Práctica 2: (Tema II-4) Driver sencillo en MaRTE OS

- **Objetivos:**
  - **Instalar MaRTE OS (ya hecho en Prog. Concurrente)**
  - **Practicar el uso del entorno de desarrollo de MaRTE OS en el laboratorio (ya hecho en Prog. Concurrente)**
  - **Practicar la instalación y el uso de drivers en MaRTE OS**

Laboratorio e Instalación de MaRTE OS

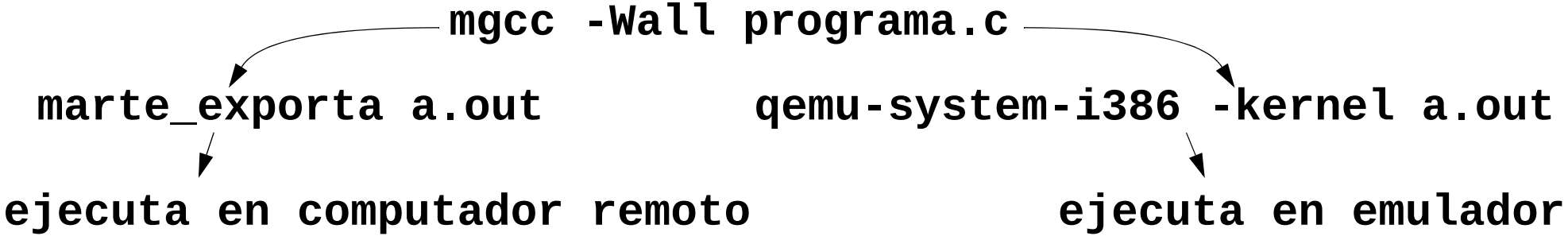
Acceso a los ordenadores del laboratorio:

- **Elegir la última de las opciones de arranque:**
  `Debian GNU/Linux, with Linux 2.6.32-5-686 (on /dev/sda6) DEC`
- **Usuario:** `Usuario de MaRTE OS y GNAT GPL de AdaCore`
- **Contraseña:** `marteos`

Instalar MaRTE OS en el directorio del usuario:

- **compilador GNAT y MaRTE OS en el directorio** `labCTR/`
- **instrucciones en el fichero** `INSTALL` **de MaRTE**
- **¿Ya hecho en Programación Concurrente?**

Compilar y ejecutar un programa:



Instalación de un driver en MaRTE OS

Se pretende escribir e instalar un *driver* para un puerto de comunicaciones

- **se trata de un *dispositivo simulado*: ficheros `fake_com_port.c` y `fake_com_port.h`**
- **Deberemos compilar estos ficheros junto con el *driver***

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`)**

Interfaz del dispositivo simulado (fake_com_port.h)

```
#define FAKE_COM_PORT_CONTROL_REG 0x100
#define FAKE_COM_PORT_DATA_REG 0x101

/*
 * Depending on the value of port:
 * FAKE_COM_PORT_CONTROL_REG: the communication port is
 *                            initialized
 * Any other value: no effect
 */
extern void fake_com_port_outb(unsigned short port,
                               char val);

/*
 * Depending on the value of port:
 * FAKE_COM_PORT_DATA_REG: Returns the last byte received
 * FAKE_COM_PORT_CONTROL_REG: returns 0x01 when there is
 *              new data available and 0x00 in other case
 * Any other value: returns 0x00
 */
extern char fake_com_port_inb(unsigned short port);
```

Funcionalidad de los registros del puerto de comunicaciones

Write any value on FAKE_COM_PORT_CONTROL_REG:

- **the communication port is initialized**

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`: utiliza *pooling* para leer los datos que llegan al puerto**

```
ssize_t driver_read (int fd, void *buffer, size_t bytes)
   i=0
   mientras i<bytes hacer
      si hay dato disponible entonces
         buffer[i] = dato leído del dispositivo
         i++
      fin si
   fin mientras
   retorna bytes
```

- **`remove`, `write` y `close`: no es necesario definirlas**