Práctica 3: (Tema 2) Gestión de interrupciones en MaRTE OS

- **Objetivos:**
  - **Practicar la gestión de interrupciones en MaRTE OS**
  - **Experimentar las diferencias entre los dos modelos de gestión de interrupciones de MaRTE OS**

Modificación del driver de la práctica anterior

Utilizar la nueva versión del puerto de comunicaciones simulado:

- **ficheros `fake_com_port_intr.c` y `fake_com_port_intr.h`**

El dispositivo se controla mediante dos registros en el espacio de direcciones de I/O:

- **Registro de control: FAKE_COM_PORT_CONTROL_REG (`0x100`)**
- **Registro de datos: FAKE_COM_PORT_DATA_REG (`0x101`)**

El dispositivo genera la interrupción FAKE_COM_PORT_IRQ

- **Cuando ha llegado un nuevo dato**

Funcionalidad de los registros del puerto de comunicaciones

Write any value on FAKE_COM_PORT_CONTROL_REG:

- **the communication port is initialized**
- **an interrupt is generated as soon as a new byte is received**

Read FAKE_COM_PORT_CONTROL_REG:

- **`0x01` when there is new data available and `0x00` in other case**

Read FAKE_COM_PORT_DATA_REG:

- **Returns the last byte received**

Escribir un driver para el puerto de comunicaciones anteriormente descrito

***create*:**

- **retorna 0**

***open*:**

- **inicializa el dispositivo**

***read*:**

- **permite leer los datos que llegan al puerto**
- **utiliza interrupciones: si no hay dato disponible, el thread se bloquea a la espera de la interrupción**

***remove*, *write* y *close*:**

- **no es necesario definirlas**

Dos versiones del driver

Crear dos versiones del driver que se diferencien en modelo de atención a interrupciones utilizado

Versión que utiliza `posix_intr_timedwait`

- **`read`: cuando no hay datos disponibles, el thread que llama se queda bloqueado en la llamada a `posix_intr_timedwait`**

Versión basada en semáforos

- **open: crea un semáforo**
- **`read`: cuando no hay datos disponibles, los threads que llaman se quedan bloqueados en la llamada a `sem_wait`**

Probar a utilizar las dos versiones del driver desde uno o varios threads de forma simultanea