

|                                                            |                                                                                                                                    |
|------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| Material permitido: <b>Solo calculadora no programable</b> | <b>Aviso 1:</b> Todas las respuestas deben estar debidamente razonadas.                                                            |
| Tiempo: <b>2 horas</b>                                     | <b>Aviso 2:</b> Escriba con buena letra y evite los tachones.                                                                      |
| <b>N1</b>                                                  | <b>Aviso 3:</b> Solución del examen y fecha de revisión en <a href="http://www.uned.es/71902048/">http://www.uned.es/71902048/</a> |

1. Conteste **razonadamente** a los siguientes apartados:

- (1 p) Explicar la diferencia entre un enlace duro y un enlace simbólico.
- (1 p) Describir el algoritmo de planificación basada en múltiples colas de prioridad y realimentación.

2. (2 p) Explicar **razonadamente** qué es un cambio de contexto o proceso y cuáles son las principales causas que lo motivan.

3. (2 p) Enumerar y describir **brevemente** las capas de software de E/S del núcleo de un sistema operativo.

4. (2 p) Una persona tiene en su casa una jaula llena de canarios en la que hay un plato de alpiste y un columpio. Todos los canarios quieren primero comer del plato y luego columpiarse, sin embargo sólo tres de ellos pueden comer del plato al mismo tiempo y solo uno de ellos puede columpiarse. Escribir el pseudocódigo basado en C de un programa que usando *semáforos binarios* coordine la actividad de los canarios. Dicho programa debe tener tres partes: declaración de variables y semáforos, código del proceso `canario`, y código de la función principal para inicializar los semáforos y lanzar la ejecución concurrente de los procesos.

**Nota:** Antes de escribir el pseudocódigo se debe explicar adecuadamente el significado de cada uno de los semáforos binarios y variables que se van a utilizar en el mismo.

5. (2 p) El sistema operativo en colaboración con el hardware gestiona la memoria principal mediante paginación por demanda. El tiempo medio de acceso a memoria es de 115 ns. La traducción de direcciones se realiza usando una MMU con banco de registros. El tiempo medio de acceso al banco de registros es despreciable. La atención de un fallo de página emplea en promedio 10 ms si existe disponible un marco vacío o si la página reemplazada no se modifica y 35 ms si la página reemplazada se modifica. La página que se va a reemplazar se modifica el 30 % de las veces. ¿Cuál es la tasa máxima aceptable de fallos de página para obtener un tiempo medio de despacho de una referencia a memoria menor de 230 ns? Despreciar la existencia de memoria caché.

## SISTEMAS OPERATIVOS

### Solución examen febrero 2012 (1ª semana)

#### Solución Ejercicio 1

- a) Un *enlace* permite conectar un directorio con un archivo o subdirectorio ya existente en otro directorio. Un archivo o subdirectorio referenciado por un enlace se dice que está *compartido*.

Para simplificar la explicación se utilizará el término *directorio origen* para denominar al directorio en el que reside un enlace y el término *elemento destino* para denominar al archivo o subdirectorio compartido. El directorio que contiene el elemento destino se denominará como *directorio destino*.

Una posible forma de implementar un enlace es crear en el directorio origen una nueva entrada que sea una copia parcial o completa de la entrada del directorio destino asociada al archivo o subdirectorio compartido. La nueva entrada creada en el directorio origen es marcada con algún identificador especial que permita reconocer al sistema operativo que se trata de una entrada asociada a un enlace. Los enlaces con esta implementación son denominados en algunos sistemas operativos como *enlaces duros*.

Otra forma de implementar un enlace es como un tipo especial de archivo que contiene el nombre de ruta absoluta o relativa del archivo o subdirectorio compartido. A los enlaces implementados de esta forma se les denomina *enlaces simbólicos*. La entrada asociada a un enlace simbólico en el directorio origen es completamente distinta a la entrada del directorio destino asociada al archivo o subdirectorio compartido. En este caso no hace falta marcar la entrada con algún identificador especial ya que el tipo del archivo lo identifica como enlace simbólico.

- b) En el *algoritmo de planificación basada en múltiples colas de prioridad* un proceso solo puede pertenecer a una determinada cola de preparados o cola de prioridad durante su tiempo de vida. En consecuencia, posee una prioridad estática. Con este algoritmo siempre se ejecuta un proceso perteneciente a la cola de preparados con mayor prioridad. Sólo cuando dicha cola está vacía se puede planificar un proceso perteneciente a una cola de preparados de menor prioridad. Un proceso a lo largo de su vida solo puede pertenecer a una determinada cola de preparados. Además para planificar cada cola se puede utilizar un algoritmo distinto: FCFS, SJF, turno rotatorio, etc.

Si se permite que un proceso pueda ir cambiando de cola de prioridad durante su existencia, es decir, que su prioridad pueda ser modificada dinámicamente, se dice que el algoritmo posee *realimentación*.

La implementación de la realimentación requiere definir un mecanismo que regule cómo se modifica dinámicamente la prioridad de los procesos durante su tiempo de existencia. O visto de otra forma, cuándo se produce la transición de un proceso de una cola de prioridad a otra.

Un posible mecanismo de regulación de la prioridad es el siguiente: cuando un proceso llega al sistema se coloca en la cola de mayor prioridad. Cuando es planificado puede usar el procesador durante un cuanto. Al finalizar el cuanto es colocado en la siguiente cola de menor prioridad. Cuando un proceso llega a la cola de menor prioridad, cada vez que consuma un cuanto volverá a ella hasta que consiga completarse. En consecuencia, el algoritmo de planificación de la cola de menor prioridad es de tipo turno rotatorio. El resto de colas se pueden planificar internamente con un algoritmo FCFS.

#### Solución Ejercicio 2

Se denomina *cambio de proceso* o *cambio de contexto* a la operación que realiza el sistema operativo consistente en interrumpir la ejecución de un proceso A para iniciar o continuar la ejecución de otro

proceso B. Entre las principales causas que motivan un cambio de contexto se encuentran las siguientes:

- *El proceso en ejecución pasa al estado bloqueado.* Un proceso A pasa al estado bloqueado cuando debe esperar por la aparición de algún evento. Mientras llega a producirse dicho evento se puede ejecutar otro proceso B.
- *La terminación (voluntaria o forzada) del proceso en ejecución.* Obviamente si un proceso termina se puede pasar a ejecutar otro.
- *El sistema operativo termina de atender una interrupción y existe un proceso B en estado preparado de mayor prioridad que el proceso actual A.* En dicho caso dependiendo de la política de planificación que siga el sistema operativo, se puede ejecutar dicho proceso B más prioritario o continuar con el proceso A que se estaba ejecutando.
- *El proceso A en ejecución ha excedido el tiempo máximo de ejecución ininterrumpida.* En los sistemas de tiempo compartido, a cada proceso se le asigna un tiempo máximo de ejecución ininterrumpida con objeto de poder atender las peticiones de todos los usuarios conectados al sistema. Superado dicho tiempo se produce un cambio de proceso, el proceso A en ejecución pasa al estado preparado y un proceso B en dicho estado pasa al estado ejecutándose.

### **Solución Ejercicio 3**

De forma general, el software del núcleo de un sistema operativo necesario para la gestión de las operaciones de E/S se puede organizar en tres capas:

- *Subsistema de E/S.* Es el componente del sistema operativo que se encarga de efectuar todas aquellas tareas necesarias para la realización de las operaciones de E/S que son comunes a todos los dispositivos e independientes de los mismos. Es decir, el subsistema de E/S gestiona la parte independiente del dispositivo de todas las operaciones de E/S. Entre las tareas que se encarga de realizar el subsistema de E/S se encuentran las siguientes: asignación y liberación de dispositivos dedicados, bloqueo (si procede) de procesos que solicitan una operación de E/S, planificación de la E/S, buffering, invocación del driver del dispositivo adecuado, y gestión de los errores producidos en las operaciones de E/S.
- *Drivers de dispositivos.* Un driver de dispositivo contiene el código que permite a un sistema operativo controlar un determinado tipo de dispositivo de E/S. Un driver de dispositivo interactúa con el subsistema de E/S y con el controlador de E/S que controla el dispositivo. Un driver suministra al subsistema de E/S el conjunto de funciones que se pueden realizar sobre el dispositivo, tales como lectura o escritura. Además un driver puede invocar a ciertas rutinas o procedimientos del núcleo. Los procedimientos a los que tiene acceso un driver quedan definidos por la interfaz de drivers del subsistema de E/S. El driver de un dispositivo interactúa con el controlador de E/S, cargando en sus registros diferentes órdenes para que las efectúe sobre el dispositivo, comprobando su estado e inicializándolo si es necesario.
- *Manejadores de las interrupciones.* El manejador de una interrupción es una función del núcleo encargada de atender una determinada interrupción. Con objeto de que el rendimiento del computador sea óptimo, los manipuladores de interrupciones tienen una alta prioridad de ejecución. Además su código suele ser pequeño y rápido de ejecutar. Las acciones específicas que realiza un manejador de interrupciones dependen de cada tipo de interrupción. Si el driver del dispositivo se bloqueó en espera de que el controlador de E/S estuviera preparado para procesar otra petición de E/S, entonces una acción que debe realizar un manejador de interrupción es el desbloqueo del driver del dispositivo mediante el uso del mismo mecanismo de sincronización (semáforo, mensajes,...) que utilizó el driver para bloquearse.

## Solución Ejercicio 4

La solución que se propone en la Figura 1 para este problema utiliza las siguientes variables globales y semáforos binarios:

- CC. Variable global de tipo entero para llevar la cuenta de canarios que están comiendo del plato de alpiste.
- sem1. Semáforo binario que se utiliza para garantizar la exclusión mutua en el uso de la variable global CC.
- sem2. Semáforo binario que se utiliza para sincronizar el acceso de los canarios al plato de alpiste.
- sem3. Semáforo binario que se utiliza para garantizar la exclusión mutua en el uso del columpio

```
/* Definición variables y semáforos */
int CC=0;
semáforo_binario sem1, sem2, sem3;

void canario() /* Proceso canario */
{
    wait_sem(sem1);
    CC=CC + 1;

    if(CC > 3)
    {
        signal_sem(sem1);
        wait_sem(sem2); /* Esperar a que haya un puesto libre en el plato */
    }
    else signal_sem(sem1);

    comer();

    wait_sem(sem1);
    if(CC > 3) signal_sem(sem2); /* Avisa de que hay un puesto libre en el plato */
    CC=CC-1;
    signal_sem(sem1);

    wait_sem(sem3); /* Comprueba si puede usar el columpio, en caso negativo espera */
    columpiarse();
    signal_sem(sem3); /* Avisa de que ha terminado de columpiarse */
}

void main() /*Inicialización de semáforos y ejecución concurrente*/
{
    init_sem(sem1,1);
    init_sem(sem2,0);
    init_sem(sem3,1);
    ejecución_concurrente(canario, canario,...);
}
```

**Figura 1** – Solución Ejercicio 2

## Solución Ejercicio 5

El tiempo medio de gestión de un fallo de página  $t_{gf}$  depende de si la página ha sido modificada, lo cual ocurre el 30 % de las veces. Si la página se modifica el tiempo medio de gestión del fallo es de 35 ms, en caso contrario emplea 10 ms. Luego  $t_{gf}$  se calcula de la siguiente forma:

$$t_{gf} = 0,7 \cdot 10 \cdot 10^{-3} + 0,3 \cdot 35 \cdot 10^{-3} = 0,0175 \text{ s}$$

El tiempo medio de despacho de una referencia a memoria es:

$$t_R = (1 - p) t_{am} + p t_{gf}$$

Se desea que  $t_R$  sea menor de 230 ns, es decir:

$$(1 - p) t_{am} + p t_{gf} < 230 \cdot 10^{-9}$$

Despejando  $p$  se obtiene:

$$p < \frac{230 \cdot 10^{-9} - t_{am}}{t_{gf} - t_{am}}$$

Sustituyendo valores y operando se obtiene finalmente que:

$$p < \frac{230 \cdot 10^{-9} - 115 \cdot 10^{-9}}{0,0175 - 115 \cdot 10^{-9}} \rightarrow p < 6,5715 \cdot 10^{-6}$$