Arquivo em C

Parte 1: Arquivos modo texto

Naur jr. IFSP-HTO PLP2- 2sem 2014

Conteúdo

- Revisão
  - Etapas:
    - Abertura: fopen()
    - Manipulação: leitura ou escrita...
    - Fechamento: fclose()
- Caractere por vez: escrita: fputc() e leitura: getc()
- Melhoria na abertura de arquivo: evitando erro.
- Linha a linha: escrita –fputs() e leitura – fgets()
- **Dados formatado: escrita - fprintf() e leitura - fscanf()**
- **Copiando um arquivo para outro.**

ETAPAS: Abertura, Manipulação e Fechamento

- 1 ª Abertura de Arquivo:

fopen(Arquivo, MODO): função que abre o arquivo.

FILE **fopen**(const char *nome_arquivo, const char *modo_abertura);

Ex:

```
fptr  =  fopen("caracter.txt","w");     //  Abertura do arquivo texto para escrita
fptr  =  fopen("caracter.txt",“r");     //  Abertura do arquivo texto para leitura
```

“**w**” write: cria o arquivo MODO TEXTO para ESCRITA, e se existir elimina o conteúdo.

“**r**” read: abre o arquivo MODO TEXTO para LEITURA, e se existir a operação falha NULL.

ETAPAS: Abertura, Manipulação e Fechamento

- 2 ª Manipulação de Arquivo (leitura/**escrita**):
- Escrita:

fputc (CARACTER, ARQUIVO):

- Escreve no arquivo o caracter.
- Recebe dois argumentos: o caracter a ser gravado e o ponteiro para a estrutura FILE do arquivo.
- Retorna o caracter gravado ou EOF se aconteceu algum erro.

int **fputc**(int caracter , FILE *ponteiro_arquivo);

Ex: int **fputc**(ch , fptr);

ETAPAS: Abertura, Manipulação e Fechamento

- 2 ª Manipulação de Arquivo (**leitura**/escrita):
- Leitura:

fgetc (ARQUIVO):

- Lê um caractere do arquivo.
- Recebe como argumento o ponteiro para a estrutura FILE do arquivo.
- Retorna o caracter lido ou EOF se encontrar o fim do arquivo.

int **fgetc**(FILE *ponteiro_arquivo);

Ex: ch = **fgetc**(fptr)

Lembre: Quando o final do arquivo é alcançado a função devolve EOF.

ETAPAS: Abertura, Manipulação e Fechamento

- 3 ª Fechamento:

fclose (ARQUIVO):

- Fecha o arquivo e esvazia o conteúdo do buffer, garantindo que nenhuma informação seja deixado no buffer, também chamado de descarga ou flushing
- Libera as áreas de comunicação entre o programa e sistema operacional.

int **fclose**(FILE *ponteiro_arquivo);

Ex: **fclose**(fptr)

Escrita Arquivo: caracter por caracter

//Exemplo que lê caracteres do teclado e escreve os caracteres no arquivo **f**putc( )

```
1.    #include <stdio.h>
2.    #include <conio.h>

3.    int main()
4.    {
5.        FILE *fptr;
6.        char ch;

7.        //abre arquivo para escrita
8.        fptr  =  fopen("caracter.txt","w");     // 1ª Abertura do arquivo texto para escrita w
9.
10.       while( ( ch = getche( ) ) != '\r')
11.       {
12.               fputc(ch , fptr);               // 2ª  manipulação (escrita) do arquivo
13.       }
14.
15.       fclose( fptr );                         // 3ª fechamento do arquivo

16.       return 0;

17.   }
```

Leitura Arquivo: caracter por caracter

//Exemplo que le caracteres do arquivo e escreve os caracteres no TELA **f**getc( )

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

2.    int main()
3.    {
4.        FILE *fptr;
5.        short int ch;

6.        //abre arquivo para escrita
7.        fptr  =  fopen("caracter.txt", "r");        // 1ª Abertura do arquivo
8.
9.        while( (ch = fgetc(fptr) ) != EOF )         // 2ª  manipulação (leitura) do arquivo
10.       {
11.           printf("%c", ch);
12.       }
13.
14.       fclose( fptr );                             // 3ª fechamento do arquivo

15.       return 0;

}
```

Evitando erro na abertura do arquivo

```
.....
int main()
{
        FILE *fptr;
        char ch;

        //abre arquivo para escrita
        fptr  =  fopen("caracter.txt", "r");     // ?????? Se der erro? Quais os erros que pode acontecer?

        while( ( ch = getche()) != '\r' )
        {
                fputc(ch , fptr);          // 2ª  manipulação (escrita) do arquivo
        }

        fclose( fptr );                                  // 3ª fechamento do arquivo

        return 0;

}
```

Evitando erro na abertura do arquivo

```
#include <stdlib.h>  //exit()
int main()
{
        FILE *fptr;
        char ch;

        //abre arquivo para escrita
        if ( (fptr  =  fopen("caracter.txt","w") ) == NULL )      // abre quando de erro retorna NULL
        {
                printf("ERRO na abertura do arquivo");
                exit(1);   // return (1);

        }

        while( (ch = getche()) != '\r')
        {
                fputc(ch , fptr);            // 2ª  manipulação (escrita) do arquivo
        }

        fclose( fptr );                                  // 3ª fechamento do arquivo

        return 0;

}
```

Escrita Arquivo: escreve linha inteira fputs(STRING,ARQUIVO)

//Escreve uma linha inteira lida com gets no arquivo

```
1.  #include <stdio.h>
2.  #include <stdlib.h>
3.  #include <conio.h>

4.  int main( )
5.  {
6.      FILE *fptr;
7.      char texto[81];          // vetor de caracteres = string

8.      //abre arquivo para escrita
9.      if ( (fptr  =  fopen("linhaTexto.txt","w") ) ==  NULL )      // 1ª Abertura do arquivo
10.     {
11.         printf("ERRO na abertura do arquivo");
12.         exit(1);   // return 1;
13.     }
14.
15.     gets(texto);
16.     while( !feof(stdin))   // ctrl-z
17.     {
18.         fputs(texto , fptr);              // fputs(STRING,ARQUIVO)
19.         gets(texto);
20.     }
21.
22.     fclose(fptr);
23.     return 0;
24. }
```

Leitura Arquivo: leitura de uma linha com fgets(STRING,TAMANHO,ARQUIVO)

```
1.  //Escreve linha a linha no arquivo

2.  #include <stdio.h>
3.  #include <stdlib.h>
4.  #include <conio.h>

5.  int main()
6.  {
7.      FILE *fptr;
8.      char texto[81];    // vetor de caracteres = string

9.      //abre arquivo para escrita
10.     if ( (fptr  =  fopen("linhaTexto.txt","r") ) == NULL )      // 1ª Abertura do arquivo
11.     {
12.             printf("ERRO na abertura do arquivo");
13.                     exit(1);   // return  1;
14.     }
15.
16.     while( fgets( texto, 80, fptr )  !=  NULL)   // ctrl-z      fgets(STRING,TAMANHO,ARQUIVO)
17.     {
18.             printf("%s",texto);
19.     }
20.
21.     fclose(fptr);

22.     return 0;

23. }
```

Escrita Arquivo: dados formatados fprintf()

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

2.    int main( void )
3.    {
4.       unsigned int conta; // numero da conta
5.       char   nome[ 30 ]; //  nome da conta
6.       double saldo; // saldo da conta

7.       FILE *cfPtr; // cfPtr = clients.dat file pointer

8.       // fopen abre arquivo. Se nao conseguir criar arquivo sai do programa
9.       if ( ( cfPtr = fopen( "clients.dat", "w" ) ) == NULL ) {
10.         puts( "File could not be opened" );
11.      } // fim if
12.      else
13.      {
14.         puts( "Entre com conta, nome e saldo." );
15.         puts( "Entre com EOF para finalizar entrada." );
16.         printf( "%s", "? " );
17.         scanf( "%d%29s%lf", &conta, nome, &saldo );

18.         // escreve conta, nome e saldo no arquivo com fprintf
19.         while ( !feof( stdin ) )
20.         {
21.            fprintf( cfPtr, "%d %s %.2f\n", conta, nome, saldo );
22.            printf( "%s", "? " );
23.            scanf( "%d%29s%lf", &conta, nome, &saldo );
24.         } // fim while
25.
26.         fclose( cfPtr ); // fclose closes file

27.      } // fim else

28.   } // fim main
```

Leitura de Arquivo: dados formatados scanf()

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

2.    int main( void )
3.    {
4.       unsigned int conta; //  numero da conta
5.       char nome[ 30 ]; //    nome da conta
6.       double saldo; //       saldo da conta

7.       FILE *cfPtr; // cfPtr = clients.dat ponteiro para arquivo

8.       // fopen  abre arquivo, finaliza o programa se arquivo nao aberto
9.       if ( ( cfPtr = fopen( "clients.dat", "r" ) ) == NULL ) {
10.         puts( "Arquivo nao pode ser aberto" );
11.      } // end if
12.      else
13.      {    // le conta nome e saldo do arquivo com fscanf e mostra na tela com printf
14.           printf( "%-10s%-13s%s\n", "Conta", "Nome", "Saldo" );
15.           fscanf( cfPtr, "%d%29s%lf", &conta, nome, &saldo );

16.           // while nao eh fim de arquivo
17.           while ( !feof( cfPtr ) )
18.           {
19.              printf( "%-10d%-13s%7.2f\n", conta, nome, saldo );
20.              fscanf( cfPtr, "%d%29s%lf", &conta, nome, &saldo );
21.           } // fim while

22.           fclose( cfPtr ); // fclose fecha o arquivo
23.      } // fim else
24.   } // fim main
```

Este campos devem ser do mesmo tamanho e sequencia de quando escrever

Manipulando entrada por linha de comando: int main(int argc, char *argv[ ])

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

3.  int main( int argc, char *argv[])
4.  {
5.      int i;
6.
7.      if ( argc  <  3 )
8.      {
9.          printf("faltando argumentos");
10.         return 0;
11.     }
12.
13.     printf("\n\n");

14.     for( i = 0 ; i < argc ;  i++)
15.         printf("O argumentos %d:   %s \n", i, argv[ i ] );
16.
17.     return 0;
```

```
char   *argv [ ]  -> vetor de strings
ind i   0     1     2      3    n
argv [“t   ,  “t  ,  “ t   ,  “ t    ....  ]
        e  ,   e,     e,      e
        x  ,   x,     x,      x
        t  ,   t,     t,      t
        o  ,   o,     o,      o
        1  ,   2,     3,      4
       \0”,  \0” ,  \0” ,   \0”

char   *argv   -> apenas uma string
ind i    0    1     2    3   4     5
argv [“ t     e     x    t    o    \0 “  ] ou
argv [  ‘t’ , ‘ e ‘,  ‘ x’, ‘ t’,  ‘o’,  ‘ \0’  ]
```

```
/* programa que copia arquivo caractere por caractere em um arquivo copia */
1.  #include <stdio.h>
2.  #include <stdlib.h>  // exit()

3.  int main( int argc, char *argv [ ] )
4.  {
5.          FILE *original,*copia;
6.          char caracter;

7.          if(argc < 3) {
8.                  printf("\nSintaxe correta:\n\n");
9.                  printf("copiar ARQUIVO_ORIGEM   ARQUIVO_DESTINO\n\n");
10.                 exit(1); // return 1
11.          }

12.         if ( strcmp( argv[1] , argv[2] )  == 0 )        // verifica se o nomes são iguais
13.         {
14.                 printf("\nO nome do arquivo original não pode ser igual ao da copia.\n\n");
15.                 exit(1);
16.          }
17.
18.         if (  (original = fopen(argv[1],"r")) == NULL) {          // abre o arquivo_origem para leitura
19.                 printf("\nErro ao abrir o arquivo original.\n\n");
20.                 exit(1);
21.         }
22.
23.         if ( (copia = fopen(argv[2],"w")) == NULL) {              // abre o arquivo_destino para escrita
24.                 printf("\nErro ao abrir o arquivo cópia.\n\n");
25.                  exit(1);
26.         }
```

```
26.         while( !feof(original) ) {
27.               caracter = getc(original);
28.               putc(caracter,copia);
29.         }

30.         fclose(original);
31.         fclose(copia);

32.         printf("\n%s copiado com sucesso com o nome de   %s. \n\n”  ,argv[1],  argv[2]);

33.         return(0);

    } // fim main
```

Exercício

- Alterar o programa que copia arquivo, copiando linha ao invés de caractere por caractere.
  - Dicas:

(1) alterar as funções que copia caracter por linha

```
26.     while( !feof(original) ) {
27.         caracter = getc(original);
28.         putc(caracter,copia);
29.     }
```

(2) trocar a declaração da variável de caractere para vetor de caractere

```
6.      char caracter;
```

```
26.         while( !feof(original) ) {
27.             fgets(linha,80,original);     // caracter = getc(original);
28.             fputs(linha,copia);           // putc(caracter,copia);
29.         }

30.         fclose(original);
31.         fclose(copia);

32.         printf("\n%s copiado com sucesso com o nome de   %s. \n\n"  ,argv[1],
            argv[2]);

33.         return(0);

}
```

Linha 6. char caracter;
char linha[81];

Referências

- Video Aulas (YouTube)
- Programar em C - Manipulação de Arquivos txt em C / Ler Dados - https://www.youtube.com/watch?v=y_euDUgoND8
- **Programar em C - Manipulação de Arquivos txt em C / Incluir Dados - Aula 84**

https://www.youtube.com/watch?annotation_id=annotation_238568&feature=iv&src_vid=y_euDUgoND8&v=USsUSMpNGsM

Sites sobre manipulação de Arquivos em C

http://homepages.dcc.ufmg.br/~joaoreis/Site%20de%20tutoriais/c_int/arquivos.htm

http://pt.wikibooks.org/wiki/Programar_em_C/Entrada_e_sa%C3%ADda_em_arquivos

http://www.vivaolinux.com.br/artigo/Manipulando-arquivos-em-C-%28parte-1%29/?pagina=4

http://homepages.dcc.ufmg.br/~joaoreis/Site%20de%20tutoriais/c_int/arquivos.htm

http://pt.wikibooks.org/wiki/Programar_em_C/Entrada_e_sa%C3%ADda_em_arquivos

http://www.ime.usp.br/~elo/IntroducaoComputacao/Manipulacao%20de%20arquivo.htm

Próximas aulas

- Arquivos Binário.
  - Conta Corrente: (rever em vetor ==)
  - Fazer cada funcionalidade passo a passo. (revisão)

- Estoque (trabalho)
- Agenda (juntos)
- Trabalho = Prova 1 mas em arquivo
- Prova = comando em SQL -> fazer em C. Ex: Select * from aluno;