

**Nota:** este trabalho é só para quem terminou o trabalho da aula anterior - não avance antes de terminar esse trabalho.

### 1. Ligação série entre duas máquinas virtuais

O qemu permite ligar a porta série da máquina virtual a um "named pipe". Um "named pipe" é um dispositivo FIFO, no qual se podem escrever bytes, que depois são lidos pela mesma ordem.

Os comandos necessários são os seguintes:

#### Criar "named pipe" no linux:

```
mkknod linha-serie p
```

#### Lançar o qemu ligando a porta serie ao named pipe:

```
qemu fdos-10meg.img -serial pipe:linha-serie
```

Se lançar duas máquinas virtuais com este comando (em dois terminais), elas vão ficar ligadas através de uma linha série, simulada pelo "named pipe" linha-serie.

Experimente enviar o ficheiro numa das máquinas (com o programa da aula 11) e receber na outra (com o programa da aula 10). Não esqueça que é preciso enviar o **Ctrl-D** para marcar o fim do ficheiro!

### 2. Recepção com interrupções

Pretende-se fazer um programa para receber um ficheiro pela porta série a através de interrupções. Siga o pseudo código que se segue. Para ver os valores das constantes (**RBR**, **MCR**, **IER**, **PIC\_COMMAND**, **PIC\_MASK**, **EOI**, etc), consulte a documentação das aulas anteriores.

#### Programa principal:

```
Inicializar vector de interrupções com RX_Serie
Ligar interrupções de recepção na UART
Ligar interrupções da porta série no PIC
Abrir ficheiro
Repetir
    c = ReceberPortaSerie
    escrever c no ficheiro
Até c = Ctrl-D
Fechar ficheiro
Desligar interrupções da porta série no PIC
Desligar interrupções de recepção na porta série
```

#### ReceberPortaSerie:

```
Enquanto Buffer estiver vazio
    esperar
c = tirar caracter o buffer
retornar c
```

# Arquitectura de Computadores 2010/2011

## Aula prática 12– Interrupções e dispositivos de I/O - 2

**interrupt RX\_Serie** (rotina de atendimento da interrupção):

c = Ler byte do registo de dados (**RBR**)

Se buffer estiver cheio

Perdemos o caracter c

Senão

Por c no buffer

Escrever código **EOI** (end of interrupt) no **PIC\_COMMAND**

Ligar interrupções de recepção na UART:

Por bit 0 do registo **IER** a 1

Por bit 3 do registo **MCR** a 1

Ligar interrupções da porta série no PIC:

Por interrupt flag a 0 (disable)

Por bit 4 do registo **PIC\_MASK** a 0

Por interrupt flag a 1 (enable)

Desligar interrupções de recepção na UART:

Por bit 0 do registo **IER** a 0

Por bit 3 do registo **MCR** a 0

Desligar interrupções da porta série no PIC:

Por interrupt flag a 0 (disable)

Por bit 4 do registo **PIC\_MASK** a 1

Por interrupt flag a 1 (enable)

**Nota:** este programa escreve o o caracter de terminação (Ctrl-D) no ficheiro. Acha isso bem? Como corrigiria o problema?