

Arquitetura de Computadores 2022/23

Ficha 10

Tópicos: Programação da Porta Série com Interrupções.

Introdução

O ambiente de execução desta aula é o mesmo da aula anterior pelo que se aconselha a leitura do enunciado da ficha 9.

Na arquitetura que inicialmente se designava IBM PC (hoje apenas designada PC), a comunicação com as portas série (COM1 e COM2) é realizada por um circuito (chip) designado UART: Universal Asynchronous Receiver-Transmitter; para interagirmos com a porta série COM1 temos de “programar”/aceder aos seguintes registos:

Porto 0x3F8 – THR (Transmission Hold Register) – registo de dados de envio. Contém o carácter a ser enviado pela porta série.

Porto 0x3FC – MCR (Modem Control Register) – registo de controle da porta série

Porto 0x3F9 – IER (Interrupt Enable Register) – registo de ativação das interrupções na porta série

Estes registos podem ser acedidos através das funções `inportb` e `outportb` referidas na ficha da aula anterior. Por exemplo:

```
#include <dos.h>
#define MCR 0x3fc
...
b = inportb( MCR ); /* ler registo MCR */
b = b | 1; // colocar bit 0 a 1
outport(MCR, b);
```

Trabalho a realizar

Enviar um ficheiro para a porta série utilizando o mecanismo de interrupções. No clip está o ficheiro `ficha10.zip` com a implementação parcial deste programa, o qual recorre aos seguintes ficheiros fonte:

- **escint.c** programa principal
- **bufcir.h** e **bufcir.c** implementação de um buffer circular
- **int.prj** ficheiro projecto do TurboC que define que ficheiros compõem a aplicação, ou seja, que ficheiros devem dados ao linker para este produza o ficheiro executável final

O programa deve colocar os caracteres no buffer circular. Ao mesmo tempo, a rotina de tratamento de interrupções vai retirar os caracteres desse buffer e enviá-los para a porta série. A rotina de tratamento de interrupções será chamada através do mecanismo hardware das interrupções, sempre que as interrupções da porta série estiverem ligadas e o registo de transmissão da UART (o controlador da porta série) estiver livre.

Nota 1: como o buffer pode ser acedido em qualquer altura pela rotina de tratamento de interrupções, o programa principal deve acedê-lo com as interrupções do CPU desligadas (com a *interrupt flag* a 0).

Nota 2: quando a rotina de tratamento de interrupções é chamada, as interrupções são desligadas. Devemos ligá-las o mais cedo possível, pois podem acontecer interrupções mais importantes (de nível mais elevado) que seja importante tratar imediatamente.

Nota 3: qualquer rotina de tratamento de interrupções deve terminar enviando o comando EOI - End of Interrupt – com o valor 0x20 para o registo de comando (porto 0x20) do PIC.

ANEXO

Programação das interrupções

Sugerimos que implemente funções que permitam ligar e desligar as interrupções da porta série e ligar e desligar o atendimento de interrupções da linha IRQ4 do PIC.

1. Ligar interrupções da porta série

Na UART é necessário:

- colocar o bit 3 do registo MCR a 1, para permitir as interrupções da UART;
- colocar o bit 1 do registo IER a 1, para permitir gerar interrupções após o envio de um carácter;

No PIC é necessário:

- colocar o bit 4 do registo máscara (porto 0x21) a 0, de modo a deixar passar as interrupções que vêm da porta série.

Nota: a alteração do registo máscara do PIC deve ser efetuada com as interrupções do CPU desligadas (i.e. com a *interrupt flag* a 0).

2. Desligar interrupções da porta série

Na UART é necessário:

- colocar o bit 3 do registo MCR a 0 e o bit 1 do registo IER a 0.

No PIC é necessário:

- colocar o bit 4 do registo máscara (0x21) a 1.

Programação de entradas/saídas em TurboC

Endereços de I/O:

- UART (COM1):
 - Registo de Dados de Escrita (THR): 0x3F8
 - Registo de Dados de Leitura (RBR): 0x3F8
 - Registo de Controlo das Interrupções (IER): 0x3F9
 - Registo de Controlo do Modem (MCR): 0x3FC

- Registo de Estado (LSR): 0x3FD
- PIC:
 - Registo de Comandos: 0x20
 - Registo de Máscara: 0x21

Comandos do PIC:

- Fim de Interrupção (EOI): 0x20

Programação de interrupções no TurboC:

Deve incluir o ficheiro de cabeçalho **dos.h**

- **enable()** - função do TurboC para ligar as interrupções no CPU (coloca interrupt flag a 1);
- **disable()** - função do TurboC para desligar as interrupções no CPU (coloca interrupt flag a 0);
- Para colocar na mesma entrada o endereço da uma rotina de tratamento de interrupções **setvect()**

```
#include <dos.h>

...

void interrupt handlingRoutine(){
    ...
}

...

void some_function() {
    ...
    setvect(12, handlingRoutine);
    ...
}
```

Para mais detalhes ver o ficheiro `int.pdf` no CLIP.