Arquivos
Aula 8

Diego Padilha Rubert

Faculdade de Computação
Universidade Federal de Mato Grosso do Sul

Algoritmos e Programação II

Conteúdo da aula

Redirecionamento de entrada e saída

- `prompt$ ./programa > arquivo_saida`
- `prompt$ ./programa < arquivo_entrada`
- `prompt$ ./programa > arquivo_saida < arquivo_entrada`

Redirecionamento de entrada e saída

- `prompt$ ./programa > arquivo_saida`
- `prompt$ ./programa < arquivo_entrada`
- `prompt$ ./programa > arquivo_saida < arquivo_entrada`

Redirecionamento de entrada e saída

- `prompt$ ./programa > arquivo_saida`
- `prompt$ ./programa < arquivo_entrada`
- `prompt$ ./programa > arquivo_saida < arquivo_entrada`

Funções de abertura e fechamento

- `FILE *pt1, *pt2;`
- `FILE *fopen(char *nome, char *modo)`
-

modo	Descrição
r	modo de leitura de texto
w	modo de escrita de texto
a	modo de adicionar texto
r+	modo de leitura e escrita de texto
w+	modo de leitura e escrita de texto
a+	modo de leitura e escrita de texto
b no final	equivalente aos acima, mas em modo binário

- `int fclose(FILE *ptarq)`

Funções de abertura e fechamento

- `FILE *pt1, *pt2;`
- `FILE *fopen(char *nome, char *modo)`

modo	Descrição
r	modo de leitura de texto
w	modo de escrita de texto
a	modo de adicionar texto
r+	modo de leitura e escrita de texto
w+	modo de leitura e escrita de texto
a+	modo de leitura e escrita de texto
b no final	equivalente aos acima, mas em modo binário

- `int fclose(FILE *ptarq)`

Funções de abertura e fechamento

- `FILE *pt1, *pt2;`
- `FILE *fopen(char *nome, char *modo)`
-

modo	Descrição
r	modo de leitura de texto
w	modo de escrita de texto
a	modo de adicionar texto
r+	modo de leitura e escrita de texto
w+	modo de leitura e escrita de texto
a+	modo de leitura e escrita de texto
b no final	equivalente aos acima, mas em modo binário

- `int fclose(FILE *ptarq)`

Funções de abertura e fechamento

- `FILE *pt1, *pt2;`
- `FILE *fopen(char *nome, char *modo)`
modo	Descrição
r	modo de leitura de texto
w	modo de escrita de texto
a	modo de adicionar texto
r+	modo de leitura e escrita de texto
w+	modo de leitura e escrita de texto
a+	modo de leitura e escrita de texto
b no final	equivalente aos acima, mas em modo binário
- `int fclose(FILE *ptarq)`

Funções de abertura e fechamento

```
FILE *f;
f = fopen("entrada.txt", "r");
fclose(f);
```

Funções de entrada e saída

- `int fgetc(FILE *ptarq)`
- `int fputc(int caracter, FILE *ptarq)`
- `int fscanf(FILE *ptarq, char *formato, ...)`
- `int fprintf(FILE *ptarq, char *formato, ...)`
- `fread, fwrite ...`

Funções de entrada e saída

- `int fgetc(FILE *ptarq)`
- `int fputc(int caracter, FILE *ptarq)`
- `int fscanf(FILE *ptarq, char *formato, ...)`
- `int fprintf(FILE *ptarq, char *formato, ...)`
- `fread, fwrite ...`

Funções de entrada e saída

- `int fgetc(FILE *ptarq)`
- `int fputc(int caracter, FILE *ptarq)`
- `int fscanf(FILE *ptarq, char *formato, ...)`
- `int fprintf(FILE *ptarq, char *formato, ...)`
- `fread, fwrite ...`

Funções de entrada e saída

- `int fgetc(FILE *ptarq)`
- `int fputc(int caracter, FILE *ptarq)`
- `int fscanf(FILE *ptarq, char *formato, ...)`
- `int fprintf(FILE *ptarq, char *formato, ...)`
- `fread, fwrite ...`

Funções de entrada e saída

- `int fgetc(FILE *ptarq)`
- `int fputc(int caracter, FILE *ptarq)`
- `int fscanf(FILE *ptarq, char *formato, ...)`
- `int fprintf(FILE *ptarq, char *formato, ...)`
- `fread, fwrite ...`

Funções de entrada e saída

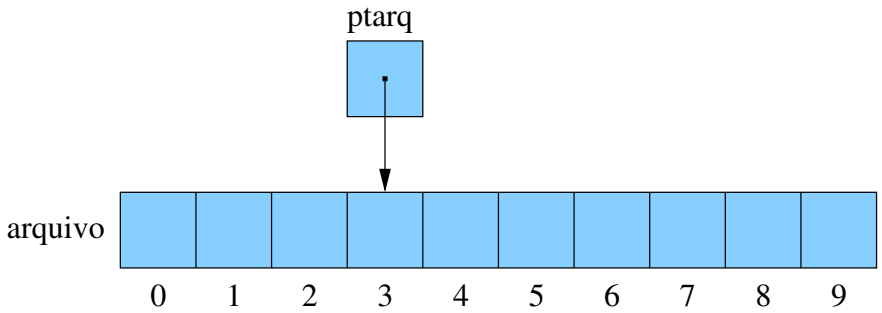
```
int x;
FILE *f;
f = fopen("entrada.txt", "r");

while (fscanf(f, "%d", &x) != EOF)
   printf("Leu: %d\n", x);

fclose(f);
```

Lendo e escrevendo no mesmo arquivo

Ao escrever “no meio” do arquivo, **NÃO** “empurramos” o texto como ocorre em editores de texto, o texto seguinte é **sobrescrito**



Funções de controle

- `int fflush(FILE *ptarq)`

- `int feof(FILE *ptarq)`

- ```
  int fgetpos(FILE *ptarq, fpos_t *pos)
  int fsetpos(FILE *ptarq, fpos_t *pos)
  ```

- `long int ftell(FILE *ptarq)`

- `int fseek(FILE *ptarq, long int desloca, int apartir)`

SEEK_SET	A partir do início do arquivo
SEEK_CUR	A partir da posição atual
SEEK_END	A partir do fim do arquivo

- `void rewind(FILE *ptarq)`
  ```

Funções de controle

- `int fflush(FILE *ptarq)`
- `int feof(FILE *ptarq)`
-
```
int fgetpos(FILE *ptarq, fpos_t *pos)
int fsetpos(FILE *ptarq, fpos_t *pos)
```
-
```
long int ftell(FILE *ptarq)
```
-
```
int fseek(FILE *ptarq, long int desloca, int apartir)
```

SEEK_SET	A partir do início do arquivo
SEEK_CUR	A partir da posição atual
SEEK_END	A partir do fim do arquivo

-
```
void rewind(FILE *ptarq)
```

Funções de controle

- `int fflush(FILE *ptarq)`
- `int feof(FILE *ptarq)`
- `int fgetpos(FILE *ptarq, fpos_t *pos)`
  `int fsetpos(FILE *ptarq, fpos_t *pos)`
- `long int ftell(FILE *ptarq)`
- `int fseek(FILE *ptarq, long int desloca, int apartir)`

SEEK_SET	A partir do início do arquivo
SEEK_CUR	A partir da posição atual
SEEK_END	A partir do fim do arquivo

- `void rewind(FILE *ptarq)`

Funções de controle

- `int fflush(FILE *ptarq)`
- `int feof(FILE *ptarq)`
- `int fgetpos(FILE *ptarq, fpos_t *pos)`
  `int fsetpos(FILE *ptarq, fpos_t *pos)`
- `long int ftell(FILE *ptarq)`
- `int fseek(FILE *ptarq, long int desloca, int apartir)`

SEEK_SET	A partir do início do arquivo
SEEK_CUR	A partir da posição atual
SEEK_END	A partir do fim do arquivo

- `void rewind(FILE *ptarq)`

Funções de controle

- `int fflush(FILE *ptarq)`
- `int feof(FILE *ptarq)`
- `int fgetpos(FILE *ptarq, fpos_t *pos)`
  `int fsetpos(FILE *ptarq, fpos_t *pos)`
- `long int ftell(FILE *ptarq)`
- `int fseek(FILE *ptarq, long int desloca, int apartir)`

`SEEK_SET`	`A partir do início do arquivo`
`SEEK_CUR`	`A partir da posição atual`
`SEEK_END`	`A partir do fim do arquivo`

- `void rewind(FILE *ptarq)`

Funções de controle

- `int fflush(FILE *ptarq)`
- `int feof(FILE *ptarq)`
- `int fgetpos(FILE *ptarq, fpos_t *pos)`
  `int fsetpos(FILE *ptarq, fpos_t *pos)`
- `long int ftell(FILE *ptarq)`
- `int fseek(FILE *ptarq, long int desloca, int apartir)`

`SEEK_SET`	`A partir do início do arquivo`
`SEEK_CUR`	`A partir da posição atual`
`SEEK_END`	`A partir do fim do arquivo`

- `void rewind(FILE *ptarq)`

Funções de controle

```
int x;
char c;
FILE *f;
f = fopen("entrada.txt", "r+");

fscanf(f, "%d", &x);
printf("Pos. atual: %ld\n", ftell(f));
fseek(f, -2, SEEK_END);
fscanf(f, "%c", &c);
printf("Ultima letra: %c\n", c);
rewind(f);
fputc(c, f);

fclose(f);
```

Funções de controle

```
int x;
char c;
FILE *f;
f = fopen("entrada.txt", "r+");

fscanf(f, "%d", &x);
printf("Pos. atual: %ld\n", ftell(f));
fseek(f, -2, SEEK_END);
fscanf(f, "%c", &c);
printf("Ultima letra: %c\n", c);
rewind(f);
fputc(c, f);

fclose(f);
```

```
123
a
b
k
```

```
k23
a
b
k
```