

Nombre: _____

DNI: _____

Primer control de laboratorio

Justifica todas tus respuestas. Una respuesta sin justificación se considerará errónea.

Importante: para cada uno de los ejercicios tienes que partir de la versión de Zeos original que te hemos suministrado.

1. (6 puntos + 0.5 puntos por implementación) fork2

En el fichero zeos.tar.gz encontrarás una implementación funcional de ZeOS, en la que queremos implementar una nueva llamada a sistema que nos permita crear dos procesos a la vez (en lugar de 1 como lo que hemos visto en clase). Los procesos deben ser copias exactas del proceso actual y continuar su ejecución justo después de esta llamada.

```
int fork2(int pids[2]);
```

Esta llamada a sistema recibe como **parámetro** un vector de 2 posiciones para retornar los pids de los procesos creados, que el sistema rellenará:

- 1) en el caso del padre, con los pids de los 2 procesos creados y
- 2) en el caso de los hijos, con un zero(0) para el caso del hijo y con un -1 para el otro hijo.

El **valor de retorno** de esta llamada será -1 si no ha podido crear los 2 procesos, o 0 si lo ha conseguido.

Ejemplo de uso:

```
int pids[2];

p = fork2(pids);

if ( p < 0 ) { perror(); exit(-1); }
if ( pids[0] == 0 ) {
    // Código del primer hijo (&& pids[1]==-1)
} else if ( pids[1] == 0 ) {
    //Código del segundo hijo (&& pids[0]==-1)
} else {
    // Código del padre (pids[0]>0 && pids[1]>0)
}
```

Para implementar esta nueva funcionalidad usaremos la entrada 190 de la tabla de interrupciones, dado que queremos separar las llamadas a sistema habituales de las llamadas a sistema originadas por enunciados surrealistas como éste. Además no se puede llamar directamente a la función de sistema sino que hay que crear una tabla (similar a la que has usado para crear las llamadas a sistema hasta ahora) que nos permita llamar ahora a la función de sistema *fork2* y en el futuro añadir nuevas funciones sin demasiadas complicaciones (sólo modificando esta tabla). Se debe usar la entrada 4 de esta tabla.

SO2

Nombre:

DNI:

- a) Indica el código ensamblador necesario en el wrapper para invocar la llamada a sistema *fork2*.

- b) Código del handler.

- c) Código para definir la nueva tabla

SO2

Nombre:

DNI:

- d) ¿Hay que hacer algo durante la inicialización del sistema? ¿Por qué? (En caso afirmativo, copia el código que has añadido/modificado)

- e) Muestra el código necesario para actualizar la variable pids desde el proceso padre y en qué condiciones debe ejecutarse.

- f) Muestra el código necesario para actualizar la variable pids para el 1r proceso hijo y en qué condiciones debe ejecutarse.

SO2

Nombre:

DNI:

g) Muestra el código necesario para devolver el valor de retorno correcto en el padre

h) Muestra el código necesario para preparar el contexto del 2 hijo

i) ¿Qué valor retorna el 2º hijo?¿Cómo?

j) (opcional) Implementa esta nueva llamada a sistema.

SO2

Nombre:

DNI:

2. (4 puntos + 0.5 puntos) Gestión de procesos

- a) (1 puntos) Queremos modificar la planificación de procesos para que se ejecute el siguiente proceso pendiente de ejecutar cada vez que se pulse una tecla. ¿Qué rutina/s tienes que modificar? ¿Cómo?

- b) (3 puntos) Modifica el código necesario del fork para que realice la herencia de datos de usuario usando la región de memoria 0x150000-0x151000 en lugar de la región actual.

- c) (opcional) Implementa los apartados anteriores.

Entrega

Sube al Racó , si lo has hecho, el código que hayas creado en cada ejercicio. Para ello utiliza:

```
> tar zcfv ejercicioX.tar.gz zeos
```