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) q3mailbox

En el fichero zeos.tar.gz encontrarás una implementación funcional de ZeOS, en la que queremos implementar un mecanismo simple de comunicación entre procesos. Para ello hay que crear un buzón (el q3mailbox) y el nuevo mecanismo tiene que permitir a un proceso enviar un mensaje a este buzón. A partir de este momento, otro proceso puede recibir este mensaje recuperando el contenido del buzón. El buzón sólo puede contener un único mensaje y su tamaño está limitado a MAX bytes (20).
Este buzón se inicializará al inicio del sistema y a partir de este momento puede recibir y enviar mensajes.
En concreto hay que implementar dos llamadas a sistema nuevas:

*int q3send(char *buffer, int size)*

Envía un mensaje al buzón. Esta llamada copia *size* bytes del mensaje *buffer* en el buzón. Si size > MAX, los byte por exceso se ignoran.
El retorno de esta función es:

- 0 si el mensaje se ha escrito en el buzón, o
- <0 en caso contrario y, según el caso, *errno* contendrá:
  - EAGAIN: el buzón todavia contiene un mensaje sin leer
  - EINVAL: parámetro *size* <0
  - EFAULT: *buffer* fuera del espacio de direcciones del proceso

*int q3recv(char *buffer, int* size)*

Lee un mensaje del buzón, dejándolo libre para recibir otros mensajes. El contenido del mensaje es copiado en el buffer de usuario *buffer* y notifica el tamaño leído mediante el parámetro de salida *size* (NOTA: el parámetro *size* que le pasa el usuario debe contener el tamaño máximo de *buffer* y, por lo tanto, todos los bytes del mensaje que superen este límite deben ser descartados).
El retorno de esta función es:

- 0 si se ha leído el mensaje del buzón, o
- <0 en caso contrario y, según el caso, *errno* contendrá:
  - EAGAIN: el buzón está vacio
  - EFAULT: *buffer* o *size* fuera del espacio de direcciones del proceso

En el Código 1 puedes ver un ejemplo de uso de estas llamadas a sistema.

Para implementar estas nuevas funcionalidades del sistema no se debe usar el mecanismo habitual de llamadas a sistema, sino que hay que usar las entradas 60 y 61 de la tabla de interrupciones para *q3send* y *q3recv* respectivamente.

Nombre:
DNI:

```
int x;
if (fork() > 0) { // padre
        char *b = "hola";
        do {
                x = q3send(b, strlen(b));
        } while (x!=0);  //Condiciones de error NO comprobadas
} else { //hijo
        int s;
        char b[5];
        do {
                s = 5;
                x = q3recv(b, &s);
        } while (x!=0);
        write(1, b, strlen(b));
}
```

Código 1 Ejemplo de uso de las nuevas llamadas a sistema. Este código crea 2 procesos, el padre envia un mensaje "hola" y el hijo lo lee.

a) Indica el código ensamblador necesario en el wrapper para invocar la llamada a sistema q3send.

b) Código del handler para q3recv.

Nombre:
DNI:

c) ¿Qué estructura/s de datos tenemos que utilizar dentro de sistema para representar un buzón? Justifica cada estructura y cada uno de sus campos.

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

e) Sabiendo que "char *data" es una variable que apunta a la zona de memoria donde el buzón guarda el mensaje, completa el código necesario en la rutina sys_q3send para copiar en esta variable el contenido del parámetro "char *buffer".

```
int sys_q3send(char *buffer, int size) {
	...
	/* At this point all the parameters have been checked and
	everything is correct. The next thing to do is to copy the
	buffer to the mailbox. */
```

Nombre:
DNI:

f) Copia el código necesario del sys_q3send para el caso en que tiene que devolver un EAGAIN

g) (opcional) Implementa esta nueva funcionalidad (y sube al Racó el directorio Zeos con ´tar zcvf q3mailbox.tar.gz zeos´)

2. (4 puntos + 0,5 puntos de implementación) Gestión de procesos

a) (1 puntos) ¿No seria necesario añadir la instrucción set_cr3() en la rutina sys_exit después del primer bucle? ¿Por qué?

b) (3 puntos) Modifica el código necesario del fork para que realice la herencia de datos de usuario usando la región de memoria 0x150000-0x170000 en lugar de la región actual. [Si lo implementas sube al Racó el directorio zeos con ´tar zcvf fork.tar.gz']