

## Tratamento de ficheiros em Assembly

### Identificação dos ficheiros pelo sistema operativo

Do ponto de vista do seu armazenamento, os ficheiros podem ser divididos como residentes em disco (ex. ficheiros de texto, ficheiros de programas, bases de dados) ou residentes no hardware (ex. ecran, teclado, portas). O tratamento de ambos os tipos é no entanto feito de modo semelhante.

Embora para o utilizador os ficheiros sejam identificados pelo seu nome (um certo conjunto de caracteres) para a máquina eles têm de ser identificados por um número, designado por handle.

Os handles de dois ficheiros pré-definidos residentes no hardware, que já foram usados nas aulas, são 0=teclado (Standard Input Device) e 1=ecrã (Standard Output Device). Estes valores são reservados e não podem ser atribuídos a outros ficheiros. Para os ficheiros residentes em disco o sistema operativo irá atribuir-lhes um handle variável que será diferente em cada instante e que dependerá do número de ficheiros abertos nesse instante além de outras condições.

### Tipos de ficheiros

Quanto ao seu conteúdo, existem dois grandes tipos de ficheiros: ficheiros de texto e ficheiros de dados. O primeiro tipo corresponde aos ficheiros cujo conteúdo é um conjunto de caracteres directamente legíveis pelo ser humano, os quais são armazenados através dos respectivos códigos ASCII e que podem ser visualizados por qualquer programa de tratamento de texto (ex. Notepad, Word, etc). Os ficheiros de dados normalmente contêm valores binários em forma não directamente legível pelo ser humano, mas que podem ser interpretados por programas que conheçam a estrutura desses dados (ex. ficheiros com programas, com imagens ou ficheiros compactados).

### Utilização dos ficheiros

A utilização de ficheiros é feita em três passos: criação/abertura, processamento e fecho.

Se um ficheiro é inexistente então a primeira operação é a sua criação. Uma vez o ficheiro criado (ou caso ele já existisse), a abertura consiste em atribuir (pelo S.O.) um handle disponível para o ficheiro além de ser indicado o tipo de utilização que ele vai ter e que pode ser leitura (apenas vai ser lido o conteúdo do ficheiro sem que este seja alterado), escrita (o conteúdo do ficheiro vai ser alterado), ou ainda leitura/escrita (em que o conteúdo vai ser lido e também modificado). No passo seguinte, processamento, o ficheiro é processado de acordo com o tipo de abertura (por ex. se o ficheiro foi aberto para leitura, então o seu conteúdo pode ser enviado para o ecran ou para a impressora; se o ficheiro foi aberto para escrita então o seu conteúdo pode ser modificado ou até ser apagado). Após o processamento o ficheiro deve ser fechado para que o sistema operativo liberte o respectivo handle e aguarde o ficheiro de forma segura.

Importa notar que em qualquer um destes passos podem acontecer erros que os programas devem prever e tratar. Por exemplo, a tentativa de abrir um ficheiro inexistente, ou de escrever num ficheiro aberto para leitura, provocará erros que serão assinalados pelo sistema operativo.

**NOTA:** o *handle* de um ficheiro corresponde ao número de aluno dentro de uma escola: os alunos têm nome, ao qual é atribuído um número, que identifica inequivocamente o aluno, pois os nomes podem repetir-se mas os números não.

S.O. = Sistema Operativo

As operações sobre ficheiros são executadas através de chamadas ao **int 21h**:

- **Criação**: função 3Ch (ao ser criado o ficheiro fica logo aberto) → quando o ficheiro é criado o S.O. atribui-lhe um handle, o qual está contido no registo AX
- **Abertura**: função 3Dh → abre um ficheiro dado o seu handle
- **Processamento**: → usa o handle de um ficheiro
  - Leitura: função 3Fh (já usada para ler do teclado)
  - Escrita: função 40h (já usada para escrever no ecran)
- **Fecho**: função 3Eh → fecha um ficheiro dado o seu handle

**NOTA IMPORTANTE:** as funções de criação/abertura exigem que o nome dos ficheiros (a colocar no registo DX) estejam no formato designado por **ASCIIZ**(ASCII Zero). Este não é mais que o formato ASCII normal, mas em que uma string (conjunto de caracteres) deve terminar com o carácter NULL (carácter 0 da tabela ASCII). Em muitas linguagens isto é conhecido por “null terminated strings”

Ex (em FASM):

nome db 'teste.txt' ;nome é uma string ASCII normal

nomez db 'teste.txt',0 ;nomez é uma string em formato ASCIIZ (termina com o carácter 0)

### **Diagrama resumo**

