

# SSC0800 - ICC1 – Teórica

---

## Introdução à Ciência da Computação I

### Arquivos

Prof. Claudio Fabiano Motta Toledo: [claudio@icmc.usp.br](mailto:claudio@icmc.usp.br)

# Sumário

---

- Tipos de arquivo
- Modos de abertura
- Funções para manipulação de arquivos
- Posição de escrita e leitura

# Arquivo: tipo texto

---

- *Stream* de texto
- Um arquivo texto é uma sequência de linhas, onde cada linha contém zero ou mais caracteres e termina com um ou mais caracteres que assinalam o fim da linha
- O comprimento máximo de uma linha é 255 caracteres
- Uma linha não é uma *string*, pois seu término não é indicado por '\0'
- Na linguagem C o final de linha é indicado pelo caractere '\n' e ao salvar num arquivo, o '\n' é convertido para CR-LF (Carriage-return/linefeed). Na leitura ocorre o inverso.

# Arquivo: tipo binário

---

- *Stream* binário
- Todos os dados são gravados e lidos exatamente como estão na memória (*memory dump*)
- A visualização direta desses dados em editores de texto não é possível devido a falta de formatação

# Localização e nome de arquivos

---

- Os nomes são armazenados em *strings*
- Deve-se seguir as regras de nomenclatura do sistema operacional
  - Note que no Windows o caminho é indicado pela barra invertida
- O nome do arquivo pode ser informado junto com sua localização no disco
  - Ex: c:\data\conta.txt
- Como em C a barra invertida também tem um significado especial, para representar o caminho numa string é necessário preceder cada barra com outra barra invertida
  - `char *filename = "c:\\data\\conta.txt";`

# Modos de abrir um arquivo

---

- “r” abre um arquivo para leitura. O arquivo deve existir.
- “w” cria um arquivo vazio para escrita. Se o arquivo já existir o seu conteúdo é apagado e o arquivo é tratado como se fosse novo.
- “a” insere dados no arquivo. Operações de escrita adicionam dados no final do arquivo. Se o arquivo não existir o mesmo é criado
- “r+” abre um arquivo para leitura e escrita. O arquivo deve existir.
- “w+” cria um arquivo vazio para leitura e escrita. Se o arquivo já existir o seu conteúdo é apagado e o arquivo é tratado como se fosse novo.
- “a+” abre um arquivo para leitura e inserção. Todas as escritas são realizadas no final do arquivo.

# Exemplo

---

```
FILE *fp;
char *filename = "conta.txt";
char *mode = "w";
if ((fp = fopen(filename,mode)) != NULL)
    printf("Arquivo aberto com sucesso \n");
else
    printf("Erro na abertura do arquivo \n");
fclose(fp);
```

fopen retorna um ponteiro para o tipo FILE, que é uma estrutura declarada em STDIO.H

# Funções para manipulação de arquivos do tipo texto

---

- Entrada e saída de dados formatados
- Função de saída:  
`int fprintf ( FILE * stream, const char * format, ... );`
- Função de entrada:  
`int fscanf ( FILE * stream, const char * format, ... );`
- O *fprintf* e o *fscanf* funcionam de modo semelhante ao *printf* e o *scanf*, porém a partir de um *stream* especificado, e não do *stdin* (padrão)



# Exemplo com fprintf

---

```
FILE *fp;
char *filename = "conta";
char *mode = "w";
float salario = 4000.25;
char *nome = "Joao";
float aux_nome[5], aux_sal;

if ((fp = fopen(filename,mode)) != NULL){
    printf("Arquivo texto aberto com sucesso \n");
    fprintf(fp,"%s %f\n", nome, salario);
}
else{
    printf("Erro na abertura do arquivo texto \n");
}
fclose(fp);
```

# Exemplo com fscanf

---

```
FILE *fp;
char *filename = "conta";
char *mode = "r";
float salario = 4000.25;
char *nome = "Joao";
float aux_nome[5], aux_sal;

if ((fp = fopen(filename,mode)) != NULL){
    fscanf(fp,"%s %f", aux_nome, &aux_sal);
    printf("Salario = %s %f\n", aux_nome, aux_sal);
}
else{
    printf("Erro na abertura do arquivo texto \n");
}
fclose(fp);
```

# Funções para manipulação de arquivos do tipo texto

---

- Entrada e saída de caracteres
- Entrada
  - `int getc (FILE * stream );`
  - `int fgetc (FILE * stream );`
  - `char *fgets (char* str, int num, FILE *stream );`
    - <http://www.cplusplus.com/reference/cstdio/fgets/>
  - `getc` pode ser implementada como macro, enquanto que `fgetc` não

Exemplo para inserir no código visto anteriormente

```
int c;  
do{  
    c = getc (fp);  
    printf ("%c\n",c);  
} while (c != EOF);
```

# Entrada e saída de caracteres

---

- Saída

- int putc (int character, FILE \* stream );

- <http://www.cplusplus.com/reference/cstdio/putc/>

- int fputs (const char \* str, FILE \* stream );

- <http://www.cplusplus.com/reference/cstdio/fputs/>

# Funções de reposicionamento

---

- Para obter a posição atual de leitura/escrita
  - `long int ftell ( FILE * stream );`
- Para retornar à posição zero
  - `void rewind ( FILE * stream );`
- Mover para uma posição específica
  - `int fseek ( FILE * stream, long int offset, int origin );`
  - <http://www.cplusplus.com/reference/clibrary/cstdio/fseek/>

# fseek()

---

```
FILE * pFile;  
pFile = fopen ( "example.txt" , "w" );  
fputs ( "This is an apple." , pFile );  
fseek ( pFile , 9 , SEEK_SET );  
fputs ( " sam" , pFile );  
fclose ( pFile );
```

SEEK\_SET: início do arquivo  
SEEK\_CUR: posição atual  
SEEK\_END: final do arquivo

# Referência

---

- <http://www.cplusplus.com/reference/library/>

---

# FIM Aula 19