

Nombre:

DNI:

Segundo control de teoría

Justifica todas las respuestas brevemente. **Una respuesta sin justificar se puntuará como no contestada.**

1. (3.5 puntos) Preguntas cortas

- a) Explica cómo se implementa la compartición de memoria entre 2 threads.

- b) ¿Qué es un dispositivo virtual?

- c) ¿Qué utilidad tiene replicar información del disco en memoria?

- d) ¿Qué efecto tiene sobre un proceso su bloqueo?

- e) ¿Qué hace el sistema operativo cuando recibe una interrupción de la tarjeta de red?

- f) ¿Qué comando Linux nos permite crear una pipe con nombre? (Indica la sintaxis completa):

Nombre:

DNI:

2. (2 puntos) Sistema de ficheros

La compañía Shamsung nos ha donado un disco de Nueva Generación¹, con tecnología no volátil de 8Mbytes (8.388.608 bytes) de capacidad, cuyos sectores ocupan 4096 bytes. Además nos ha facilitado las siguientes rutinas **síncronas** para acceder al dispositivo:

```
int shame_open(); // Para inicializar el disco
int shame_read(int n, char *buff);
    // Para leer el contenido del sector 'n' del disco en el buffer 'buff'
int shame_write(int n, char *buff);
    // Para escribir el contenido de 'buff' en el sector 'n' del disco
int shame_close(); // Para dejar de usar el disco
```

Sobre este disco queremos montar un sistema de ficheros similar a FAT (encadenado en tabla) con las siguientes características: puede contener ficheros y directorios; los nombres de los ficheros pueden tener una longitud máxima de 255 caracteres; un bloque de este sistema de archivos se corresponde con un sector del disco; y los bloques se identifican con un número de 32 bits. Responde a las siguientes preguntas:

- a) Sabiendo que el primer bloque está reservado para el superbloque ¿Cuántos bloques ocupa la FAT?

- b) ¿Cómo se representa el espacio libre del disco y dónde lo guardas?

- c) ¿Qué debe contener una entrada del directorio y cuánto ocupa? (Si su tamaño es variable indica mínimo y máximo)

¹ Concretamente del rodaje de la conocida serie de ciencia ficción *ST: La Nueva Generación* (1987).

SO2

Nombre:

DNI:

- d) Suponiendo que el contenido del directorio raíz (/) se encuentra en el primer bloque de datos disponible del SF ¿en qué sector del disco encontramos este directorio?

- e) Suponiendo que el directorio raíz contiene un único fichero ¿cuál es el tamaño máximo de este fichero en este disco?

- f) Si limitamos el tamaño de los directorios para que ocupen un bloque como máximo ¿Cuál es el número máximo y mínimo de ficheros que puede contener?

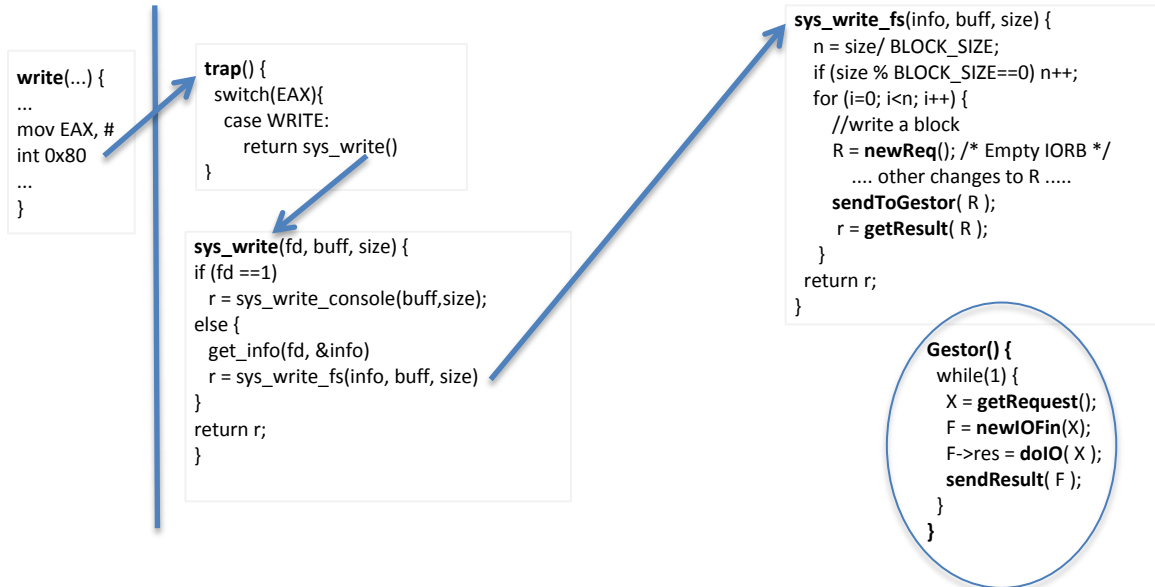
- g) ¿Este sistema de ficheros sufre el problema de la fragmentación externa? ¿Y de fragmentación interna?

Nombre:

DNI:

3. (2.5 puntos) Gestores

Con el disco anterior en el sistema, el señor BakaBaka quiere implementar un sistema de ficheros para facilitar su uso. Para ello crea un nuevo proceso de sistema que ejecuta la rutina *Gestor*; añade 2 nuevas llamadas a sistema (*open* y *close*) para acceder a los ficheros; añade la función *get_info* que devuelve una estructura con la información dinámica y el bloque inicial de un fichero; y actualiza las llamadas a sistema (*read* y *write*) de ZeOS de forma que pueda acceder a estos ficheros. Por ejemplo, la llamada *write* queda así:



Responde a las siguientes preguntas justificadamente:

- a) La rutina *write* de la figura ¿implementa una llamada síncrona o asíncrona?

- b) ¿Cuál/cuáles de las funciones destacadas de la figura NO son bloqueantes?

SO2

Nombre:

DNI:

- c) ¿Qué campos son necesarios en el IORB para enviar una petición al gestor?

- d) ¿Cómo intercambiará información la llamada *sys_write_fs* con este nuevo proceso gestor?

- e) Indica qué debe hacer la rutina *getResult*.

- f) Indica el pseudocódigo de la rutina *doIO* del gestor.

```
int doIO(struct IORB * iorb) {
```

- g) ¿Sería correcto mover la instrucción “*r = getResult(R);*” de la rutina *sys_write_fs* fuera del bucle? ¿Por qué?

SO2

Nombre:

DNI:

- h) Suponiendo que las llamadas a sistema read/write definidas fueran síncronas ¿sería posible tener un proceso de usuario que hiciera lecturas/escrituras asíncronas?

- i) ¿Qué información debemos devolver en el IOFin?.

4. (2 puntos) Memoria dinámica

El señor BakaBaka define una región de memoria así:

```
char memory_region[8192];
```

Y el compilador le asigna la dirección 0xCC000. Sobre esta región de memoria implementa un algoritmo de *Buddy Allocator* para disponer de memoria dinámica. En concreto quiere esta memoria dinámica, **única y exclusivamente**, para crear y destruir 2 tipos de estructuras diferentes (A y B) que ocupan 42 y 666 bytes respectivamente. Para facilitar su trabajo añade 2 rutinas *newA()* y *newB()* para crear una nueva petición de memoria para cada estructura. Responde a las siguientes preguntas justificadamente:

- a) Suponiendo la siguiente secuencia de llamadas:

```
p = newA();  
q = newA();  
r = newB();  
s = newA();
```

Indica la cantidad de memoria libre disponible.

SO2

Nombre:

DNI:

- b) ¿Cuántos elementos tipo A se podrían crear después de la secuencia anterior?

- c) Suponiendo que el algoritmo siempre asigna las direcciones más bajas primero. ¿Cuáles serán las direcciones asignadas a p, q, r y s?

- d) En el peor caso, ¿qué porcentaje de esta región de memoria no se podría usar por culpa de la fragmentación interna?