginación, donde la memoria tiene un ciclo de acceso de 200ns y hay una probabilidad del 50% de hallar en el cache del procesador la entrada correspondiente en la tabla de traducción de páginas. Indique la velocidad promedio de acceso a memoria.

La velocidad promedio de acceso a memoria será de 300 n/s.

Si hallamos el descriptor en el TLB => un acceso a memoria aplicable al 50% de los casos.

Si no hallamos el descriptor en el TLB => 2 accesos a memoria aplicable al 50% de los casos.

Como la probabilidad de hallar en el TLB el descriptor es del 50 %, entonces  $T = 0.5 * 200 + 0.5 * 400 = 300$

Pregunta 4

a) Demuestre la verdad o falsedad de cada uno de los siguientes enunciados:

1) las cuatro condiciones necesarias para el bloqueo mutuo son también suficientes si existe solo un recurso de cada uno de los tipos implicados en la espera circular.

=====

Verdadera

Dada la definición de grafo de asignación de recursos, puede mostrarse fácilmente que si el grafo contiene un ciclo y cada tipo de recursos tiene exactamente una instancia, entonces un ciclo implica que se ha producido un bloqueo mutuo.

En otras palabras, si todos tienen recursos asignados y están esperando por otros en una cola circular, (no pudiendo ni compartir los recursos ni quitársele la asignación a ningún proceso), entonces no van a poder liberar sus recursos hasta tanto no obtengan el recurso que esperan, que no será liberado hasta que alguien de la cadena desista o libere su recurso. Como esto no sucederá (porque no puede quitársele el recurso) entonces quedarán esperando eternamente.

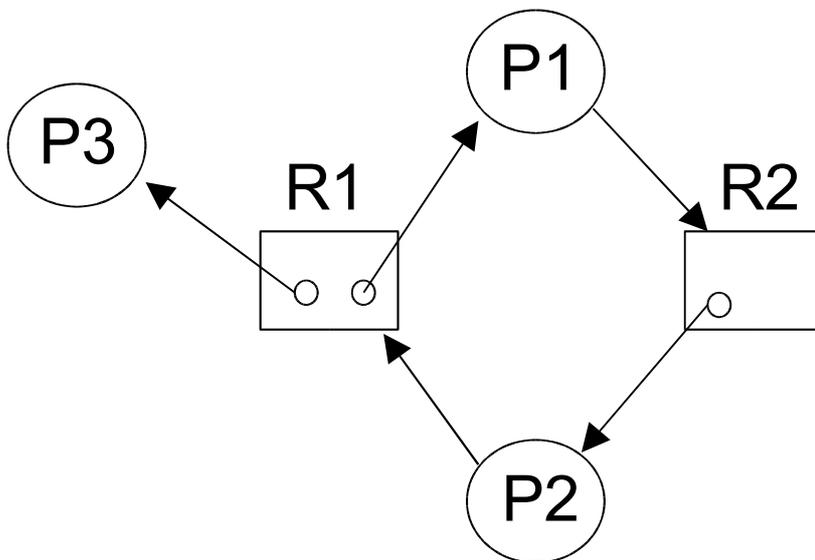
=====

2) las cuatro condiciones necesarias para un bloqueo mutuo son también suficientes si existen varios recursos de cada uno de los tipos implicados en la espera circular.

=====

Falsa

Esto puede verse fácilmente con el siguiente contraejemplo:



Aunque aquí hay un ciclo y se cumplen las 4 condiciones necesarias, cuando P3 libere una de las instancias de R1, dejará de existir el ciclo y allí se verá claramente que no hay bloqueo mutuo, pues ahora se le puede asignar a P2 una instancia de R1.

=====

b) Compare el uso directo de instrucciones tipo Test&Set con el uso de semáforos

**Test&Set**

Requiere sólo soporte de hardware

Se produce espera ocupada

Mecanismo de bajo nivel (instrucción de máquina)

No permite priorizar las esperas

Dependiente de la arquitectura del procesador

**Semáforo**

Requiere soporte del Sistema Operativo y Hardware

Procesos esperan en estado bloqueado

Mecanismo de mayor nivel (llamada al sistema)

Procesos bloqueados pueden priorizarse

Dependiente del sistema operativo (más portable)

Pregunta 5

Especifique los requerimientos de hardware para soportar eficientemente y en modo protegido un esquema de segmentación de memoria.

El procesador deberá contar con:

- diferentes niveles de ejecución de instrucciones según el esquema habitual.
- instrucciones privilegiadas y protegidas que entre otros cometidos permitan asegurar la integridad de los descriptores.
- El direccionamiento del espacio de memoria deberá ser a través del uso de descriptores protegidos. Los mismos deberán contar también con la indicación del largo del segmento, debiéndose producir una excepción en ocasión de indexar un descriptor con un valor superior a éste.

## Problema 2

En una empresa pública los clientes deben pedir hora de atención en N mostradores habilitados, presentando su Documento de identidad. Aquí se les dará una fecha y hora para atención, en función de la descripción de su caso y de su número de documento de identidad.

Los clientes deberán elegir el mostrador con menor cantidad de gente en la cola.

Se desea modelar este problema utilizando mailboxes, usando solo las primitivas send y receive.

Se contará con la función:

*Function fecha\_atención (documento\_identidad: in integer, Descripción\_caso: in string) : datetime*

Si se desea se podrá implementar únicamente una tarea auxiliar (Portero).

El programa principal deberá ejecutar el siguiente código:

Begin

Co-begin

Portero; /\* Opcional \*/

Cliente (Documento\_identidad1, "descripción caso 1");

Cliente (Documento\_identidad2, "descripción caso 2");

...

Cliente (Documento\_identidadM, "descripción caso M");

Mostrador (1);

Mostrador (2);

....

Mostrador (N);

Co-end;

End;

=====

**Utilizaremos mailboxes infinitos con resolución FIFO.**

```

type
  reg_atencion : record
    clase: enum {PEDIR, TERMINAR};
    indice: integer;
  end record;
  reg_cliente_razon : record
    Doc_Id: integer;
    descripcion: string;
  end record;
  esperando : array[1 .. n] of Integer;

var
  mutex_portero : mailbox(Void);
  mutex_mostrador : array[1 .. n] of mailbox(Void);

  portero_env : mailbox(reg_atencion);
  portero_rec : mailbox(Integer);

  mostrador_env : array[1 .. N] of mailbox(Integer);
  mostrador_rec : array[1 .. N] of mailbox(Integer);

  procedure Cliente(Doc_Id: in Integer, descripcion: in string);
var
  idx: integer;
  fechahora: datetime;
begin
  receive(mutex_portero, nil);

```

```

    {Mutuexcluye el pedido de nmero al portero }
    send(portero_env, {PEDIR, 0});
    { Le pide el nmero de mostrador al Portero }
    receive(portero_rec, idx);
    { Recibe el nmero de mostrador del Portero }
    send(mutex_portero, nil);

    receive(mutex_mostrador[idx], nil);
    send(mostrador_env[idx], {Doc_id, descripcion});
    { Le enva el doc de identidad y la descripcin al mostrador }
    receive(mostrador_rec[idx], fechahora);
    { Recibe la fechahora del mostrador }
    send(mutex_mostrador[idx], nil);

    ... PROCESAR TICKET ...

    send(portero_env, {TERMINAR, idx});
    { Le avisa a Portero que liber el mostrador}
end Cliente;

procedure Mostrador(nro_mostrador : in Integer);
var
    cli : cliente_razon;
    fechahora : datetime;
begin
    send (mostrador_env[nro_mostrador], nil)
loop
    receive (mostrador_env[nro_mostrador], cli);
    fechahora:=fecha_atencion(cli.Doc_Id, cli.descripcion);
    send (mostrador_rec[nro_mostrador], fechahora);
end loop
end Cliente;

procedure Portero(nro_tarjeta : in Integer);
var
    idx : integer;
    largo_colas : esperando;
    ra : reg_atencin;
begin
    for (idx = 1 .. n) do
        largo_colas[idx] := 0;
    end for;

    send(mutex_portero, nil);

loop
    receive(portero_env, ra);

    if (ra.clase = PEDIR) then
        idx := mas_corta(largo_colas);
        largo_colas[idx] := largo_colas[idx] + 1;
        send(portero_rec, idx);
    else { ra.clase = TERMINAR }
        largo_colas[ra.indice] := largo_colas[ra.indice] - 1;
    end if;
end loop;
end Cliente;

begin

```

```
co-begin
  Portero
  Cliente(Documento_identidad1,"descripción caso 1")
  Cliente(Documento_identidad2,"descripción caso 2")
  ...
  Cliente(Documento_identidadM,"descripción caso M")
  Mostrador(1)
  ...
  Mostrador(n)
co-end
end
```

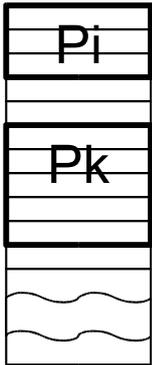
=====

### Problema 3

Se desea implementar un sistema operativo con multiprogramación y memoria virtual paginada sobre un equipo que tiene, entre otros, los componentes hardware siguientes:

- CPU con 32 bits para su conexión al bus de direcciones y un registro de reubicación (registro base).
- Bus de Datos de 32 bits.
- Un disco de 6GB y sectores de 512B, para utilizarlo en exclusiva como soporte de la memoria virtual.
- Una MMU con una caché interna (TLB) de 16 entradas.
- Una memoria principal de 64MB.

La memoria virtual, en un momento dado, tiene el aspecto de la figura siguiente:



Puede observarse que las tres primeras páginas están ocupadas por el proceso **Pi**, las dos siguientes están libres y las cinco siguientes están ocupadas por el proceso **Pk**.

Precisando más, nuestro modelo de memoria virtual debe cumplir lo siguiente:

- Las páginas serán de 4KB y los descriptores de página (entradas en la tabla de traducción de páginas) de 2 Bytes.
- Se dispondrá de un **único espacio de direcciones virtuales** para todos los procesos que se creen (tal y como se ve en la figura), o sea un espacio Virtual Común Compartido
- Las direcciones de los operandos del código objeto se encuentran expresadas relativas al comienzo de cada módulo.
- La Tabla de Páginas completa residirá en Memoria Principal.
- Debe poder precisarse si una página se encuentra en memoria principal o no así como si fue modificada desde que se cargó en memoria virtual
- Respetando las decisiones anteriores, el espacio de memoria virtual deberá ser lo más grande que se pueda

**Se pide:**

1. Indicar cuáles son los significativos de una dirección real para esta arquitectura concreta, el número de marcos, el tamaño de la tabla de páginas y un posible formato del descriptor de página. Justifique su respuesta.

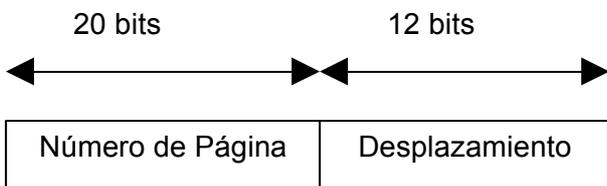
El tamaño de la memoria virtual es 4 GB debido a que la máxima capacidad de direccionamiento de la CPU es de  $2^{32}$  y que dicha cantidad cabe en el disco de 6 GB que disponemos

2. Supuesto que:

- La memoria virtual tiene el aspecto de la figura.
  - Utilizamos reubicación dinámica utilizando el registro base provisto por el hardware.
  - La primera página del proceso **Pk** está cargada en el marco 9 y el descriptor de dicha página se encuentra en la caché de la MMU.
  - La página cero también está cargada en memoria principal (en el marco 4) y su descriptor también se encuentra en la caché de la MMU.
  - Se intenta ejecutar la primera instrucción del proceso **Pk** que está justo al principio de su primera página y que resulta ser: **"move R0, 0x800"**, es decir, escribir el contenido del registro R0 en la dirección 0x800. Esta instrucción ocupa 4 octetos. (Nota: se entiende que la dirección es relativa a la dirección de carga del proceso)

Indicar a qué direcciones reales se accederá hasta que se complete la ejecución de la instrucción mencionada. Explicar los pasos que siguió el sistema (Hw/Sw) para calcular dichas direcciones reales.

Formato de dirección lógica (virtual)



Para expresar un desplazamiento en una página de 4KB se necesitan 12 bits. Como la arquitectura es de 32 bits, quedan 20 para dar el número de página.

- 3. a) Indicar cómo podría organizarse el disco para dar soporte a la memoria virtual (plantearse básicamente dónde se colocan las páginas). Señalar qué sectores ocuparía la página 0, la 1 y la última página de la memoria virtual.
- b) Considera Ud que el archivo de paginado puede ser un archivo convencional del tamaño necesario o debiera ser prealocado según algún procedimiento especial?. Justifique brevemente su respuesta.

Formato de descriptor de página



Donde los bits indican:

P: Presencia

M: Modificación

- 4. Se desea llevar la gestión de memoria virtual libre mediante un mapa de bits. Indicar detalladamente cuál es el tamaño del mapa de bits y dónde residiría.

El número de bits significativos de una dirección real es 26 ya  $2^{26} = 64$  MB

- 5. Indicar qué haría el Sistema Operativo cuando termina un proceso, para que desaparezca efectivamente de la memoria virtual. Explicitar las estructuras de datos que habría que modificar y en qué sentido.

Como quedan 14 bits para indicar el marco, la cantidad de marcos es  $2^{14} = 16384$

- 6. Escriba la cantidad máxima de memoria real (en bytes) que puede manejar este sistema. Justifique su respuesta.

La tabla de páginas tendrá  $2^{20}$  entradas y como cada entrada ocupa 2 bytes, el tamaño de la Tabla de Páginas será de 2 MB. ( $2 \text{ MB} = 2^1 * 2^{20}$  bytes)