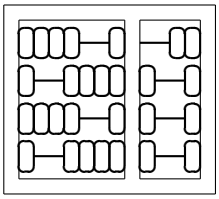

MC404 - Organização de Computadores e Linguagem de Montagem
Instituto de Computação
Universidade Estadual de Campinas

Lab. 05

1 Objetivo

Neste laboratório serão praticados os serviços do DOS para manuseio de arquivo e gerenciamento de múltiplos segmentos de dados em um programa em linguagem de montagem.

2 Preliminar

Como referência para as atividades deste laboratório por favor consulte a seção 6. Utilize os arquivos de testes disponibilizados na página do curso para testar os programas a serem desenvolvidos neste laboratório. Será necessário renomear os arquivos de acordo com as atividades a seguir.

3 Atividade 1

Escreva um programa em linguagem de montagem que leia um arquivo chamado `entrada.txt` e gere um arquivo chamado `saida.txt` com o mesmo conteúdo. Ou seja, seu programa deve fazer um cópia do arquivo de entrada.

4 Atividade 2

Altere o programa da atividade 1 para que o conteúdo copiado apareça em ordem inversa. Ou seja, o arquivo `saida.txt` deve ser uma cópia invertida do arquivo `entrada.txt`. Exemplo:

```
ENTRADA.TXT:              ==>  SAIDA.TXT:
   Hello World                      dlroW olleH
```

Considere o limite máximo de tamanho de arquivo como sendo 128Kbytes.

5 Atividade 3 (Opcional)

Escreva um programa que, dado um arquivo de entrada, gere um arquivo de saída com as linhas invertidas. Exemplo:

```
ENTRADA.TXT:              ==>  SAIDA.TXT:
   Linha 1                          Linha 2
   Linha 2                          Linha 1
```

6 Serviços do DOS para manuseio de arquivos

A maneira mais fácil de manipular arquivos no DOS é através do uso de *file handlers*. Um handler é um identificador do arquivo, e deve ser usado na chamada de todos os serviços do DOS que manipulem arquivos (com exceção do serviço que abre/cria o arquivo).

6.1 Abrindo um arquivo

Um handler é adquirido pela chamada do serviço `0x3D` de abertura de arquivo do DOS, apresentado logo abaixo:

```
Service 0x3D           Open File with Handle
-----------------------------------------------------------------
        Entry   AH    = 0x3D
                AL    = Open mode (access code)
                DS:DX = Pointer to ASCIIZ filename

        Return  AX = File handle, if CF clear
                or
                AX = Error code, if CF is set
                     | 0x02 File not found
                     | 0x03 Path not found
                     | 0x04 No handle available
                     | 0x05 Access denied
                     | 0x0C Open mode invalid
```

O registrador `AL` deve conter o modo de abertura:

```
0 - read-only access
1 - write-only access
2 - read/write access
```

Uma string ASCIIZ é uma string terminada com o caracter nulo (`0x00`).

6.2 Criando um arquivo

Funciona de forma semelhante ao serviço de abertura de arquivo, exceto que o arquivo não necessita existir previamente. Caso o arquivo exista, o mesmo é aberto e seu tamanho é truncado para 0 (ou seja, o conteúdo é perdido). Segue a descrição do serviço:

```
Service 0x3C           Create File with Handle
-----------------------------------------------------------------
        Entry   AH    = 0x3C
                CX    = File attributes
                DS:DX = Pointer to ASCIIZ filename

        Return  AX = File handle, if CF clear
                or
                AX = Error code, if CF is set
                     | 0x03 Path not found
                     | 0x04 No handle available
                     | 0x05 Access denied
```

Para fins práticos considere como atributo o valor `0x00` (`CX` = 0).

6.3 Fechando um arquivo

Para fechar um arquivo aberto, utilize o serviço `0x3E`:

```
Service 0x3E           Close File with Handle
-----------------------------------------------------------------
        Entry   AH = 0x3E
                BX = File handle

        Return  AX = Error code, if CF is set
                   | 0x06 Invalid handle
```

6.4 Lendo um arquivo

Para ler um arquivo aberto, use o serviço `0x3F`. Todo arquivo aberto contém um *file pointer*, que nada mais é do que um ponteiro para o próximo byte a ser lido do arquivo. Quando um arquivo é aberto, o ponteiro aponta para o primeiro byte do arquivo.

Após efetuada a leitura através do serviço `0x3F`, o DOS automaticamente atualiza o *file pointer*. Desta forma, uma segunda leitura adquire os bytes sucessores dos últimos bytes lidos na primeira leitura.

A descrição do serviço `0x3F` é apresentada a seguir:

```
Service 0x3F            Read File or Device
-----------------------------------------------------------------
        Entry   AH    = 0x3F
                BX    = Handle of file or device
                CX    = Number of bytes to read
                DS:DX = Address of buffer

        Return  AX = Number of bytes read into buffer
                or
                AX = Error code, if CF is set
                     | 0x05 Access denied
                     | 0x06 Invalid handle
```

Você pode especificar o número de bytes a serem lidos (registrador `CX`), e deve fornecer o endereço do buffer para onde os dados serão copiados (registradores `DS:DX`).

O registrador `AX` armazena o número de bytes lido para o buffer. Se no retorno do serviço a flag de carry (CF) estiver limpa, e `AX = 0`, então o *file handler* estava no fim do arquivo quando o serviço foi solicitado. Se a CF estiver limpa e `AX` for menor o que o número de bytes requisitados, então a leitura atingiu o final do arquivo.

Também é possível alterar o *file pointer* para uma posição específica dentro de um arquivo aberto. Isto é feito através do serviço `0x42`, apresentado a seguir:

```
Service 0x42            Move File Pointer (LSEEK)
-----------------------------------------------------------------
        Entry   AH    = 0x42
                BX    = File handle
                CX:DX = Offset, in bytes (32-bit integer)
                AL    = Mode code:
                         0   Move file pointer CX:DX bytes from
```

```
                        beginning of file (offset = un-signed value)
                    1   Move file pointer CX:DX bytes from
                        current location (offset = signed value)
                    2   Move file pointer CX:DX bytes from
                        end of file (offset = signed value)

        Return  DX:AX = New file position (unsigned 32-bit), from
                        start of file
                or
                AX = Error code, if CF is set
                     | 0x01 Invalid function (bad mode code)
                     | 0x06 Invalid handle
```

6.5 Escrevendo para um arquivo

Antes de proceder na escrita é necessário que o arquivo esteja aberto para escrita (arquivos criados automaticamente têm essa propriedade). A descrição do serviço é:

```
Service 0x40            Write To File or Device
-------------------------------------------------------------------
        Entry   AH    = 0x40
                BX    = File handle
                CX    = Number of bytes to write (a zero value truncates
                        the file to the current file position)
                DS:DX = Address of write buffer

        Return  AX = Number of bytes written
                or
                AX = Error code, if CF is set
                     | 0x05 Access denied
                     | 0x06 Invalid handle
```