

Nombre:

DNI:

Segundo control de teoría

Contesta todas las preguntas justificando brevemente tu respuesta. Una respuesta sin justificar se considerará como incorrecta.

1. (4 puntos) Preguntas cortas

- a. Enumera las estructuras relativas al sistema de ficheros para almacenar la información dinámica de los dispositivos lógicos

- b. Explica cuál es el objetivo principal de tener dispositivos lógicos en un sistema de ficheros

- c. Explica, brevemente, cual es la función principal de un Inodo

- d. Enumera, justificándolo brevemente, que estructuras del sistema de ficheros se modifican cuando se ejecuta la llamada al sistema write.

- e. Explica, brevemente, para qué sirve la llamada al sistema mknod.

SOA2 (28/05/2021)

Nombre:

DNI:

- f. En un sistema operativo que solamente puede ejecutar un proceso (como msdos), ¿qué estructuras del sistema de ficheros no son necesarias?

- g. ¿Por qué toda la gestión de semáforos se tiene que hacer con llamadas al sistema?

- h. Describe, brevemente, cual es el uso estándar de un semáforo, dependiendo cómo se inicializa su contador en la llamada `sem_init`.

- i. ¿Por qué la estructura que guarda los `task_struct` en ZeOS tiene que declararse con un vector estático?

- j. ¿Qué información es propia del thread de un proceso?

2. (2 puntos) Sistema de ficheros:

Estamos diseñando un sistema de ficheros en el que sabemos que vamos a almacenar única y exclusivamente ficheros de 8192 bytes. El tamaño de un bloque de datos es de 1024 bytes, equivalente a 4 sectores de disco. El sistema de ficheros contiene 1024 Inodos y 4096 bloques de datos. Siempre se utilizarán Inodos para guardar la información relativa a un fichero y el número de Inodos será estático (siempre habrá 1024).

Indica, justificándolo, el número de ficheros máximo que podemos crear si el mecanismo de asignación de bloques a ficheros es:

SOA2 (28/05/2021)

Nombre:

DNI:

a) Continuo

b) Encadenado (sin tabla)

c) Indexado

d) Indexado multinivel

e) Indica, cual será el tamaño de los mapas de bits de Inodos y bloques de datos libres en esta partición.

3. (4 puntos) /dev/null

Queremos implementar el dispositivo lógico /dev/null. Este dispositivo lógico es un “agujero negro”: todo lo que se escribe en él, no se almacena en el sistema de ficheros sino que desaparece. Las lecturas a /dev/null siempre devuelven final de fichero (es decir, read

SOA2 (28/05/2021)

Nombre:

DNI:

devuelve 0). Las escrituras siempre devuelven el número de bytes escritos. Por ahora supondremos que existe un gestor asociado.

La sucesión de llamadas a funciones que se hace hasta llegar al código del driver de /dev/null, para la llamada al sistema read, es:

- 1.- int sys_read(int fd, void *buffer, size_t size)
- 2.- int read_gestor(struct Inode *Inode, int offset, void *buffer, size_t size)
- 3.- int read_driver(struct opaque* data, int offset, void *buffer, size_t size)

a) Enumera las estructuras del sistema de ficheros a las que accede sys_read

b) Describe los campos que debería tener un iorb

c) ¿Cómo encuentra sys_read la dirección de la función “read_gestor”?

d) ¿Cuántos semáforos se necesitan para enviar una petición al gestor de /dev/null? Describe para qué sirve cada uno.

SOA2 (28/05/2021)

Nombre:

DNI:

e) Escribe el código del read dependiente del gestor de /dev/null.

```
int read_gestor(struct Inode* Inode, int offset, void *buffer, size_t size)
{

}

}
```

f) Escribe el código del write dependiente del gestor de /dev/null

```
int write_gestor(struct Inode* Inode, int offset, void *buffer, size_t size)
{

}

}
```

g) Escribe el código del read del driver de /dev/null

```
int read_driver(struct opaque* data, int offset, void *buffer, size_t size)
{

}

}
```

h) Escribe el código del write del driver de /dev/null

```
int write_driver(struct opaque* data, int offset, void *buffer, size_t size)
{

}

}
```

SOA2 (28/05/2021)

Nombre:

DNI:

- i) Escribe el código del gestor de /dev/null

- j) Dado el funcionamiento de este dispositivo, ¿sería necesario tener un gestor? En caso negativo indica qué funciones de las anteriores se tendrían que modificar y cómo se modificarían.