

# Sistemas Operativos. Septiembre 2005. Ingeniería Informática

*Duración: 2 horas*

## 1. Procesos-hilos (2 puntos).

- a) Qué es un proceso en estado zombie y qué recursos mínimos mantiene que no pueden ser liberados por el sistema operativo.
- b) JAVA es un lenguaje que permite hilos (thread) como objeto de programación. ¿Es cierto o falso que la ejecución de un programa JAVA con hilos requiere que el Sistema Operativo del sistema donde se ejecuta admita asimismo la gestión de hilos? ¿Por qué?

## 2. Memoria (2 puntos).

- a) Enumerar los tipos de regiones de memoria que puede tener un proceso en UNIX
- b) Considerar un sistema con paginación por demanda en el que se obtienen los siguientes índices de utilización: uso de CPU 20%, uso de disco de swap 97%, uso de otros dispositivos 5%. ¿Cuáles de las siguientes acciones mejorarán probablemente el uso de CPU? (a) Instalar una CPU más rápida. (b) Instalar un disco de swap mayor. (c) Incrementar el grado de multiprogramación. (d) Disminuir el grado de multiprogramación. (e) Instalar más memoria principal. (f) Instalar un disco más rápido o múltiples controladores con múltiples discos duros. (g) Añadir prepaginación a los algoritmos de fetch de páginas. (h) Aumentar el tamaño de página.

## 3. Sistemas de Ficheros (2 puntos).

- a) Lista los componentes de la estructura de un sistema de ficheros clásico tipo UNIX
- b) ¿Qué relación guardan entre sí “sistema de ficheros”, “disco” y “partición”?

## 4. Planificación (2 puntos).

- a) ¿Qué se entiende por “época” en las política de planificación aplicada en LINUX
- b) Indicar cuáles de las siguientes transiciones entre los estados de un proceso no se pueden producir en un sistema con un algoritmo de planificación SJF (“Primero el trabajo más corto”):
  - 1) Bloqueado a listo.
  - 2) Ejecutando a listo.
  - 3) Ejecutando a bloqueado.
  - 4) Listo a bloqueado.
  - 5) Listo a ejecutando.

## 5. Concurrency (2 puntos).

- a) ¿En qué circunstancias puede aparecer la situación conocida como “inversión de prioridad”?
- b) Empareja los distintos patrones de modelos de concurrencia con los problemas que se resolverían más adecuadamente con ellos:

- Exclusión mutua	Acceso y modificación del fichero de contraseñas
- Señalización	Comunicación, por pipe (cauce), entre dos procesos de UNIX
- Productor/Consumidor	Operaciones de backup a CDROM de varios usuarios
- Lector/Escritor	Tomas periódicas de medidas y presentación de resultados

# Sistemas Operativos. Septiembre 2005. Ingeniería Informática

*Duración: 2 horas*

## 1. (Concurrencia)

Considérese la siguiente implementación del problema clásico productores-consumidores utilizando semáforos:

### PRODUCTORES:

```
while (TRUE) {
    v = producir();
    wait(Hay_Hueco);
    lock(mutex);
    acceso_buffer_in;
    unlock(mutex);
    signal(Hay_Dato);
};
```

### CONSUMIDORES:

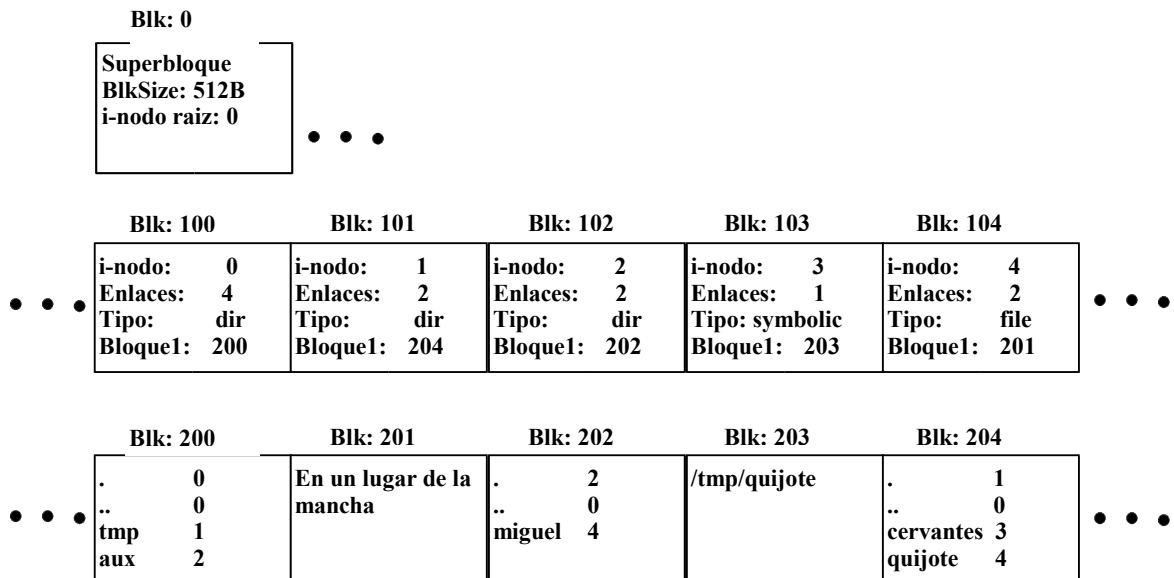
```
while (TRUE) {
    wait(Hay_Dato);
    lock(mutex);
    acceso_buffer_out;
    unlock(mutex);
    signal(HayHueco);
    consumir(w);
};
```

Modifique la implementación para que se dé preferencia a los consumidores sobre los productores: si hay productores y consumidores esperando para acceder al buffer compartido, se elegirá un consumidor para que realice su operación cuando éste quede libre.

**Sugerencia:** Inspirarse en el problema de lectores escritores

## 2. (Sistema de Ficheros)

Dado el sistema de ficheros de la figura, cuyo tamaño de bloque es de 512 bytes.



Se pide ejecutar las siguientes instrucciones en el orden indicado indicando

- 3) ¿Qué información del sistema de ficheros es utilizada en cada momento y para qué?
- 4) ¿Cuáles son las modificaciones introducidas en el sistema de ficheros?

6. cat /aux/miguel
7. rm /tmp/quijote
8. rm /tmp/cervantes

NOTA: Comando *cat*: muestra un fichero de texto por pantalla.

Comando *rm*: borra un fichero.