

Arquivos

Roberto Rocha

Armazenar dados em disco

Unidade 4 – Desenvolvimento

4.4 Noções sobre armazenamento de dados

:

4.4.2 – Em memória secundária (arquivos)

:

Arquivos



Arquivos são importantes em computação.

São a forma de se perpetuar dados em dispositivos de armazenamento,

Toda a organização dos computadores se baseia no conceito de arquivos, inclusive as metáforas de uso de dados e de sua hierarquia de armazenamento (pastas, diretórios).

É fundamental para um programador saber registrar dados no formato de arquivos, bem como conhecer as operações de manipulação destes.

Um **arquivo** é, também, fonte de dados para o programa, podendo servir como **entrada de dados** (input); ou é, também, o destino dos dados gerados pelo programa, ou sua **saída de dados** (output).

Arquivos

Ao contrario de outras linguagens, em que os arquivos são vistos como possuidores de uma estrutura interna ou registro associado, um arquivo, em C, é apenas um conjunto de bytes colocados uns após os outros de forma sequencial.

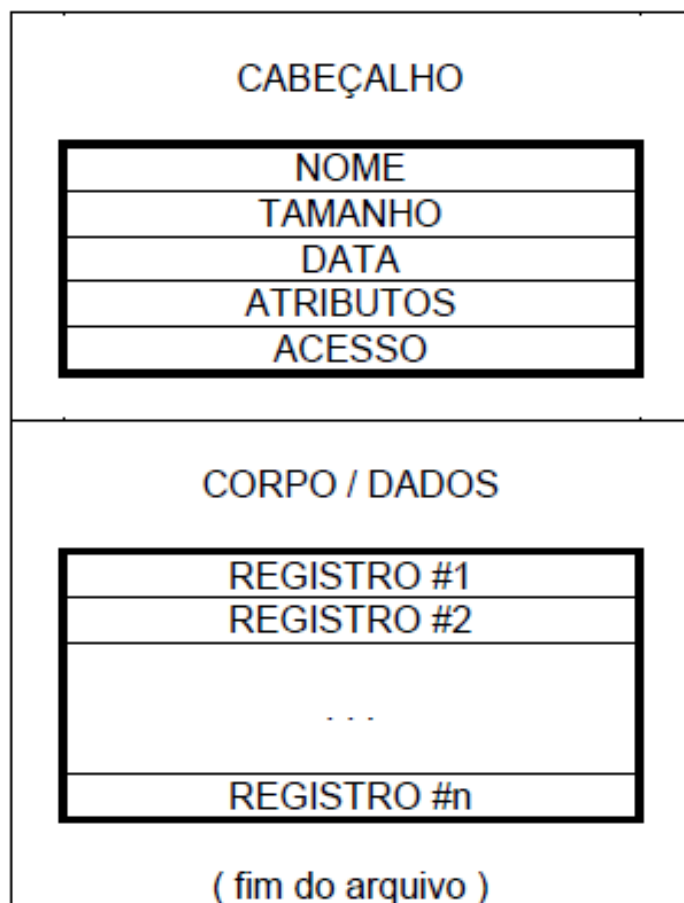
Independentemente do periférico usado para a entrada ou saída de dados, a Linguagem C processa os bytes por meio de streams (conjunto sequencial de caracteres, ou de bytes, sem qualquer estrutura interna).

As streams, portanto, independem do dispositivo utilizado e representam um conceito universal de fluxo de dados.

Cada stream está ligada a um arquivo, que pode não corresponder fisicamente a uma estrutura existente no disco, como é o caso do teclado ou da tela do computador.

Arquivos

Pode-se chamar de **arquivo** um conjunto organizado de registros, armazenado em algum dispositivo de memória secundária. Um arquivo terá a seguinte organização para efeitos práticos:



- Cabeçalho - é um registro padrão contendo:
 - Nome - nome externo do arquivo (no diretório ou índice)
 - Tamanho - tamanho de arquivo em número de registros
 - Data - data de criação ou de última alteração
 - Atributos - informações sobre segurança, visibilidade, disponibilidade para leitura etc.
 - Acesso - forma de organização:

Arquivos - Acesso

Sequencial

Os registros estão dispostos fisicamente um após o outro, segundo a ordem de gravação, e para se acessar um determinado registro, deve-se passar, obrigatoriamente, por todos os anteriores a ele. A cada operação coloca-se o próximo registro à disposição.

Direto

O acesso aos registros pode ser feito fora da ordem, por meio da indicação de um índice. O índice natural de um registro é o seu número de ordem.

Arquivos sequenciais em C

Abrindo arquivos

Para se abrir um arquivo, declaramos uma variável do tipo **FILE** (essa é, na verdade, um ponteiro para **FILE** – que está definido em **stdio.h** e não é do tipo primitivo):

```
FILE *fp; // ponteiro para FILE
```

e então utilizamos a função **fopen()**:

```
FILE * fopen (const char * filename, const char * mode);
```

```
/* nome do arquivo e modo de abertura */
```

Modos de abertura

“r” – read (leitura - se erro na abertura - retorna NULL);

“w” – write (escrita - cria novo arquivo; se não puder criar, retorna NULL);

“a” – append (inclui novos registros - se não existir ele o cria).

Arquivos sequenciais em C

Fechando arquivos

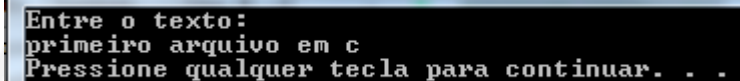
A sintaxe dessa função é:
`int fclose(FILE * arq);`

Nessa sintaxe, arq é o ponteiro para o arquivo aberto. Se o fechamento do arquivo ocorrer **sem problemas**, será retornado o **valor zero**. Qualquer outro valor indica erro de fechamento.

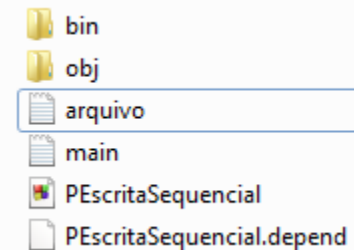
Arquivos sequenciais em C

Escrevendo no arquivo txt e apagando o que existia

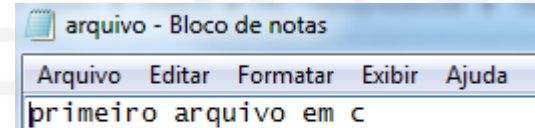
```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    FILE * arquivo;
    char mensagem[80];
    printf("Entre o texto: \n");
    gets(mensagem);
    if ((arquivo = fopen("arquivo.txt","w")) == NULL) // modo de escrita ativo
    {
        printf("Erro de abertura! \n");
    }
    else
    {
        fprintf(arquivo, "%s \n", mensagem);
        fclose(arquivo);
    }
    system("pause");
    return 0;
}
```



Entre o texto:
primeiro arquivo em c
Pressione qualquer tecla para continuar. . .



- bin
- obj
- arquivo
- main
- PEscritaSequencial
- PEscritaSequencial.depend



arquivo - Bloco de notas

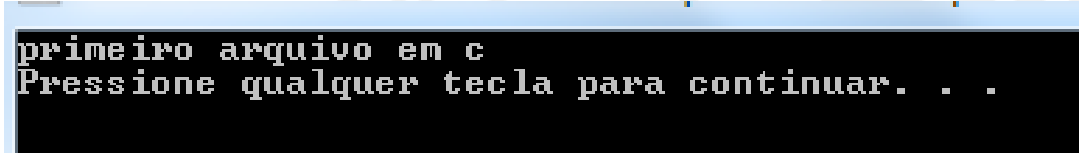
Arquivo	Editar	Formatar	Exibir	Ajuda
---------	--------	----------	--------	-------

primeiro arquivo em c

Arquivos sequenciais em C

Lendo o arquivo txt

```
#include <stdio.h>
#include <stdlib.h>
int main()
{char buffer[128];
  FILE *arquivo;
  if ((arquivo = fopen("arquivo.txt","r")) == NULL)
  { printf("Erro de abertura! \n");
  }
  else
  { fgets(buffer, 80, arquivo);
    while (!feof(arquivo))
    { printf("%s", buffer);
      fgets(buffer, 80, arquivo);
    }
    fclose(arquivo);
  }
  system("pause");
  return 0;
}
```

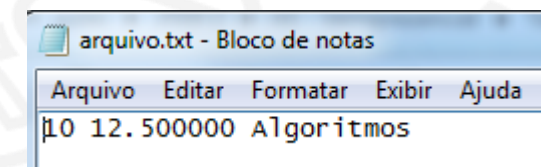
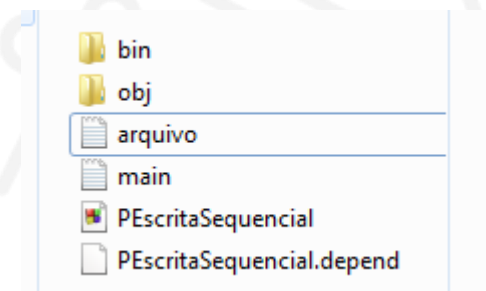


```
primeiro arquivo em c
Pressione qualquer tecla para continuar. . .
```

Arquivos sequenciais em C

Escrevendo no arquivo txt e apagando o que existia

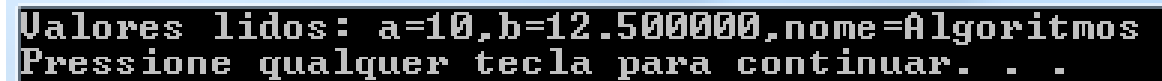
```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    FILE * arquivo;
    int a=10;
    float b=12.5;
    char nome[]="Algoritmos";
    if ((arquivo = fopen("arquivo.txt","w")) == NULL) // modo de escrita ativo
    {
        printf("Erro de abertura! \n");
    }
    else
    {
        fprintf(arquivo, "%i %f %s ",a,b,nome);
        fclose(arquivo);
    }
    system("pause");
    return 0;
}
```



Arquivos sequenciais em C

Lendo o arquivo txt

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int a;
    float b;
    char nome[20];
    FILE *arquivo;
    if ((arquivo = fopen("arquivo.txt","r")) == NULL)
    { printf("Erro de abertura! \n");
    }
    else
    { fscanf(arquivo,"%d",&a);
      fscanf(arquivo,"%f",&b);
      fscanf(arquivo,"%s",nome);
      printf("Valores lidos: a=%d,b=%f,nome=%s\n",a,b,nome);
      fclose(arquivo);
    }
    system("pause");
    return 0;
}
```

A screenshot of a terminal window with a black background and white text. The text displays the output of the C program: 'Valores lidos: a=10,b=12.500000,nome=Algoritmos' followed by 'Pressione qualquer tecla para continuar. . .' on the next line.

Valores lidos: a=10,b=12.500000,nome=Algoritmos
Pressione qualquer tecla para continuar. . .

Arquivos sequenciais em C

Exercícios

Fazer um programa com um menu para:

- a) - gravar em um arquivo os 64 primeiros valores da série: 1 1 2 3 5 8 13 21 34 . . .
- b) - ler o arquivo letra a e armazenar os valores pares em um arquivo e os impares em outro arquivo
- c) - ler o arquivo letra a e armazenar apenas os primos em um arquivo
- d) - leia um conjunto de valores (flag -1) e verifique para cada valor se ele se encontra ou não no arquivo gravado na letra a. Caso não exista deverá mostrar o valor anterior e o posterior a esse valor. Exemplo: caso seja lido o valor 8 – deverá ser mostrada a informação o valor 8 está no arquivo na posição 6, caso o valor lido seja o 17 – deverá ser mostrada a informação o valor 17 não está no arquivo, valores mais próximos: 13 e 21.



PUC Minas
Virtual