



INSTITUTO FEDERAL
Sul-rio-grandense

Câmpus
Charqueadas

EDUCAÇÃO
PÚBLICA
100%
GRATUITA

Manipulação de Arquivos

Programação Estruturada

Prof. André del Mestre

Abrir e fechar arquivos

Arquivos

Preliminares

- Utiliza-se funções nativa da linguagem C para manipular arquivos
 - Biblioteca **stdio.h**
 - As funcionalidades são limitadas
 - Abrir, fechar, ler, escrever
 - Todas as funções trabalham com o conceito **ponteiro de arquivo**
 - Novo tipo de dado para arquivos: **FILE**
 - Todas as letras maiúsculas
 - Exemplo, **fp** eh o ponteiro de arquivos que permitira manipular arquivos

```
FILE *fp;
```

Arquivos

Abrindo um arquivo

- Para abrir arquivos, use a função `fopen()`

```
FILE *fopen(char *nome_arquivo, char *permissoes);
```

- A string `nome_arquivo` deve conter todo o caminho de diretórios do arquivo a ser aberto
 - Caminho absoluto: `"C:/Projeto/docs/dados.txt"`
 - Caminho relativo: `". /docs/dados.txt"`
 - Prefira o caminho **relativo**

Arquivos

Abrindo um arquivo

- Para abrir arquivos, use a função `fopen()`

```
FILE *fopen(char *nome_arquivo, char *permissao);
```

- A string `permissao` determina o tipo de uso do arquivo e tem as seguintes opções

Permissão	Descrição	Exigência
"r"	Abre arquivo em modo de leitura.	Arquivo E diretório devem existir
"w"	Abre OU cria arquivo em modo de escrita.	Diretório deve existir
"a"	Abre OU cria arquivo em modo de escrita ADICIONAL.	Diretório deve existir
"r+"	Abre arquivo em modo de leitura E escrita.	Arquivo E diretório devem existir
"w+"	Abre OU cria arquivo em modo de leitura E escrita.	Diretório deve existir
"a+"	Abre OU cria arquivo em modo de leitura E escrita ADICIONAL.	Diretório deve existir

Arquivos

Abrindo um arquivo

- Para abrir arquivos, use a função `fopen()`

```
FILE *fopen(char *nome_arquivo, char *permissao);
```

- A string `permissao` determina o tipo de uso do arquivo e tem as seguintes opções

Permissão	Descrição	Exigência
"rb"	Adicionar a letra b na permissão do arquivo indica a manipulação de um arquivo binário. Arquivos binários NAO serão cobrados nesta disciplina. Somente arquivos texto.	As mesmas do arquivo texto
"wb"		
"ab"		
"rb+"	A descrição das permissões são as mesmas do arquivo texto	
"wb+"		
"ab+"		

Arquivos

Abrindo um arquivo

- Caso haja falha ao abrir o arquivo, tipicamente a função não pode continuar a executar
 - Nesse caso o ponteiro apontará para NULL
 - Como boa pratica, o programador pode tratar essa exceção
 - Exemplo

```
int main() {  
    FILE *fp;  
    fp = fopen("./dados.txt", "r");  
    if(fp==NULL) {  
        printf("Erro ao abrir o arquivo\n");  
        return 0;  
    }  
    ...  
}
```

Arquivos

Fechando um arquivo

- Para fechar arquivos, use a função `fclose()`

```
int fopen(FILE *fp);
```

- Sempre após o uso do arquivo, deve-se fecha-lo.

- Libera memoria
- Evita arquivos corrompidos

```
int main() {  
    FILE *fp;  
    fp = fopen("./dados.txt", "r");  
    if (fp==NULL) {  
        printf("Erro ao abrir o arquivo\n");  
        return 0;  
    }  
    //FACA ALGO  
    fclose(fp);  
}
```


Leitura e Escrita em Arquivos

Leitura e Escrita em Arquivos

Funções

- Uma vez aberto, é possível ler e escrever no arquivo
- Abaixo a lista de funções para ler e escrever arquivos
- **Recomendação:**
 - Use-as sempre aos pares. Exemplo:
 - Se leu um arquivo com `fgets()`, escreva no mesmo arquivos com `fputs()`

Leitura	Escrita	Descrição
<code>fgetc()</code>	<code>fputc()</code>	Le/Escreve um caracter por vez. Não quebra linha
<code>fgets()</code>	<code>fputs()</code>	Le/escreve uma string por vez. Cuidado com quebras de linha
<code>fscanf()</code>	<code>fprintf()</code>	Quebra de linha automática. Indicado para leitura de float e int
<code>fread()</code>	<code>fwrite()</code>	Utilizado em arquivos binários.

Leitura e Escrita em Arquivos

Caracter - fgetc e fputc

- Para escrever um caracter em arquivos, use a função fputc()

```
int *fputc(char ch, FILE *fp);
```

- No exemplo, a leitura termina quando a marcação EOF eh encontrada
 - EOF eh abreviatura de “End of File” e significa fim de arquivo
 - Para digitar EOF, tente Ctrl+Z

```
int main() {  
    FILE *fp;  char ch;  
    fp = fopen("./dados.txt", "w");  
    if(fp==NULL){ return 0; }  
  
    while( (ch=getchar()) != EOF)  
        fputc(ch, fp);    //nao usa o retorno int para nada  
  
    fclose(fp);  
}
```

Leitura e Escrita em Arquivos

Caracter - fgetc e fputc

- Para ler um caracter em arquivos, use a função fgetc()

```
char *fgetc(FILE *fp);
```

- No exemplo, a leitura termina quando a marcação EOF eh encontrada
 - EOF eh abreviatura de “End of File” e significa fim de arquivo
 - Para digitar EOF, tente Ctrl+Z

```
int main() {  
    FILE *fp; char ch;  
    fp = fopen("./dados.txt", "r");  
    if(fp==NULL){ return 0; }  
  
    while( (ch=fgetc(fp)) != EOF)  
        printf("%c",ch);  
  
    fclose(fp);  
}
```

Leitura e Escrita em Arquivos

Strings - fgets e fputs

- Para escrever uma string em arquivos, use a função fputs()

```
int fputs(char *str, FILE *fp);
```

- **Cuidado!** Você deve colocar uma quebra de linha na string a ser escrita!
 - Caso contrario, a string será colocada na mesma linha de outra string

```
int main() {  
    FILE *fp; char str[30]="Demo fputs";  
    fp = fopen("./dados.txt", "w");  
    if(fp==NULL){ return 0; }  
  
    printf("%i", fputs(str, fp));  
    printf("%i", fputs("Nao quebrou linha antes!\n", fp));  
  
    fclose(fp);  
}
```

Leitura e Escrita em Arquivos

Strings - fgets e fputs

- Para ler uma string em arquivos, use a função fgets()

```
char *fgets(char *str, int tamanho, FILE *fp);
```

- Le a string ate encontrar quebra de linha limitada ao parâmetro tamanho
- **Cuidado!** Garanta que tamanho seja suficiente para ler a string inteira

```
int main() {  
    FILE *fp; char str[30], *result;  
    fp = fopen("./dados.txt", "r");  
    if(fp==NULL){ return 0; }  
  
    result = fgets(str, 30, fp);  
    printf("result = %s", result);  
    printf("str = %s", str);  
  
    fclose(fp);  
}
```

Leitura e Escrita em Arquivos

Int float - fscanf e fprintf

- Para escrever float e int em arquivos, use a função fprintf()

```
int fprintf(FILE *fp, char *formato, [lista de valores]);
```

- Utilização idêntica a função printf()
- Utilize fprintf() para char e string quando houverem números a serem escritos na mesma linha

```
int main(){
    FILE *fp;    float fl=3.14;    int x=2;
    fp = fopen("./dados.txt", "w");
    if(fp==NULL){    return 0;    }

    fprintf(fp, "%i %f\n", x, fl);
    printf("%i %f\n", x, fl);

    fclose(fp);
}
```

Leitura e Escrita em Arquivos

Int float - fscanf e fprintf

- Para ler float e int em arquivos, use a função `fscanf()`

```
int fscanf(FILE *fp, char *formato, [lista de valores]);
```

- Utilização idêntica a função `scanf()`
- Utilize `fscanf()` para char e string quando houverem números a serem lidos na mesma linha

```
int main(){
    FILE *fp;    float fl=3.14;    int x=2;
    fp = fopen("./dados.txt", "r");
    if(fp==NULL){    return 0;    }

    fscanf(fp, "%i %f\n", &x, &fl);
    printf("%i %f\n", x, fl);

    fclose(fp);
}
```


MUITO
OBRIGADO

Prof. André del Mestre

www.ifsul.edu.br
almmartins@charqueadas.ifsul.edu.br