Aula 14:
- Estruturas e arquivos

Prof. João Henrique Kleinschmidt
Material elaborado pelo prof. Jesús P. Mena-Chalco

3Q-2018

Arquivos

Arquivo é uma forma de manter os dados de um programa salvos, mesmo após seu encerramento. Também é útil para carregar dados externos em determinados programas. Em linguagem C, arquivos são manipulados por meio de um ponteiro para FILE.

Para abrir arquivos, usa-se a função *fopen(char *nome do arquivo, char modo),* em que *nome do arquivo* é o nome do arquivo que se quer abrir, e *modo* é a forma de abertura que se quer realizar (conforme Tabela abaixo). Para fechar um arquivo, usa-se a função *fclose(FILE *fp),* em que *FILE *fp* é o ponteiro para o arquivo.

Modo	Descrição	Tipo de arquivo
"r"	Abre arquivo somente para leitura	Texto
"w"	Abre arquivo novo para escrita (caso já exista, é sobrescrito)	Texto
"a"	Abre arquivo para escrita a partir da última linha	Texto
"rb"	Abre arquivo somente para leitura	Binário
"wb"	Abre arquivo novo para escrita (caso já exista, é sobrescrito)	Binário
"ab"	Abre arquivo para escrita a partir da última linha	Binário

Arquivos

Existem diversas funções em C para escrita e leitura de dados em arquivos.

Para escrever dados em arquivos de texto pode-se utilizar, por exemplo, as funções *fprintf(), fputs(), putc()* e *fputc().* Para ler dados de arquivos de texto, por exemplo, pode-se utilizar a função *fscanf().*

Os próximos exemplos ilustram o uso de algumas destas funções.

Arquivos - Escrita de dados em arquivo de texto

```c
#include <stdio.h>
int main(){
    FILE *parq;                          // declaração de ponteiro para FILE
    parq = fopen("dados.txt", "w"); // abertura de arquivo no modo escrita
    if (parq == NULL)                    // caso não seja possível abrir arquivo
       return 1;
    fputs("ola mundo usando fputs\n", parq);      // escrita de texto no arquivo
    fprintf(parq, "ola mundo usando fprintf\n"); // escrita de texto no arquivo
    fputc('A', parq);                                  // escrita de caractere no arquivo
    fclose(parq);                        // fechamento de arquivo
    return(0);
}
```

Arquivos – Leitura de dados a partir de arquivo de texto

```c
#include <stdio.h>
int main(){
    FILE *parq;                          // declaração de ponteiro para FILE
    parq = fopen("dados.txt", "r"); // abertura de arquivo no modo leitura
    if (parq == NULL)                    // caso não seja possível abrir arquivo
       return 1;
    char c;
    // enquanto houver algo a ser lido, leia
    while(fscanf(parq, "%c", &c) == 1)
        printf("%c", c);                 // imprima o caractere lido
    printf("\n");
    fclose(parq);                        // fechamento de arquivo
    return(0);
}
```

Arquivos e structs

Também é possível ler e gravar estruturas (*structs*) inteiras em arquivos, de uma só vez. Para isso, pode-se utilizar as funções *fread()* e *fwrite(),* para ler e escrever, respectivamente.

No próximo exemplo, as linhas 21-22 gravam no arquivo “\coordenadas.dat" os pontos 1 e 2 declarados nas linhas 3-13. O **primeiro argumento** de *fwrite()* é um buffer com o conteúdo a ser gravado (o endereço do conteúdo). O **segundo** é o tamanho do tipo de dado a ser gravado, no caso, o tamanho da estrutura coordenada cartesiana. O **terceiro** argumento representa a quantidade de estruturas que serão gravadas na chamada da função e o **último** é o ponteiro para o arquivo.

Os mesmos argumentos são utilizados para a função *fread*, com o diferencial de que o primeiro argumento é o endereço da variável em que serão armazenados os *structs* lidos.

Arquivos – Escrita em arquivo de dados do tipo struct

```c
#include <stdio.h>
int main(){
    struct coordenada_cartesiana {
        float x;
        float y;
    };
    struct coordenada_cartesiana ponto1;
    ponto1.x = 13.4;
    ponto1.y = 1.6;

    struct coordenada_cartesiana ponto2;
    ponto2.x = 2.3;
    ponto2.y = 1.1;

    FILE *parq;                                  // declaração de ponteiro para FILE
    parq = fopen("coordenadas.dat", "wb"); // abertura de arquivo no modo escrita
    if (parq == NULL)                            // caso não seja possível abrir arquivo
        return 1;

    // fwrite(buffer, tamanho, quantidade, arquivo)
    fwrite(&ponto1, sizeof(struct coordenada_cartesiana), 1, parq);
    fwrite(&ponto2, sizeof(struct coordenada_cartesiana), 1, parq);
 return 0;
}
```

Arquivos – Leitura de arquivos contendo dados do tipo struct

```c
#include <stdio.h>
int main(){
    struct coordenada_cartesiana {
        float x;
        float y;
    };
    // declaração de vetor de coordenada_cartesiana
    struct coordenada_cartesiana pontos[2];

    FILE *parq;                                  // declaração de ponteiro para FILE
    parq = fopen("coordenadas.dat", "rb"); // abertura de arquivo no modo escrita
    if (parq == NULL)                            // caso não seja possível abrir arquivo
        return 1;

    // fread(buffer, tamanho, quantidade, arquivo)
    fread(pontos, sizeof(struct coordenada_cartesiana), 2, parq);

    for (int i = 0; i < 2; i++)
        printf("(%.1f,%.1f)\n", (*(pontos + i)).x, (*(pontos + i)).y);
 return 0;
}
```

Exercício

Defina uma estrutura para armazenar informações de hora, minuto e segundo e salve em um arquivo.

Leia o arquivo anterior.