



Engenharia da Computação
www.eComp.Poli.br

Arquivos

Disciplina: DCExt Programação Imperativa

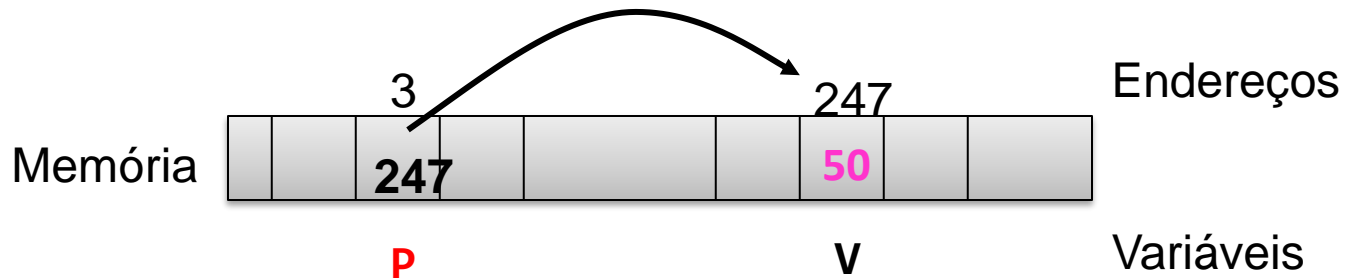
Prof. Hemir Santiago

hcs2@poli.br

Material cedido pelo Prof. Joabe Jesus

Tipo Ponteiro/**Endereço**

- **MANIPULAM** (consultam ou modificar) **ENDEREÇO** indicado



Programa

```

int V;
int* P;
V = 10;
P = &V;
*P = 50;
  
```

```

// alocando/reservando espaço V
// Tipo PONTEIRO para variável int
// armazenando valor 10 em V
// P recebe o endereço da variável V
// P altera o valor de V para 50
  
```

Operador de Indireção *

- Manipula/Acessa o endereço apontado por um ponteiro

```
int x=1, y=2, z[3];
int *ip;           // ip é um ponteiro para int
ip = &x;           // ip aponta para o endereço de x
y = *ip;           // y = valor apontado por ip (ou seja, 1)
*ip = 0;
ip = &z[0];
*ip = 10;
```

x	0
y	1
ip	
z[0]	10
z[1]	?
z[2]	?
⋮	⋮



Situação Problema 1

- Suponha o código abaixo:

$x = 5;$

$y = x;$

$x = x + 1;$

– Como atualizar o valor de Y sempre que X for alterado para que *$X = Y$ sempre* ?

- **Solução 1:** Qualquer alteração em X deve ser seguida pela linha $Y = X$
- **Solução 2:** Usar Y como ponteiro

Solução 1 – cópia para y

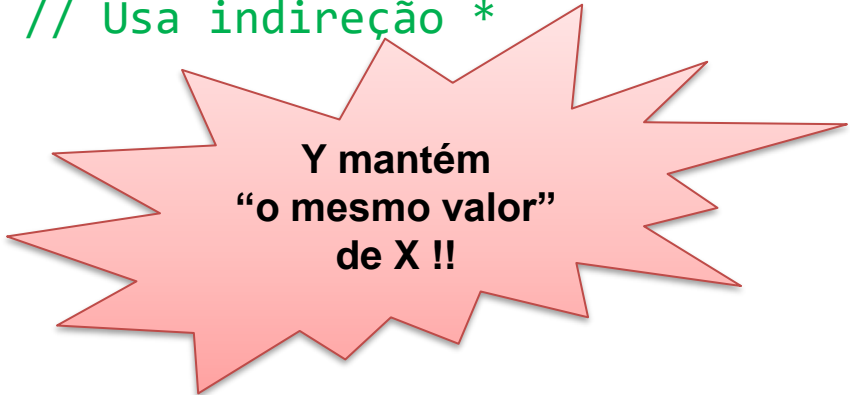
```
int x, y;  
x = 5;  
y = x; // Modificou x acima, então faz CÓPIA para y  
printf("X = %d, Y= %d\n", x, y);  
x = x + 1;  
y = x; // Modificou x acima, então faz CÓPIA para y  
printf("X = %d, Y= %d\n", x, y);
```

Solução 2 – y é um Ponteiro

```
int x, *y = &x; // y aponta para x, independente do valor de x  
x = 5;
```

```
printf("X = %d, Y= %d\n", x, *y); // Usa indireção *  
x = x + 1;
```

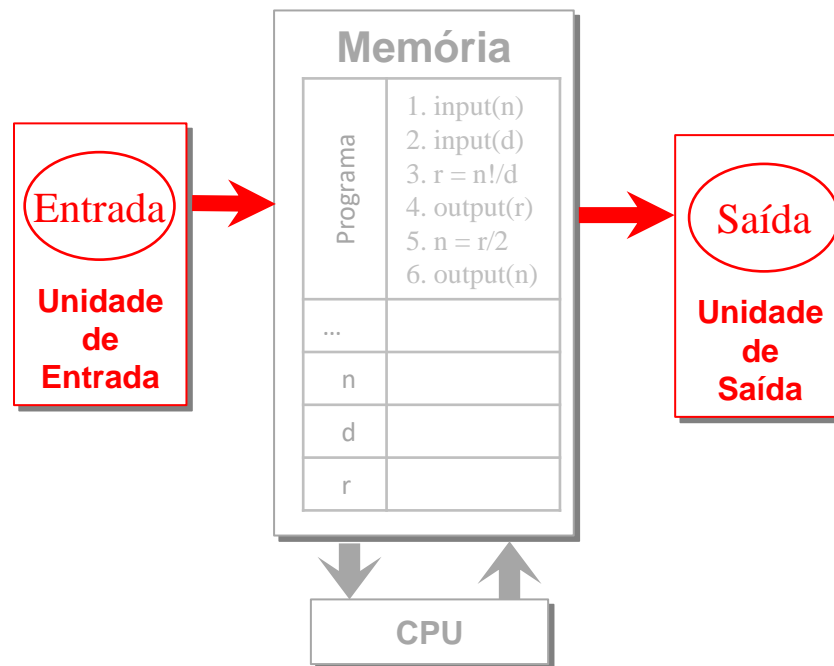
```
printf("X = %d, Y= %d\n", x, *y); // Usa indireção *
```



**Y mantém
“o mesmo valor”
de X !!**

Files

ARQUIVOS (FICHEIROS)



Arquivos

- Armazenamento primário – memória
 - Programas em execução e suas variáveis
- Armazenamento secundário – discos
 - Executáveis dos programas
 - Arquivos que armazenam dados
- Programas podem manipular dados de um arquivo
 - Leitura, escrita
 - Disco -> Memória, Memória -> Disco

Operações com Arquivos em Disco

- Existem dois tipos de arquivo: texto e binário
- Arquivo **texto**:
 - Sequência de caracteres agrupados em linhas
 - Linhas separadas por um caractere denominado LF
 - No Windows, as linhas são separadas por CR/LF
 - Windows fornece uma indicação de fim de arquivo ao programa quando ele tenta ler alguma informação após último caractere

Operações com Arquivos em Disco

- Arquivo **binário**:
 - Nenhuma conversão é feita
 - qualquer caractere é lido ou gravado sem alteração
 - Nenhuma indicação de fim de arquivo é reconhecida
 - dados são armazenados da mesma forma que representados na memória

Abrindo arquivos

```
FILE *fptr; /* ponteiro para arquivo */
```

```
fptr = fopen("arqtext.txt", "w");
```

nome do arquivo

Tipo de abertura

- “r”** Abrir arquivo texto para leitura. O arquivo deve estar presente no disco.
- “w”** Abrir arquivo texto para gravação. Se o arquivo existir ele será destruído e reinicializado. Se não existir, será criado.
- “a”** Abrir um arquivo texto para gravação. Os dados serão adicionados no fim do arquivo existente, ou cria um novo.

Abrindo arquivos

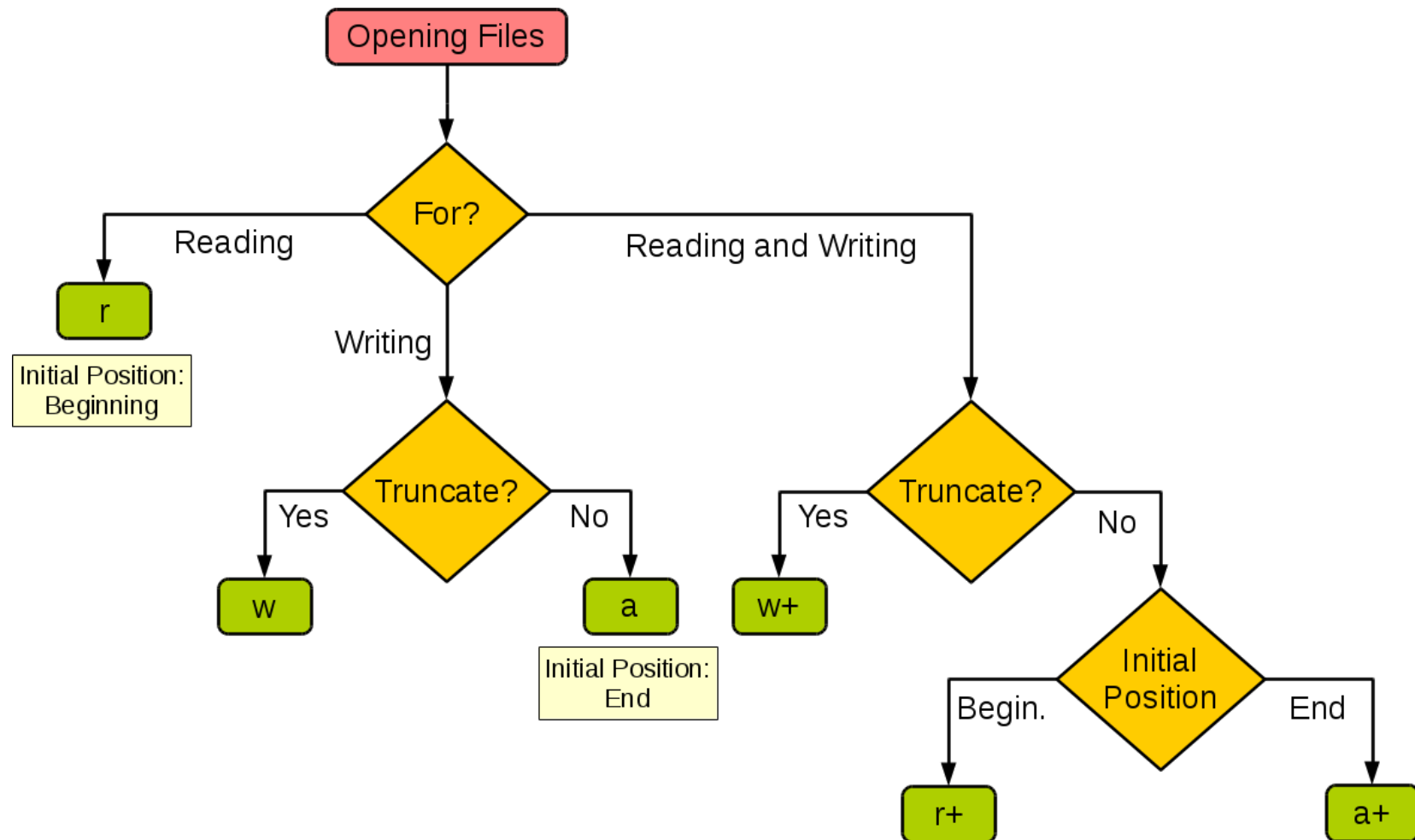
```
FILE *fptr; /* ponteiro para arquivo */  
fptr = fopen("arqtext.txt", "w+");
```

nome do
arquivo

Tipo de abertura

- “r+”** Abrir arquivo texto para leitura e gravação. O arquivo deve existir e pode ser **atualizado**.
- “w+”** Abrir arquivo texto para leitura e gravação. Se o arquivo existir ele será **destruído e reinicializado**. Se não existir, será criado.
- “a+”** Abrir um arquivo texto para atualização e para adicionar dados no fim do arquivo existente, ou cria um novo.

Abrindo arquivos



Abrindo arquivos

```
FILE *fptr; /* ponteiro para arquivo */  
fptr = fopen("arqtext.arq", "wb");
```

nome do arquivo

Tipo de abertura

- “rb”** Abrir arquivo binário para leitura. O arquivo deve estar presente no disco.
- “wb”** Abrir arquivo binário para gravação. Se o arquivo existir ele será destruído e reinicializado. Se não existir, será criado.
- “ab”** Abrir um arquivo binário para gravação. Os dados serão adicionados no fim do arquivo existente, ou cria um novo.

Abrindo arquivos

```
FILE *fptr; /* ponteiro para arquivo */  
fptr = fopen("arqtext.arq", "wb+");
```

nome do
arquivo

Tipo de abertura

- “rb+”** Abrir arquivo binário para leitura e gravação. O arquivo deve existir e pode ser **atualizado**.
- “wb+”** Abrir arquivo binário para leitura e gravação. Se o arquivo existir ele será **destruído e reinicializado**. Se não existir, será criado.
- “ab+”** Abrir um arquivo binário para atualização e para adicionar dados no fim do arquivo existente, ou cria um novo.

Fechando arquivos

```
FILE *fptr; /* ponteiro para arquivo */  
fptr = fopen("arqtext.txt", "w");  
fclose(fptr);
```


Escrevendo em arquivos

```
FILE *fptr; /* ponteiro para arquivo */  
char ch;  
fptr = fopen("arqtext.txt", "w");  
while((ch=getche( )) != '\r')  
    fputc(ch, fptr);  
fclose(fptr);
```

Lendo arquivos

```
FILE *fptr; /* ponteiro para arquivo */  
char ch;  
fptr = fopen("arqtext.txt", "r");  
while((ch=fgetc(fptr)) != EOF)  
    printf("%c",ch);  
fclose(fptr);
```

Fim do arquivo

Cuidados ao abrir arquivos

- A operação para abertura de arquivos pode falhar !!!
 - Falta de espaço em disco
 - Arquivo ainda não criado
- Se o arquivo não puder ser aberto, a função `fopen()` retorna o valor `NULL`

Cuidados ao abrir arquivos

```
FILE *fptr; /* ponteiro para arquivo */
char ch;

fptr = fopen("arqtext.txt", "r");

if((fptr == NULL)){
    printf("Nao foi possivel abrir o arquivo arqtext.txt");
    exit(1);
}

while((ch=fgetc(fptr)) != EOF)
    printf("%c",ch);

fclose(fptr);
```

Gravando um arquivo linha a linha

```
FILE *fptr; /* ponteiro para arquivo */
char string[80];
fptr = fopen("arqtext.txt", "w");

if(fptr == NULL) {
    printf("Nao foi possivel abrir o arquivo arqtext.txt");
    exit(1);
}

while(strlen(gets(string)) > 1) {
    fputs(string, fptr);
    fputs("\n", fptr);
}
fclose(fptr);
```

Lendo um arquivo linha a linha

```
FILE *fptr; /* ponteiro para arquivo */  
char string[80];  
  
fptr = fopen("arqtext.txt", "r");  
  
if(fptr == NULL) {  
    printf("Nao foi possivel abrir o arquivo arqtext.txt");  
    exit(1);  
}  
  
while((fgets(string,80,fptr)) != NULL)  
    printf("%s",string);  
  
fclose(fptr);
```

Gravando um arquivo de maneira formatada

fprintf é similar à função *printf*, exceto que um ponteiro para **FILE** é incluído como primeiro argumento

```
FILE *fptr;
char titulo[30];
int regnum;
float preco;
fptr = fopen("arqtext.txt", "w");
if(fptr == NULL) {
    printf("Nao foi possivel abrir o arquivo arqtext.txt");
    exit(1);
}
do {
    printf("\nDigite titulo, registro e preco:\n");
    scanf("%s %d %f", titulo, &regnum, &preco);
    fprintf(fptr,"%s %d %.2f \n", titulo, regnum, preco);
} while(strlen(titulo) > 1) ;
fclose(fptr);
```

Arquivos Binários: *fwrite*

```
struct livros {  
    char titulo[30];  
    int regnum;  
} livro;  
  
...  
FILE *fptr;  
fptr = fopen("livros.arq", "wb");  
if(fptr == NULL){...}  
  
do { printf("\n Digite o titulo: ");  
    gets(livro.titulo);  
    printf("\n Digite o registro: ");  
    scanf("%d",&livro.regnum);  
    fwrite(&livro, sizeof(struct livros), 1, fptr);  
    printf("\n Adiciona outro livro (s/n) ?");  
    fflush(stdin);  
} while (getche( ) == 's');  
fclose(fptr);
```


Lendo Estruturas com *fread*

```
struct livros {
    char titulo[30];
    int regnum;
};
...
struct livro livro;
FILE *fptr;
fptr = fopen("livros.arq", "rb");
if(fptr == NULL) {...}

while (fread(&livro, sizeof(struct livros), 1, fptr) == 1) {
    printf("\n Titulo: %s\n", livro.titulo);
    printf("\n Registro: %03d\n", livro.regnum);
}
fclose(fptr);
```

Gravando vetores com *fwrite*

```
int tabela[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
```

```
FILE *fptr;
```

```
if((fptr = fopen("tabela.arq", "wb")) == NULL)  
{...}
```

```
fwrite(tabela, sizeof(int), 10, fptr);
```

```
fclose(fptr);
```

Navegação em arquivos

- *rewind(FILE * p)*
 - Reposiciona o indicador de posição no início do arquivo
 - Arquivo precisa ser aberto no modo leitura/escrita (+), se precisar ler e escrever ao mesmo tempo
- *fseek(FILE *p, long numbytes, int origem)*
 - Modifica o indicador de posição para qualquer localidade dentro do arquivo
 - **origem** usa uma das seguintes constantes (*stdio.h*)
 - SEEK_SET: início do arquivo
 - SEEK_CUR: posição atual
 - SEEK_END: final do arquivo

Navegação em arquivos

```
char str[80];  
FILE *fp;  
  
if ((fp = fopen("teste.txt", "w+")) == NULL) { .. }  
  
gets(str);  
strcat(str, "\n"); //acrescenta nova linha  
fputs(str, fp);  
/* agora le a string */  
rewind(fp);  
fgets(str, 80, fp);  
printf(str);  
  
fclose(fp);
```

Navegação em arquivos

```
int main () {  
    FILE *fp;  
  
    fp = fopen("file.txt","w+");  
    fputs("This is tutorialspoint.com", fp);  
  
    fseek(fp, 7, SEEK_SET);  
    fputs(" C Programming Language", fp);  
    fclose(fp);  
  
    return(0);  
}
```

Exercício 01

- Escreva um programa que grava um texto qualquer em um arquivo e, em seguida, conta o número de caracteres do arquivo
 - Use a função *rewind(FILE * p)* para “rebobinar” o arquivo

Exercício 02

- Escreva um programa que grave o nome, 1ª nota, 2ª nota e média dos alunos de uma turma em um arquivo.
- Após a gravação do arquivo, ele deve ser novamente aberto para listar os alunos com suas respectivas notas.

	DATA	AULA
1	22/08/2024	Apresentação da disciplina Introdução à Programação Imperativa
2	29/08/2024	Introdução à Linguagem de Programação C
3	05/09/2024	Conceitos Fundamentais
4	12/09/2024	Tipos de Dados Especiais em C
5	19/09/2024	Estruturas Condicionais e de Repetição
6	26/09/2024	Pré-processamento
7	03/10/2024	Registros/Estruturas de Dados
8	10/10/2024	Ponteiros
9	17/10/2024	1º Exercício Escolar

Plano de Aulas

	DATA	AULA
10	24/10/2024	Correção do 1º EE Proposta de projeto
11	31/10/2024	Arquivos
12	07/11/2024	Acompanhamento de projetos 2ª Chamada
13	14/11/2024	Acompanhamento de projetos
14	21/11/2024	Acompanhamento de projetos
15	28/11/2024	Apresentação parcial
16	05/12/2024	Apresentação de projetos
17	12/12/2024	Avaliação Final

Plano de Aulas