

---

# Ficheiros

<b>Formação</b>	<b>Jogo do Rato</b>
<b>Local</b>	Marinha Grande
<b>Formador</b>	Sérgio Lopes, knitter.is@gmail.com [mailto:knitter.is@gmail.com]
<b>Ficha</b>	3 - Ficheiros

## Ficheiros

A escrita de ficheiros divide-se em dois tipos: ficheiros binários e ficheiros de texto.

Os ficheiros de texto permitem a escrita formatada de dados, tal como se fazia para o ecrã. Os ficheiros binários guardam os dados em formato binário e só podem ser lidos pela aplicação que os escreveu, ou por uma aplicação que conheça, exactamente a forma como os dados foram gravados.

Para o programador, embora usando funções diferentes, o processo de escrita e leitura de um ficheiro é igual, quer no formato binário quer no formato de texto.

## Modos de Abertura de Ficheiros

<i>r</i>	Abertura para leitura. Não será possível escrever para este ficheiro nem guardar qualquer alteração. O ficheiro tem de existir, caso contrário será emitido um erro.
<i>w</i>	Abertura para escrita. Se o ficheiro existir, qualquer informação que contenha será destruída, caso contrário o ficheiro será criado vazio. Este modo só permite a escrita e não a leitura dos dados no ficheiro.
<i>a</i>	Abertura para actualizar. Permite que seja adicionado conteúdo ao fim de um ficheiro existente. Se o ficheiro não existir será criado completamente vazio. Este modo não permite a leitura do conteúdo do ficheiro.
<i>r+</i>	Abre o ficheiro para leitura e escrita. Se o ficheiro não existir será emitido um erro.
<i>w+</i>	Abre o ficheiro para leitura e escrita. Se o ficheiro não existir será criado completamente vazio.
<i>a+</i>	Abertura para actualizar. Permite a leitura e escrita, se o ficheiro não existir será criado.
<i>b</i>	Este modo não pode ser usado isoladamente, apenas em conjunto com um dos seis modos anteriores. Ao ser aplicado a algum dos modos anteriores, altera a abertura do ficheiro de modo a que o mesmo seja aberto em formato binário, em oposição a uma abertura em modo de texto que é a opção por omissão.

## Tratamento de Erros

As situações onde podem ocorrer erros a abrir ficheiros são: - Nome de ficheiro inválido, que contenha caracteres que não são possíveis de usar ou cujo nome resulte em alguma situação ilegal. - Abrir um ficheiro não fechado, e consequentemente em uso por outro programa. - Abrir um ficheiro que não existe, esta situação é diferente do primeiro caso uma vez que o nome pode estar correcto mas o ficheiro não existir.

Durante a escrita de dados é possível que, devido a condições que tenham mudado após a abertura, surja erros. Entre os erros possíveis alguns dos mais comuns serão: - A quota de espaço do utilizador foi esgotada. - Foi feita uma tentativa para escrever uma quantidade de informação superior à permitida para o processo - A escrita foi interrompida por um sinal do sistema operativo. - Ocorreu um erro de I/O durante a escrita, por exemplo, o utilizador removeu o disco onde o ficheiro estava. - Não há mais espaço livre no disco.

O fecho de um ficheiro pode falhar, principalmente, se ocorrer algum erro na operação de escrita final.

## Funções de Manipulação de Ficheiros

### Tabela 1. Manipulação Geral

<i>fopen</i>	Permite abrir o ficheiro no modo especificado.
<i>fclose</i>	Fecha um ficheiro anteriormente aberto.
<i>fflush</i>	Força a gravação da informação, esvaziando o <i>buffer</i> e gravando os dados no suporte físico do ficheiro.
<i>setbuf</i>	Define um <i>buffer</i> a ser usado pelas operações de leitura e escrita.
<i>sevbuf</i>	Define um <i>buffer</i> a ser usado pelas operações de leitura e escrita.

### Tabela 2. Posição de Cursor

<i>fgetpos</i>	Obtém a posição actual do cursor.
<i>fsetpos</i>	Define a posição actual do cursor.
<i>fseek</i>	Movimenta o cursor dentro do ficheiro. Podem ser usadas algumas constantes como <i>SEEK_SET</i> , <i>SEEK_CUR</i> e <i>SEEK_END</i> .
<i>ftell</i>	Indique a posição inicial para escrita ou leitura.
<i>rewind</i>	Coloca o cursor no início do ficheiro.

### Tabela 3. Escrita e Litura de Dados

<i>fwrite</i>	Escreve um conjunto de informação no ficheiro.
<i>fread</i>	Lê um conjunto de informação do ficheiro.

### Tabela 4. Escrita e Leitura Formatada de Texto

<i>fscanf</i>	Efectua a leitura formatada de dados num ficheiro aberto em modo de texto.
<i>fprintf</i>	Efectua a escrita formatada de dados num ficheiro aberto em modo de texto.

### Tabela 5. Escrita e Leitura de Caracteres

<i>getc</i>	Permite ler um carácter. Esta função é genérica e pode ser usada com ficheiros ou com o teclado.
<i>fgetc</i>	Permite a leitura de um carácter a partir de um ficheiro.
<i>fgets</i>	Permite a leitura de uma <i>String</i> a partir de um ficheiro.

<i>putc</i>	Escreve um carácter para o ficheiro. Tal como a função <i>getc</i> é genérica e pode ser usada com mais coisas que apenas ficheiros.
<i>fputc</i>	Escreve um carácter para um ficheiro.
<i>fputs</i>	Escreve uma string para um ficheiro.
<i>fgetchar</i>	Obtém um carácter de um ficheiro.
<i>fputchar</i>	Escreve um carácter para um ficheiro.

## Tratamento de Erros

*feof*  
Verifica se atingimos o fim do ficheiro.

*clearerr*  
Limpa a variável de controlo de erros.

*ferror*  
Verifica se a última leitura ou escrita provocou um erro.

### Exemplo.

```
#include <stdio.h>
#include <stdlib.h>

#define NOME "palavras.txt"

int main()
{
    FILE *ficheiro;
    char palavra[250], lixo;
    int i = 0;

    ficheiro = fopen(NOME, "w+");

    while(strcmp(palavra, "terminar"))
    {
        i++;
        printf("Escrever Texto: ");
        scanf("%[^\n]", palavra);
        while((lixo = getchar()) != '\n' && lixo != 'EOF');
        fprintf(ficheiro, "Palavra %d: %s\n", i, palavra);
    }

    fclose(ficheiro);

    return 0;
}
```

## Gestão de Centrais

Pegue no projecto entregue, e que contém já todas as funções implementadas, e adicione a gravação e a leitura de ficheiros, permitindo que o nome do ficheiro seja indicado pelo utilizador, e que seja possível escolher o formato de gravação ou leitura (binário ou texto).