

Nombre:

DNI:

---

---

## Segundo control de laboratorio

### Grupo 11

Crea un fichero que se llame “respuestas.txt” donde escribirás las respuestas para los apartados de los ejercicios del control. Indica para cada respuesta, el número de ejercicio y el número de apartado (por ejemplo, 1.a).

**Importante:** para el primer ejercicio tienes que partir de la versión de Zeos original que te hemos suministrado.

#### 1. (4 puntos) Zeos Scheduler

Queremos modificar el planificador de ZeOS para que modifique el quantum de los procesos teniendo en cuenta el número de llamadas al sistema. Así, un proceso que realice menos de 4 llamadas al sistema por tick de reloj, verá como se le decrementa el quantum en 1 tick de reloj. El proceso que invoque al sistema 4 o más veces por tick de reloj, se le incrementará el quantum en 1 tick de reloj. El cálculo de llamadas al sistema por tick de reloj solamente se tiene en cuenta durante un quantum, es decir, se resetearán cuando el proceso salga de la CPU. El quantum mínimo que puede tener un proceso es de 1 tick y el máximo es de 5. Un proceso heredará el quantum de su padre en fork.

Responde a los siguientes apartados:

- a) Indica que modificaciones se tienen que realizar del task\_struct de un proceso
- b) Indica como tienes que modificar el código de ZeOS para actualizar las estadísticas de llamadas al sistema por tick de reloj para un proceso
- c) Indica en qué punto del código se modificará el quantum del proceso y qué código utilizarás para calcular el nuevo quantum
- d) Implementa en Zeos el mecanismo descrito.

#### 2. (6 puntos) Sockets

Queremos implementar un servidor centralizado de copias de seguridad que se llamará *bserver*.

Este servidor estará esperando conexiones en el puerto indicado a través de la línea de comandos:

```
> bserver port_number
```

Lo que hace este servidor es guardar en disco los ficheros que le envíen los clientes a través de este puerto. Los clientes tienen la siguiente sintaxis:

```
> bclient port_number filename
```

# SOA (28/05/2019)

Nombre:

DNI:

Siendo *port\_number* el número de puerto de *bserver* y *filename* el nombre del fichero que se quiere enviar.

Implementa:

- a) Indica el conjunto de llamadas al sistema que tiene que realizar *bserver* hasta que acepta conexiones por el socket creado
- b) Indica el conjunto de llamadas al sistema que tiene que realizar el *bclient* para conectarse al servidor
- c) Escribe el pseudocódigo del bucle principal del *bserver*, en el caso de que éste procese las peticiones secuencialmente, desde que acepta la petición hasta que acaba de escribir el fichero en disco.
- d) Escribe el pseudocódigo del bucle principal del *bclient* en el que envía el fichero al servidor
- e) Indica qué cambios se tienen que hacer en el *bserver* del apartado anterior para que procese un número limitado de peticiones de forma paralela mediante procesos hijos.
- f) Implementa un *bserver* secuencial, un *bclient* y un *bserver\_extra* que acepte las peticiones en paralelo mediante procesos hijos.

### 3. (1 punto) Generic Competences Third Language (Development Level: mid)

The following paragraph belongs to the book *Understanding the Linux Kernel* by D. Bovet and M. Cesati:

*“Memory allocation requests can be satisfied in two different ways. If enough free memory is available, the request can be satisfied immediately. Otherwise, some memory reclaiming must take place, and the kernel control path that made the request is blocked until additional memory has been freed. However, some kernel control paths cannot be blocked while requesting memory— this happens, for instance, when handling an interrupt or when executing code inside a critical region. In these cases, a kernel control path should issue atomic memory allocation requests. An atomic request never blocks: if there are not enough free pages, the allocation simply fails.”*

Create a text file named “generic.txt” and answer the following questions (since this competence is about text understanding, you can answer in whatever language you like):

1. How can a memory request be satisfied?
2. In which situations a memory request will be considered as critical?
3. In a critical memory request, what is the default behavior of the kernel if not enough memory is available?

### 4. Entrega

Sube al Racó los ficheros “respuestas.txt”, “generic.txt” y el código que hayas creado en cada ejercicio.

# SOA (28/05/2019)

---

Nombre:

DNI:

---

---

Para entregar el código, si lo has hecho, de cada ejercicio utiliza:

```
> tar zcfv dni.tar.gz directorio_ejercicio1 directorio_ejercicio2
```