

## Examen Diciembre de 2004

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. ¿Qué se logra en los sistemas multiprogramados? Justifique.
2. ¿Que tipo de procesamientos existen en un sistema multiprocesador? Describa brevemente cada uno de los que nombre.
3. ¿Qué permite el bit de modo provisto a nivel de hardware?
4. ¿Cómo se protege el excesivo uso de CPU por parte de un proceso?
5. ¿Describa 3 servicios que debe proveer el sistema operativo?
6. ¿Qué son los llamados al sistema (system calls)?
7. Describa las 4 (cuatro) condiciones necesarias que un sistema operativo debe garantizar para prevenirse de que ocurra deadlock.
8. En un sistema con manejo de memoria a través de segmentación, se tiene el siguiente acceso a memoria en el ambiente.  
La dirección esta formada por <numero de segmento>,<desplazamiento>

	+-----+		
		10	1324
	+-----+		
		...	
	+-----+		
10		1112	32122
	+-----+		
11		10121	32235
	+-----+		
		...	

**Acceso a memoria del proceso P.**

**Tabla de Segmento para el proceso P.**

¿El acceso a memoria es válido? Justifique.

9. Describa 3 (tres) métodos utilizados para tener cuenta que bloques de discos están libres.
10. ¿Que ventaja se tiene en un sistema que utiliza DMA (Direct Memory Access)? Explique los pasos que son llevados a cabo para la transferencia.
11. Defina brevemente que es "Mandatory Access Control" en oposición al concepto de "Discretionary Access Control"
12. Defina brevemente que es la ACL (Access Control List)

## Problema 2

Sea una máquina de hacer tortas, constituida por las siguientes unidades:

- Cinco expendedoras de ingredientes. Ofrecen ingredientes a la cocinera mediante el ofrecimiento del encuentro QUIERO\_INGREDIENTE cuando lo hay. La función HAY\_INGREDIENTE indicará la falta del mismo, en cuyo caso deberá comunicar al operador que lo necesita y no podrá continuar con sus tareas hasta que se le reponga el ingrediente.
- Un operador. El mismo, entre otras tareas, se encarga de reponer los ingredientes de las máquinas expendedoras. El operador invocará a la función OTRAS\_TAREAS y cuando finalice con ella deberá verificar si alguna de las máquinas necesita ingredientes. En caso afirmativo repondrá los ingredientes de las máquinas que los necesiten mediante el procedimiento RECARGAR\_INGREDIENTE y volverá a realizar sus otras tareas.
- Una Cocinera. Pide ingredientes a las expendedoras invocando a los encuentros QUIERO\_INGREDIENTE y cocina utilizándolos completamente invocando para ello a la rutina COCINAR. Considerar que la unidad que cocina comienza a cocinar una vez que tiene los cinco ingredientes necesarios.
- Una unidad de control. Es la encargada de verificar la producción debiendo invocar al procedimiento ALARMA si transcurre más de un minuto sin que la tarea cocinera este cocinando (Proponer la interfase con cocinera).

### Se pide:

Implementar utilizando ADA el sistema descrito de forma de que cada unidad sea una tarea.

### Notas:

- Se puede asumir que las rutinas HAY\_INGREDIENTE, OTRAS\_TAREAS, ALARMA y RECARGAR\_INGREDIENTE ya están implementadas.
- RECARGAR\_INGREDIENTE recibe como parámetro el número de la expendedora que necesita la recarga.
- Cada expendedora puede obtener su número invocando a la función OBTENER\_NÚMERO.

### Problema 3

Se tiene un sistema operativo monoprocesador multiprogramado con las siguientes características:

- Memoria virtual con paginación bajo demanda
- Planificación con dos niveles de prioridad dinámica utilizando round-robin (quanto de 100ms) dentro del mismo nivel de prioridad, recalculando la misma según la siguiente regla:
  - Si se consume el quanto, baja la prioridad
  - Si no se consume el quanto, sube la prioridad
  - Los procesos entran al sistema con la prioridad alta
- Tabla de paginas de dos niveles, con 1024 entradas tanto en la tabla de primer y segundo nivel
- Direcciones virtuales de 32 bits.

En este sistema, se tienen dos procesos que van a comenzar a ejecutar desde el instante  $t = 0$ . Estos procesos se comportan de la siguiente manera:

- Ejecutan un bucle durante 50ms, utilizando las paginas 2 y 4 de memoria virtual
- Se bloquean durante 100ms (Por ejemplo, con operaciones de E/S)
- Ejecutan un bucle durante 150ms, utilizando las paginas 2 y 3 de memoria virtual

Se dispone de 4 marcos de páginas libres (además de los empleados para almacenar el sistema operativo y las tablas de páginas de los dos procesos, que NUNCA se liberan)

Se pide:

- a) Dibujar un esquema o diagrama de planificación (tiempo versus procesos), en el que se indique el estado de cada proceso (listo/ejecutando/bloqueado) en cada intervalo de tiempo. Además se debe indicar en cada caso, el nivel de prioridad (alta/baja) en el que se encuentra el proceso.
- b) Indicar para cada intervalo de tiempo del diagrama anterior cual es el contenido de las tablas de paginas (numero de marco, bits de presente o ausente, Referenciado y Modificado) de cada proceso, y que paginas están en cada uno de los marcos. Se asume que el algoritmo de reemplazo es FIFO. Cuando se desconoce la información, marcarlo con el símbolo “?”

Se puede asumir que:

- El tiempo necesario para traer una página de disco, así como para llevarla de memoria a disco es 0.
- El tiempo necesario para realizar un cambio de contexto es 0.

=====

Soluciones

=====

Problema 1.

**1: ¿Qué se logra en los sistemas multiprogramados? Justifique.**

Se logra el incremento de utilización de CPU. Esto se logra al cargar varios procesos en memoria y alternar la ejecución cuando se producen requerimientos de E/S por ejemplo.

**2: ¿Que tipo de procesamientos existen en un sistema multiprocesador? Describa brevemente cada uno de los que nombre.**

Multiprocesamiento simétrico: el núcleo del sistema operativo puede correr en cualquier procesador.

Multiprocesamiento asimétrico: cada procesador tiene asignadas tareas específicas. Existe un procesador maestro donde corre el núcleo del sistema.

**3: ¿Qué permite el bit de modo provisto a nivel de hardware?**

El bit de modo permite tener dos modos de ejecución: modo monitor y modo de usuario.

El sistema operativo ejecuta en modo monitor y los procesos de usuario ejecutan en modo de usuario.

De esta forma, se permite proteger el sistema de programas que funcionan mal.

**4: ¿Cómo se protege el excesivo uso de CPU por parte de un proceso?**

A través de un timer. Cada cierta unidad de tiempo se genera una interrupción a nivel de hardware.

Esto permite al sistema expropiarle el recurso de CPU a un proceso y asignárselo a otro que esté listo para ejecutar.

**5: ¿Describa 3 servicios que debe proveer el sistema operativo?**

Ejecución de programas

Operaciones de E/S

Manejo del sistema de archivos

Comunicación

Detección de Errores

**6: ¿Qué son los llamados al sistema (system calls)?**

Los llamados al sistema proveen una interfase entre los procesos y el sistema operativo.

**7: Describa las 4 (cuatro) condiciones necesarias que un sistema operativo debe garantizar para prevenirse de que ocurra deadlock.**

Mutua exclusión: al menos un recurso debe adquirirse de modo que no pueda compartirse; es decir, sólo un proceso a la vez podrá usar ese recurso. Si otro proceso solicita ese recurso, el proceso solicitante deberá esperar a que se haya liberado el recurso.

Retener y esperar: debe existir un proceso que haya adquirido al menos un recurso y esté esperando para adquirir recursos adicionales que ya han sido asignados a otros procesos.

No expropiación: los recursos no se pueden arrebatarse; es decir, la liberación de un recurso siempre es voluntaria por parte del proceso que lo adquirió, una vez que ha terminado su tarea.

Espera circular: debe existir un conjunto  $\{P_0, P_1, \dots, P_n\}$  de procesos tal que  $P_0$  está esperando un recurso que fue adquirido por  $P_1$ ,  $P_1$  está esperando un recurso que fue adquirido por  $P_2$ , ...,  $P_{n-1}$  está esperando un recurso que fue adquirido por  $P_n$  y  $P_n$  está esperando un recurso que fue adquirido por  $P_0$ .

**8: En un sistema con manejo de memoria a través de segmentación, se tiene el siguiente acceso a memoria en el ambiente.**

```
+-----+
|  10  | 1324 |      Acceso a memoria del proceso P.
+-----+
```

...

+-----+		
10	1112	32122
+-----+		
<b>Tabla de Segmento para el proceso P.</b>		
11	10121	32235
+-----+		
...		

**¿El acceso a memoria es válido? Justifique.**

No es válido. Si bien se accede a un lugar de memoria del proceso, este está fuera del rango del segmento, por lo que ocurre un trap.

**9. Describa 3 (tres) métodos utilizados para tener cuenta que bloques de discos están libres.**

Bit Vector: Se mantiene un arreglo de bits que indican si el bloque está siendo usado o no.

Lista encadenada: Se mantiene una lista entre los bloques libres. Existe una referencia al primer bloque libre, el cual apunta al siguiente bloque libre, y así hasta el último que apunta a un bloque inválido.

Lista encadenada agrupada: Igual que la anterior, pero se mantiene un conjunto de bloques libres por bloque y el puntero al siguiente. También es necesario la referencia al primer bloque libre.

Conteo: Se mantiene una lista con referencias a bloques contiguos libres. Cada entrada de la lista apunta a un bloque libre y tiene un contador que indica cuantos bloques más libres contiguos existen.

**10. ¿Que ventaja se tiene en un sistema que utiliza DMA (Direct Memory Access)?**

**Explique los pasos que son llevados a cabo para la transferencia.**

El uso de DMA permite que la CPU pueda realizar otras tareas mientras se realiza una transferencia de E/S.

Para la transferencia se debe primero escribir un bloque de comando DMA en memoria.

El bloque contiene un puntero al origen de los datos, un puntero al destino y la cantidad de bytes a transferir.

La CPU escribe la dirección de este bloque en el dispositivo DMA y se libera para realizar otras tareas.

El controlador DMA se encarga, sin interrumpir al procesador, de la transferencia operando directamente sobre el bus.

**11: Defina brevemente que es "Mandatory Access Control" en oposición al concepto de "Discretionary Access Control".**

En el modelo de control de acceso discrecional (Discretionary Access Control, DAC) el usuario tiene un completo control sobre los objetos que le pertenecen y los programas que ejecuta. Así mismo, el programa ejecutado por un usuario tendrá (generalmente) los mismos permisos de ese usuario que lo está ejecutando .

Esto implica que la seguridad del sistema depende de las aplicaciones que se están ejecutando y, por tanto, cuando se produce una vulnerabilidad de seguridad en una aplicación, ésta afecta a todos los objetos a los que el usuario tiene acceso. Así, si la aplicación es ejecutada por root, el atacante puede obtener los máximos privilegios en la máquina, comprometiendo la seguridad global del sistema.

Otro modelo de control de acceso es el denominado control de acceso obligatorio (Mandatory Access Control, MAC), donde existe una política de seguridad definida por el administrador y que los usuarios no pueden modificar. Esta política va más allá de establecer propietarios de archivos sino que fija "contextos", en donde se indica cuando un objeto puede acceder a otro objeto.

Un ejemplo de esto es el modelo de Bell-Lapadula pensado para restringir permisos en un esquema military, donde se le asocia a los objetos y a los sujetos del sistema dos etiquetas (sensibilidad, categoría)

Donde sensibilidad es por ejemplo: no-clasificado, confidencial, secreto y ultra secreto

Y categoría es por ejemplo: otan, misiles, afganistan, etc.

Un sujeto para poder acceder a un objeto debe tener una sensibilidad igual o mayor con la que está clasificado el objeto y tener su misma categoría.

Por ejemplo si el sujeto tiene la clasificación (ultra secreto. Otan) no puede acceder a los objetos clasificados como (no-confidenciales, afganistan).

Un sujeto no puede cambiar la etiqueta de un objeto salvo elevar su sensibilidad.

**12: Defina brevemente que es la ACL (Access Control List).**

Una lista de control de acceso (ACL) es una lista de entradas asociadas a un archivo o carpeta que especifica qué usuarios y grupos tienen acceso a esa carpeta o archivo. Cada entrada de una ACL asigna a un usuario o grupo uno o varios tipos de niveles de acceso al objeto en cuestión. Estos pueden ser los básicos de lectura, escritura, ejecución, o también cambio de permisos, borrar y otros tipos de permisos que pueda implementar el sistema operativo

Mediante las ACLs es posible implementar un mecanismo de acceso discrecional que especifica permisos en una manera mas selectiva que lo que permiten, por ejemplo, los mecanismos tradicionales de Unix.

=====

Problema 2.

```

Task type expendedora is
  entry QUIERO_INGREDIENTE;
End

Task body expendedora is
  Var nro: Integer;

  nro = OBTENER_NUMERO();
  loop
    if (HAY_INGREDIENTE())
      accept QUIERO_INGREDIENTE
        dar_ingrediente();
      end
    else
      operador.RECARGAR(nro);
    end
  end
End

Task operador is
  entry RECARGAR (in nro_exp: Integer)
End

Task body operador is
  loop
    select
      accept RECARGAR(nro_exp: Integer) do
        RECARGAR_INGREDIENTE(nro_exp);
      end
    else
      OTRAS_TAREAS();
    end
  end
End

Task control is
  entry NO_COCINO;
  entry COCINO;
End

Task body control is
  loop
    accept NO_COCINO;
    select
      accept COCINO

```

```

    or
        delay 60
        ALARMA();
        accept COCINO;
        -- para no bloquear a la cocinera
        -- si termina
    end
end
End

Var exp array [1..5] of Task expendedora

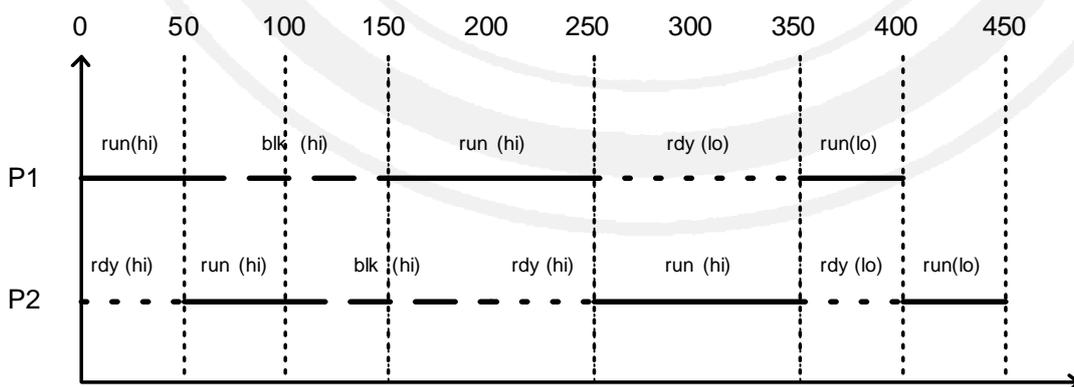
Task body cocinera is
loop
    control.NO_COCINO;
    for i:= 1 to 5 do
        exp[i].QUIERO_INGREDIENTE;
    end
    control.COCINO;
    COCINAR();
end
End

```

=====

### Problema 3.

La planificación puede verse en la siguiente figura. El único punto destacable es que existe un intervalo (entre 100 y 150) en que ninguno de los dos procesos está listo para ejecutar. Durante este intervalo se ejecutará la tarea "idle" (que se omite).



Los cambios de contexto están indicados con líneas verticales discontinuas. En la figura, **rdy** significa "listo para ejecutar", **run** indica "ejecutando", **blk** indica "bloqueado", **hi** indica prioridad alta, **lo** indica prioridad baja.

En cuanto a las tablas de páginas, dado que ninguno de los procesos accede a página con número mayor, sólo se usará la primera tabla de primer nivel en ambos procesos. Inicialmente todas las entradas de la tabla de primer nivel estarán inválidas hasta la primera referencia a memoria de los procesos considerados. Tras ésta, la primera entrada de la página de primer nivel en ambos procesos estará marcada como "presente", y contendrá el número de marco de la primera tabla de segundo nivel del proceso considerado.

Las tablas de segundo nivel de ambos procesos parten con todo inválido ("Ausente"). A continuación se indica como quedan estas en cada intervalo de planificación. En cada entrada indicamos el número de marco ("\*\*\*\*\*" si no lo sabemos o no interesa), el bit de presente/ausente ("P/A"), el bit R de referenciado ("?" cuando no lo sabemos) y el bit M de modificado ("?" cuando no lo sabemos)

T=0

Nada aún

T=50

Marcos

2 de P1	4 de P1	X	X
---------	---------	---	---

P1

P2

xxxxxxx P ??
... 1024 ...
xxxxxxx A ??

xxxxxxx A ??
xxxxxxx A ??
0000000 P R ?
xxxxxxx A ??
0000001 P R ?
... 1024 ...
xxxxxxx A ??

Ninguna
---------

T=100

Marcos

2 de P1	4 de P1	2 de P2	4 de P2
---------	---------	---------	---------

P1

xxxxxxxx P ??
... 1024...
xxxxxxxx A ??

xxxxxxxx A ??
xxxxxxxx A ??
00000000 P R ?
xxxxxxxx A ??
00000001 P R ?
... 1024...
xxxxxxxx A ??

P2

xxxxxxxx P ??
... 1024...
xxxxxxxx A ??

xxxxxxxx A ??
xxxxxxxx A ??
00000002 P R ?
xxxxxxxx A ??
00000003 P R ?
... 1024...
xxxxxxxx A ??

T=150

Siguen igual

T=250

Marcos

3 de P1	4 de P1	2 de P2	4 de P2
---------	---------	---------	---------

P1

xxxxxxxx P ??
... 1024...
xxxxxxxx A ??

xxxxxxxx A ??
xxxxxxxx A ??
00000000 A R ?
00000000 P R ?
00000001 P R ?
... 1024...
xxxxxxxx A ??

P2

xxxxxxxx P ??
... 1024...
xxxxxxxx A ??

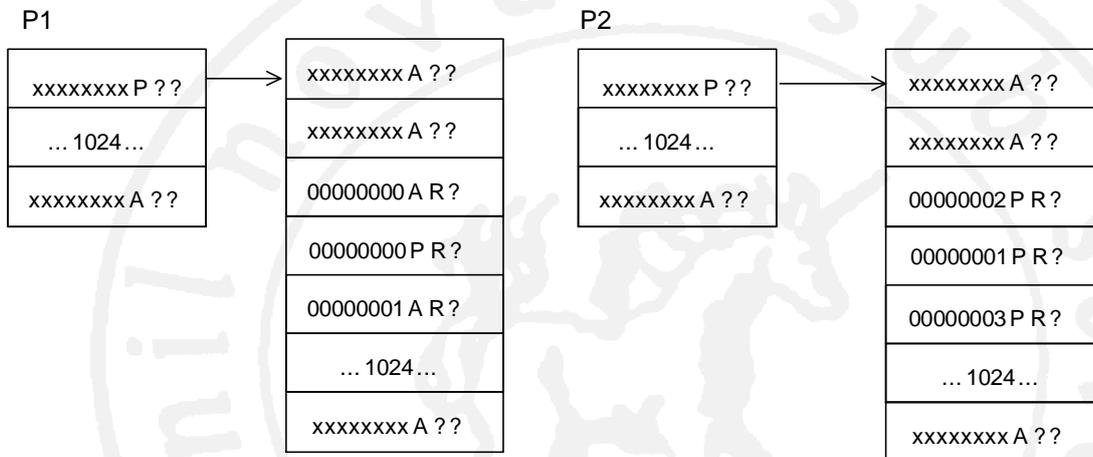
xxxxxxxx A ??
xxxxxxxx A ??
00000002 P R ?
xxxxxxxx A ??
00000003 P R ?
... 1024...
xxxxxxxx A ??

P1 referencia las páginas 2 y 3. Todo depende de cuantas referencias y en qué orden se hagan. Suponemos que tenemos una referencia a la 2 y luego otra a la 3. La 2 está, pero para incluir la 3 hay que expulsar una de las que ya tenemos, según la política FIFO. Luego la que expulsamos es la página 2 de P1.

T=350

Marcos

3 de P1	3 de P2	2 de P2	4 de P2
---------	---------	---------	---------



P2 referencia las páginas 2 y 3. Todo depende de cuantas referencias y en qué orden se hagan. Suponemos que tenemos una referencia a la 2 y luego otra a la 3. La 2 está, pero para incluir la 3 hay que expulsar una de las que ya tenemos, según la política FIFO. Luego la que expulsamos es la página 4 de P1.

T=400

Si suponemos que ahora nos mantenemos usando la página 3, no cambiaría la situación (en caso contrario tendríamos que volver a traer la 2, escogiendo la siguiente página que entró como víctima para la expulsión)

T=450

Suponiendo ahora que este proceso se mantiene usando la página 4, tampoco cambia nada (En caso contrario tendríamos que traer la página 3, expulsando alguna de las que tenemos ahora mismo)

Se debe aclarar que si los procesos hacen repetidas referencias a las páginas 3 y 4 en su última ráfaga, tendríamos que paginar bastante más, dado que con la política FIFO es muy posible que expulsemos una de las páginas que estamos necesitando (la 3 o la 4 de alguno de los dos procesos).

