

---

Nombre:DNI:

---

---

## Primer control de laboratorio

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).

**Justifica brevemente todas tus respuestas. Una respuesta sin justificar se considerará como no contestada.**

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

### 1. (5,5 puntos) Mecanismos Entrada al Sistema

Tienes que implementar una nueva llamada a sistema para que el usuario pueda decidir la posición exacta donde escribir sus mensajes en la pantalla. La sintaxis de esta llamada es:

*int gotoxy(int x, int y)*

Esta función modificará los valores de las variables globales 'x' e 'y' del sistema con los valores pasados como parámetros. De esta forma modificamos el comportamiento de la rutina *printc* que a partir de este momento escribirá en esta nueva posición. En caso de error debe devolver un -1 y actualizar la variable global *errno* con un 101 si el primer parámetro no es válido o un 102 si no lo es el segundo. Un ejemplo de uso sería:

```
gotoxy( 1,1 );  
write(1, "hola", 4); //escribe en la linea 1  
gotoxy( 1,23 );  
write(1, "adios", 5); //escribe en la linea 23
```

La consola de texto dispone de 80 columnas y 25 líneas. Como restricción, para implementar esta nueva llamada a sistema no debes usar el mecanismo habitual, sino que usarás la interrupción 0x81 para implementar esta funcionalidad en exclusiva.

Modifica el código de ZeOS para añadir esta funcionalidad y responde a estas preguntas:

- (1 puntos) ¿En qué dirección de memoria se encuentra la variable global 'x'? ¿Cómo la has encontrado?
- (0.5 puntos) ¿Qué estructura de datos de ZeOS tienes que modificar?
- (0.5 puntos) ¿En qué rutina de ZeOS la modificas?
- (0.5 puntos) ¿Es necesario el EOI?

Nombre:

DNI:

- e) (0.75 puntos) ¿Es necesario hacer el handler tal y como hemos visto en clase? ¿Cuales son las partes prescindibles?
- f) (0.5 puntos) Escribe el código del wrapper de esta funcionalidad.
- g) (0.5 puntos) Escribe el código del handler de esta funcionalidad.
- h) (0.5 puntos) Escribe el código de la rutina de esta funcionalidad.
- i) (0.75 puntos) ¿Qué contendrá la pila de sistema en el momento de ejecutar la primera instrucción de la rutina del último apartado?

## 2. (4 puntos) Gestión de procesos y memoria

Modifica el código del fork para que el proceso padre use solamente la región de memoria 0x200000-0x214000 para acceder a la zona de datos del hijo y realizar la herencia de datos de usuario, en lugar de usar la región consecutiva a la zona de datos del padre.

- a) (0.5 puntos) ¿Cuántas páginas tiene esta región temporal? (SIZE)
- b) (0.5 puntos) ¿Cuál es la primera dirección de la zona de datos de un proceso de usuario? (START\_DATA)

Suponiendo que el bucle de copia fuera tal que así:

```
for(i=0; i < NUM_PAG_DATA; i++) {  
    set_ss_page(...[1]...  
    copy_data(...[2]...  
    del_ss_page(...[3]...  
    ...[4]...  
}
```

Muestra como seria el código para hacer esta copia. Puedes usar las variables SIZE y START\_DATA de los apartados anteriores, así como usar otras variables/funciones de ZEOS o nuevas que creas convenientes siempre y cuando lo indiques y muestres su inicialización.

- c) (0.75 puntos) Indica el código de [1]
- d) (0.75 puntos) Indica el código de [2]
- e) (0.75 puntos) Indica el código de [3]
- f) (0.75 puntos) Indica el código de [4]

## 3. (1 punto) Espacio de direcciones

La rutina init\_table\_pages, inicializa las tablas de página de los procesos.

- a) Indica el rango de direcciones accesibles (inicial – final) si la MMU usara una de estas tablas tras ejecutar esta rutina.
- b) ¿Son accesibles estas direcciones por un proceso de usuario? Caso afirmativo, indica qué deberías hacer para que no lo fueran. Caso negativo, indica que deberías hacer para que lo fueran.