Arquitectura de Computadores 2009/2010

Aula prática 10 – Programação de dispositivos de I/O

Pretende-se realizar programas que enviem caracteres através da porta série da porta série do PC. Será utilizado o TurboC a correr no sistema operativo FreeDOS, emulado pelo software de virtualização *qemu*.

O Simulador de PC Qemu e o TurboC

Como sabe, as instruções máquina de entrada/saída são privilegiadas e portanto, apenas acessíveis em modo supervisor. Para contornar este problema, durante o desenvolvimento sobre o sistema Linux, vamos fazer correr os nossos programas sobre um simulador chamado **qemu**.

Este simulador simula fielmente uma arquitectura hardware de um PC com um processador Intel e todo o hardware que o rodeia - portas série e paralela, controlador de interrupções, etc. Esta simulação é feita por software pelo que não é de esperar um grande desempenho.

A emulação feita pelo **qemu** vai permitir, para os nossos trabalhos, correr o sistema de operação **FreeDOS** que, como o nome indica, é uma reimplementação do **MS-DOS**, em que todo o código do kernel, interpretador de comandos e comandos, são de domínio público.

Trata-se de um sistema de operação para 16 bits e deixa que qualquer programa tenha total controlo da máquina (execute qualquer instrução). Posteriormente podemos executar este sistema, ou o MS-DOS, directamente sobre a máquina real e testar o nosso programa em ambiente real. Sobre o **FreeDOS** vamos correr o compilador **TurboC** (TC) e o assembler **TurboAssembler** (TASM).

1. Teste do qemu com o disco FreeDOS

Existe uma imagem de um disco do **FreeDOS**, com 10MB, no CLIP, secção *Outros*. Descarregue a imagem e descompacte-a com o comando:

```
gunzip fdos-10meg.img.gz
```

Agora já pode invocar o simulador, indicando como disco de boot o ficheiro com a respectiva imagem:

```
qemu fdos-10meg.img
```

Execute alguns comandos do DOS (cd, dir, etc). Experimente invocar o **TurboC** (tc).

Para terminar a simulação feche a janela do simulador (o que corresponderia a desligar o PC).

2. mtools - Copia de ficheiros do Linux para o disco FreeDOS

É possível transferir os ficheiros da nossa directoria para o disco do FreeDOS. Para isso, assegure-se primeiro que o simulador **não** está a correr.

Na sua **homedir** crie um ficheiro com o nome **.mtoolsrc**. Esse ficheiro não vai aparecer na listagem da directoria, visto que o nome começa com **'.'**. O ficheiro deve conter apenas uma linha, semelhante à seguinte:

```
drive Z: file="fdos-10meg.img" partition=1 nolock
```

Após criar o ficheiro já pode copiar ficheiros de e para o disco do FreeDOS a partir do Linux, usando os comandos das **mtools**. Por exemplo:

```
mdir Z:
mcd Z:work
mdir Z:
mcopy inverte.c Z:
```

etc...

Notas importantes:

Os ficheiros do DOS estão limitados a nomes de 8 caracteres e extensão de três, respeite esta convenção para evitar problemas com o **FreeDOS**.

Os ficheiros de texto do Linux têm uma convenção de mudanças de linha diferente dos do DOS. Antes de copiar os ficheiros para o disco do **FreeDos** deve corrigir as mudanças de linha com o seguinte comando:

```
unix2dos < inverte.c > invdos.c
```

3. Simulação da porta série com o qemu

Para a emulação da porta série proceda aos seguintes passos:

i. Simular a porta série, executando o comando na forma:

```
qemu -serial stdio fdos-10meg.img
```

O simulador irá fazer surgir no terminal tudo o que os seus programas internamente enviarem para a porta série. Tudo o que for escrito pelo utilizador no terminal, vai surgir como estando a entrar pela porta série da máquina virtual.

ii. Configurar a porta série no FreeDOS:

```
mode com1:9600,n,8,1
```

O comando **mode** do DOS configura os parâmetros de funcionamento da porta série especificada, neste caso o **com1** (velocidade, paridade, tamanho da palavra, número de stop bits).

iii. Testar a porta série:

```
debug
=o 3f8 41
=q
```

O comando **debug** do DOS permite ler e escrever nos portos de I/O. Neste caso deve aparecer o carácter A (ascii 41H) na porta série, que é simulada pela consola onde foi lançado o qemu.

Programação da porta série com espera activa

O TurboC disponibiliza as seguintes funções para aceder aos portos de I/O:

```
/*le um byte no porto de I/O com endereço port*/
int inportb(int port);

/*escreve um byte no porto de I/O com endereço port */
void outportb(int port, unsigned char byte);
```

estas funções usam as instruções máquina **IN** e **OUT** para acederem aos portos de I/O e estão disponíveis através do ficheiro *include* **<dos.h>**.

A porta série **COM1** disponibiliza os seguintes registos:

Porto 0x3F8 (escrita) – **THR** – carácter a ser enviado pela porta série

Porto 0x3F8 (leitura) – **RBR** – carácter recebido pela porta série

Porto 0x3FD (leitura) – **LSR** – estado da porta série. Este registo contém os seguintes bits (activos a 1):

Bit 0 – Chegou um byte à porta série que está disponível no registo RBR

Bits 1-4 – Erros na recepção

Bit 5 – Registo de transmissão (THR) livre. Podemos enviar um novo byte.

Bit 6 – Acabou de ser enviado um byte pela linha série.

Estes registos podem ser acedidos através das funções `inportb` e `outportb` acima indicadas. Por exemplo:

```
#include <dos.h>

#define LSR 0x3fd

...

b = inportb( LSR ); /* ler registo de estado */

if ( (b & 1) == 1) /* testar bit 0 */

      /* podemos ler um byte da porta série */
```

Trabalho a realizar

Pretende-se fazer um programa recebe um ficheiro pela porta série, guardando-o em disco.

1. Programe a função

   `char receber_porta_serie();`

   que retorna um byte lido da porta série.
   Como sabe, o método da espera activa consiste em verificar repetidamente os bits do registo de estado, até ser possível enviar ou receber um byte.

2. Recorrendo à função `receber_porta_serie`, construa um programa constrói um ficheiro a partir de uma sequência de caracteres lidos da porta série. A sequência deve ser terminada pelo carácter **Ctrl-Z**.

Consulte o ficheiro **serie.pdf**, na secção Problemas do CLIP para saber mais detalhes sobre a programação da porta série do PC.