

Introdução à leitura/gravação de arquivos em C

Em *Computação*, todo programa e todo conteúdo de programas, devem ser arquivados, o que é feito no denominado **sistema de arquivos**. Do ponto de vista do **sistema operacional**, cada **arquivo** pode na verdade estar registrado em vários espaços não contíguos do **disco**, como ilustrado na figura 1.

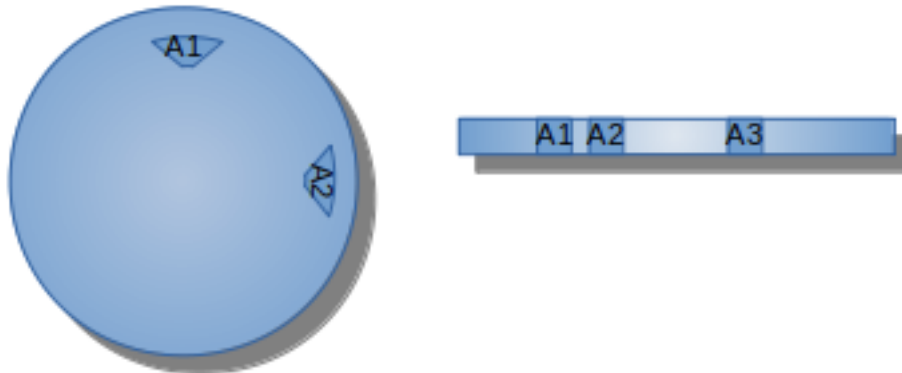


Fig. 1. Ilustração de um programa arquivado em disco rígido ou sólido (SSD).

1. Comandos para leitura de arquivo.

Apesar do arquivo poder estar espalhado pelo *disco rígido*, ele é tratado pelo *sistema operacional* como uma unidade. Assim, cada *linguagem de programação* dispõe de comandos específicos para leitura e gravação de arquivos.

Em **C** é necessário declarar um *apontador para arquivo* (`FILE *apont_arq;`), então associá-lo com o arquivo propriamente dito a partir da função `fopen` (*file open*). Por exemplo, se o arquivo for preparado apenas para leitura, deve-se usar o parâmetro "r" (*read*): `FILE *apont_arq = fopen(arquivo1, "r");`. Ao final, para evitar danificar o arquivo, usar sempre o comando para "fechar" o arquivo: `fclose(apont_arq);`.

Quanto à leitura dos dados (para variáveis internas) ou para impressão, a linguagem **C** adota uma pequena variante dos "tradicionais" comandos para leitura e impressão (`scanf` e `printf`): os comando `fscanf` para leitura e `fprintf` para gravação (*f* indicando *file*).

1.1 Leitura de arquivo com texto em C.

O código 1 ilustra a leitura de arquivo com conteúdo na forma de texto.

```
// Funcao para leitura de arquivo como linhas ("strings")
// Pressupoe declaracao previa da maior linha com: #define MAXCOMPLINHA 100
// Devolve: -1 => arquivo vazio ou inexistente; -2 => arquivo maior que
// capacidade da "string"; -3 => MAXCOMPLINHA insuficiente
int leitura_por_linhas (char *arquivo1, char vet_linhas[][MAXCOMPLINHA]) {
    FILE *apont_arq = fopen(arquivo1, "r"); // Declarar apontador para arquivo
    e associa-lo com o arquivo externo - read only
    char linha1[MAXCOMPLINHA];
```

```

int i = 0;

if (apont_arq==NULL) {
    printf("Arquivo vazio: %s\n", arquivo1);
    return -1;
}

while (1) { // elimina linhas iniciais
    if (fgets(linha1, MAXCOMPLINHA, apont_arq)==NULL || ferror(apont_arq) ||
feof(apont_arq) ) { // (A)
        break; // Final de arquivo, interromper laço
    }
    printf("linha %2d: %s", i, linha1); // em geral a linha vem com '\n',
então não precisa de quebra ao final
    // printf("linha %2d: %s\n", i, linha1); // (A) se MAXCOMPLINHA não
comportar a linha, precisa de quebra
    if (i >= MAXL) {
        printf("Erro: o arquivo de entrada tem mais que %d linhas!\n", MAXL);
        return -3;
    }
    strcpy(vet_linhas[i], linha1); // precisa da biblioteca "string":
#include <string.h>
    i++;
}
fclose(apont_arq); // necessário "fechar" o arquivo (evita erros)
return i;
} // int leitura_por_linhas(char *arquivo1, char
vet_linhas[][MAXCOMPLINHA])

```

Cód. 1. Código ilustrando a leitura de um arquivo com texto em C.

1.2 Leitura de arquivo com dados numéricos em C.

É possível ler conteúdo numérico, por exemplo, o código 2 ilustra a leitura de um arquivo com dados para uma matriz, sendo que em sua primeira linha estão os valores inteiros m e n , respectivamente, o número de linhas e de colunas da matriz, em seguida m linhas, cada uma com n valores "flutuantes".

```

// Função para leitura de arquivo com uma matriz, tendo na primeira linha o
número de linhas e de colunas
// MAXC é a dimensão máxima de colunas da matriz (obrigatório passar essa
dimensão): #define MAXC 100
int leitura_de_numeros (char *arquivo1, float mat[][MAXC], int *nl, int *nc)
{
    FILE *apont_arq = fopen(arquivo1, "r"); // read only
    int conta = 0, result, aux, i, j;
    if (apont_arq==NULL) {
        printf("Arquivo vazio: %s\n", arquivo1);
        fclose(apont_arq);
        return -1;
    }
    fscanf(apont_arq, "%d %d", nl, nc); // primeira linha deve ter 2 inteiros
    for (i=0; i<*nl; i++)
        for (j=0; j<*nc; j++) {
            result = fscanf(apont_arq, "%f", &mat[i][j]);

```

```
        conta++;
    }
    fclose(apont_arq);
    return conta;
}
```

Cód. 2. Código ilustrando a leitura de um arquivo texto com uma matriz de "flutuantes" em C. Assim, o programa que precisar fazer a leitura da matriz em um arquivo, por exemplo, de nome "*dados1.txt*", bastaria invocar a função com:

```
resp = leitura_de_numeros("dados1.txt", matriz, &m, &n);
```

ou seja, precisa dispor de uma matriz de "flutuantes" (como `matriz[100][100]`) e usar as variáveis `m` e `n`, respectivamente para total de linhas e de colunas. Note que para a função alterar os valores de `m` e `n` é necessário passá-las por **referência** (ou por **endereço**), daí a necessidade do operador `&` (que devolve o *endereço* ocupado por uma variável).

[Leônidas de Oliveira Brandão](http://line.ime.usp.br)
<http://line.ime.usp.br>

Alterações 