

Examen Febrero de 2011

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 3 horas y 30 minutos.

Problema 1 (65 puntos)**Parte teórica (29 puntos, 5/5/9/5/5)**

1. ¿Qué son los llamados al sistema (*system call*)? Realice un dibujo/diagrama que muestre como son usados.
2. Describa las características principales de un sistema con micronúcleo (*microkernel*).
3. Sea un sistema operativo que implementa memoria virtual con paginación bajo demanda. direcciones virtuales de 32bits. La traducción se realiza a través de 3 niveles de tabla de página, donde las tablas de página de primer nivel tienen 512 entradas, las de segundo 512 entradas y las de tercer nivel 16 entradas.

- a. Realice un diagrama que muestre como se realiza la traducción la siguiente dirección virtual:

128	33	5	0
-----	----	---	---

- b. ¿Qué tamaño tienen las páginas y marcos?
 - c. En un sistema con soporte de hardware, ¿quién es el encargado de realizar esta traducción?
4. Describa en qué momento se puede realizar la asignación de direcciones (*address binding*).
 5. Mencione que métodos de asignación de bloques de datos tienen:
 - a. FAT (*File Allocation Table*).
 - b. Los inodos (*inode*).

Parte práctica (36 puntos, 9/9/9)

Se tiene un sistema operativo simétrico multiprogramado con un planificador con dos colas de prioridad (alta/baja) con retroalimentación (*multi-level feedback queue* con dos niveles de prioridad). En cada una de las dos colas se utiliza un algoritmo *round-robin* con quantum de 2 unidades de tiempo.

Los procesos al ser creados se les asignan la prioridad alta, la cual es modificada según los siguientes criterios:

- Un proceso sube a la cola de mayor prioridad si en las últimas 2 unidades de tiempo no ha usado el recurso procesador.
- Un proceso baja a la cola de menor prioridad si en las últimas 2 unidades de tiempo a utilizado completamente el recurso procesador (o sea, agotó su quantum ya que lo uso las 2 unidades).

A su vez, cuando un proceso pasa a un estado bloqueado su quantum es reiniciado.

Sean tres procesos que son creados en el instante $t=0$ y ejecutan lo siguiente:

1. Ejecutan un bucle durante 1 unidad de tiempo.
 2. Se bloquean durante 3 unidades de tiempo.
 3. Ejecutan un bucle durante 5 unidades de tiempo.
-
- a. Asumiendo un sistema monoprocesador, realice un esquema o diagrama de planificación (tiempo vs. procesos), en que se indique el estado de cada proceso (listo/ejecutando/bloqueado/terminado) 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. Asumiendo un sistema con dos procesadores, realice un esquema o diagrama de planificación (tiempo vs. procesos), en que se indique el estado de cada proceso (listo/ejecutando/bloqueado/terminado) 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.
 - c. Calcule las siguientes métricas para las partes a y b.
 - I. Utilización de CPU en los primeras 10 unidades de tiempo y en total (desde el $t=0$ hasta que el último proceso termina de ejecutar).
 - II. Tiempo de espera de cada proceso y promedio.
 - d. Sea un sistema con las siguientes características:
 - Tiene disponible cuatro procesadores.
 - El sistema operativo soporta solamente el modelo de hilos (*threads*) Mx1 (*Many-To-One*).
 - Se dispone de un proceso con dos hilos de ejecución y otro proceso con un solo hilo de ejecución
 - I. Realice un esquema o diagrama de planificación. Todos los hilos tienen la estructura de ejecución que fue presentada anteriormente (1bucle/3bloqueado/5bucle).
 - II. Calcule la utilización de CPU en el tiempo total.

Notas:

- Asuma que los tiempos utilizados por el sistema operativo (ej. ejecución del planificador) y los cambios de contexto son nulos.

Problema 2 (35 puntos)

Sea una fábrica de motores en la que trabajan seis trabajadores los cuales deben apretar tuercas. Para esto se dispone de cinco llaves de tuerca ubicadas en la mesa de herramientas. El trabajador primero limpia el motor con el que va a trabajar. Luego toma una llave de la mesa de herramientas y aprieta las tuercas del motor.

Luego de apretar la tuerca el trabajador debe dejar la llave que utilizó en su lugar y dirigirse al medidor para comprobar que las tuercas fueron ajustadas con la presión correcta. En caso que esté correcta deberá presionar un botón para se le asigne otro motor mientras que en otro caso deberá volver a ajustarla.

En la fábrica hay un supervisor de herramientas que solo podrá verificar las herramientas todas a la vez por lo que para que éste supervise las llaves deben estar libres.

Además existe un controlador de stock que controla lo producido y el material existente. Mientras que el controlador esta haciendo este trabajo las llaves no pueden estar asignadas a ningún trabajador ni al supervisor.

Los trabajadores tienen prioridad sobre las llaves respecto del supervisor y del controlador, salvo que cuando está el supervisor/controlador luego de que termina puede seguirlo el controlador/supervisor o algún trabajador sin respetar la prioridad.

Se pide:

- Modelar con semáforos las tareas Trabajador, Supervisor y Controlador.

Se dispone de los siguientes procedimientos:

- **presionarBoton()** ejecutada por los trabajadores para que se le asigne un nuevo motor.
- **limpiarMotor()** ejecutada por los trabajadores para limpiar un motor.
- **apretarTuercas(llave)** ejecutada por los trabajadores para apretar las tuercas con una llave .
- **verificarAjuste(): boolean** que devuelve falso en caso que la presión no sea la correcta.
- **verificarLlaves()** ejecutada por el supervisor para verificar todas las llaves.
- **otrasTareasSupervisor()** ejecutada por el supervisor cuando no quiere verificar llaves.
- **controlarStock()** ejecutada por el controlador de stock para controlar llaves y motores.
- **otrasTareasControlador()** ejecutada por el controlador cuando no quiere controlar.

Nota:

- No se podrán utilizar tareas auxiliares.

Solución:

Problema 1 - Parte Práctica

A)

Procesos 1	ejecutando alta	bloqueado alta	bloqueado alta	bloqueado alta	ejecutando alta	ejecutando alta	listo baja	listo baja	listo alta	listo alta
Procesos 2	listo alta	ejecutando alta	bloqueado alta	bloqueado alta	bloqueado alta	listo alta	ejecutando alta	ejecutando alta	listo baja	listo baja
Procesos 3	listo alta	listo alta	ejecutando alta	bloqueado alta	bloqueado alta	bloqueado alta	listo alta	listo alta	ejecutando alta	ejecutando alta
	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9

Procesos 1	ejecutando alta	ejecutando alta	listo baja	listo baja	listo alta	listo alta	ejecutando alta	terminado	terminado	terminado
Procesos 2	listo alta	listo alta	ejecutando alta	ejecutando alta	listo baja	listo baja	listo alta	ejecutando alta	terminado	terminado
Procesos 3	listo baja	listo baja	listo alta	listo alta	ejecutando alta	ejecutando alta	listo baja	listo baja	ejecutando alta	terminado
	t10	t11	t12	t13	t14	t15	t16	t17	t18	t19

B)

Procesos 1	ejecutando procesador 1 alta	bloqueado alta	bloqueado alta	bloqueado alta	ejecutando procesador 1 alta	ejecutando procesador 1 alta	ejecutando procesador 2 baja	ejecutando procesador 2 baja	ejecutando procesador 2 baja	terminado	terminado	terminado	terminado
Procesos 2	ejecutando procesador 2 alta	bloqueado alta	bloqueado alta	bloqueado alta	ejecutando procesador 2 alta	ejecutando procesador 2 alta	listo baja	listo baja	ejecutando procesador 1 alta	ejecutando procesador 1 alta	ejecutando procesador 1 baja	terminado	terminado
Procesos 3	listo alta	ejecutando procesador 1 alta	bloqueado alta	bloqueado alta	bloqueado alta	listo alta	ejecutando procesador 1 alta	ejecutando procesador 1 alta	listo baja	ejecutando procesador 2 baja	ejecutando procesador 2 baja	ejecutando Procesador 1 baja	terminado
	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12

C)

I)

Para la parte A

Para las primeras 10 unidades de tiempo

En 9 unidades de tiempo se está utilizando la CPU por alguna tarea
entonces $9 / 10 = 0.9 \rightarrow 90\%$

Para el total del tiempo

En 18 unidades de tiempo se está utilizando la CPU por alguna tarea
entonces $18 / 19 \sim 0,95 \rightarrow 95\%$

Para la parte B

Calculo por procesador individual

Para las primeras 10 unidades de tiempo el procesador 1 es utilizado en 8 unidades de tiempo entonces $8 / 10 = 0.8 \rightarrow 80\%$

Para las primeras 10 unidades de tiempo el procesador 2 es utilizado en 7 unidades de tiempo entonces $7 / 10 = 0.7 \rightarrow 70\%$

Para el total del tiempo el procesador 1 es utilizado en 10 unidades de tiempo entonces $10 / 12 \sim 0.83 \rightarrow 83\%$

Para el total del tiempo el procesador 2 es utilizado en 8 unidades de tiempo entonces $8 / 12 = 0.66 \rightarrow 66\%$

Calculo global (ambos procesadores)

Para las primeras 10 unidades de tiempo el procesador 1 es utilizado en 8 unidades de tiempo y el procesador 2 es utilizado en 7 unidades de tiempo entonces $(8+7) / (2 * 10) = 15/20 \rightarrow 75\%$

Para el total de tiempo el procesador 1 es utilizado en 10 unidades de tiempo y el procesador 2 es utilizado en 8 unidades de tiempo entonces $(10+8) / (2 * 12) = 18/24 = 3/4 \rightarrow 75\%$

II)

para la parte A

proceso 1 - tiempo de espera 8 unidades de tiempo

proceso 2 - tiempo de espera 9 unidades de tiempo

proceso 3 - tiempo de espera 10 unidades de tiempo

tiempo promedio de espera $(8+9+10)/3 = 27/3 = 9$

para la parte B

procesos1 - tiempo de espera 0 unidades de tiempo

procesos2 - tiempo de espera 2 unidades de tiempo

procesos3 - tiempo de espera 3 unidades de tiempo

tiempo promedio de espera $(0+2+3)/3 = 5/3$

D)

I)

Proceso 1	ejecutando procesador 1 alta	bloqueado alta	bloqueado alta	bloqueado alta	ejecutando procesador 1 alta	ejecutando procesador 1 alta	ejecutando procesador 1 baja	ejecutando procesador 1 baja	ejecutando procesador 1 baja	terminado
Proceso 2 (threads)	ejecutando procesador 2 alta	bloqueado alta	bloqueado alta	bloqueado alta	ejecutando procesador 2 alta	bloqueado alta	bloqueado alta	bloqueado alta	ejecutando procesador 2 alta	ejecutando procesador 2 alta
	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9

Proceso 1	terminado							
Proceso 2 (threads)	ejecutando procesador 1 baja							
	t10	t11	t12	t13	t14	t15	t16	t17

II)

Calculo por procesador individual

Para el total del tiempo el procesador 1 es utilizado en 14 unidades de tiempo entonces $14 / 18 = 7/9 \sim 0.78 \rightarrow 78\%$

Para el total del tiempo el procesador 2 es utilizado en 4 unidades de tiempo entonces $4 / 18 = 2 / 9 \sim 0.22 \rightarrow 22\%$

Para el total del tiempo el procesador 3 es utilizado en 0 unidades de tiempo entonces $0 / 18 = 0 \rightarrow 0\%$

Para el total del tiempo el procesador 4 es utilizado en 0 unidades de tiempo entonces $0 / 18 \sim 0 \rightarrow 0\%$

Calculo global (para todos los procesadores)

Para el total de tiempo el procesador 1 es utilizado en 14 unidades de tiempo, el procesador 2 es utilizado en 4 unidades de tiempo y los procesadores 3 y 4 son utilizados en 0 unidades de tiempo entonces $(14 + 4 + 0 + 0) / (4 * 18) = 18/72 = 1/4 \rightarrow 25\%$

Problema 2

```
Semaphore mutexAcceso;
```

```
Semaphore mutexSupCont;
```

```
Semaphore semLlaves;
```

```
Llave [] llaves{llave1, llave2, llave3, llave4, llave5};
```

```
int CantTrabajadores = 0;
```

```
void Trabajador()
{
    int id; Llave llave;
    presionarBoton();
    limpiarMotor();
    while(true)
    {
        P(semLlaves);
        P(mutexAcceso);
        if(CantTrabajadores == 0)
            P(mutexSupCont);
        CantTrabajadores++;
        for(id = 0; llaves[id]!=null; id++); // Busco llave
        llave = llaves[id];
        llaves[id]=null;
        V(mutexAcceso);
        apretarTuercas(llave);

        P(mutexAcceso);
        llaves[id]=llave; // Devuelvo llave
        CantTrabajadores--;
        if(CantTrabajadores==0)
            V(mutexSupCont);
        V(mutexAcceso);
        V(semLlaves);

        if(verificarAjuste())
        {
            presionarBoton();
            limpiarMotor();
        }
    }
}
```

```
void Supervisor()
{
    while(true) {
        otrasTareasSupervisor();
        P(mutexSupCont);
        verificarLlaves();
        V(mutexSupCont);
    }
}

void Controlador()
{
    while(true) {
        otrasTareasControlador();
        P(mutexSupCont);
        controlarStock();
        V(mutexSupCont);
    }
}

begin
    init(semLlaves, 5);
    init(mutexAcceso, 1);
    init(mutexSupCont, 1);

    cobegin
        Trabajador();
        Trabajador();
        Trabajador();
        Trabajador();
        Trabajador();
        Trabajador();
        Trabajador();
        Supervisor();
        Controlador();
    coend
end
```