



Arquivo Binário – C

Prof. Nilton Cézar de Paula



Arquivo x Stream

- Arquivo é um dispositivo real, podendo ser em disco, terminal, impressora, ...
- Para manipular arquivo em diferentes dispositivos de armazenamento com diferentes formatos usa-se o conceito lógico de *stream* (fluxo) de dados.
- Dois tipos de stream: binário e texto

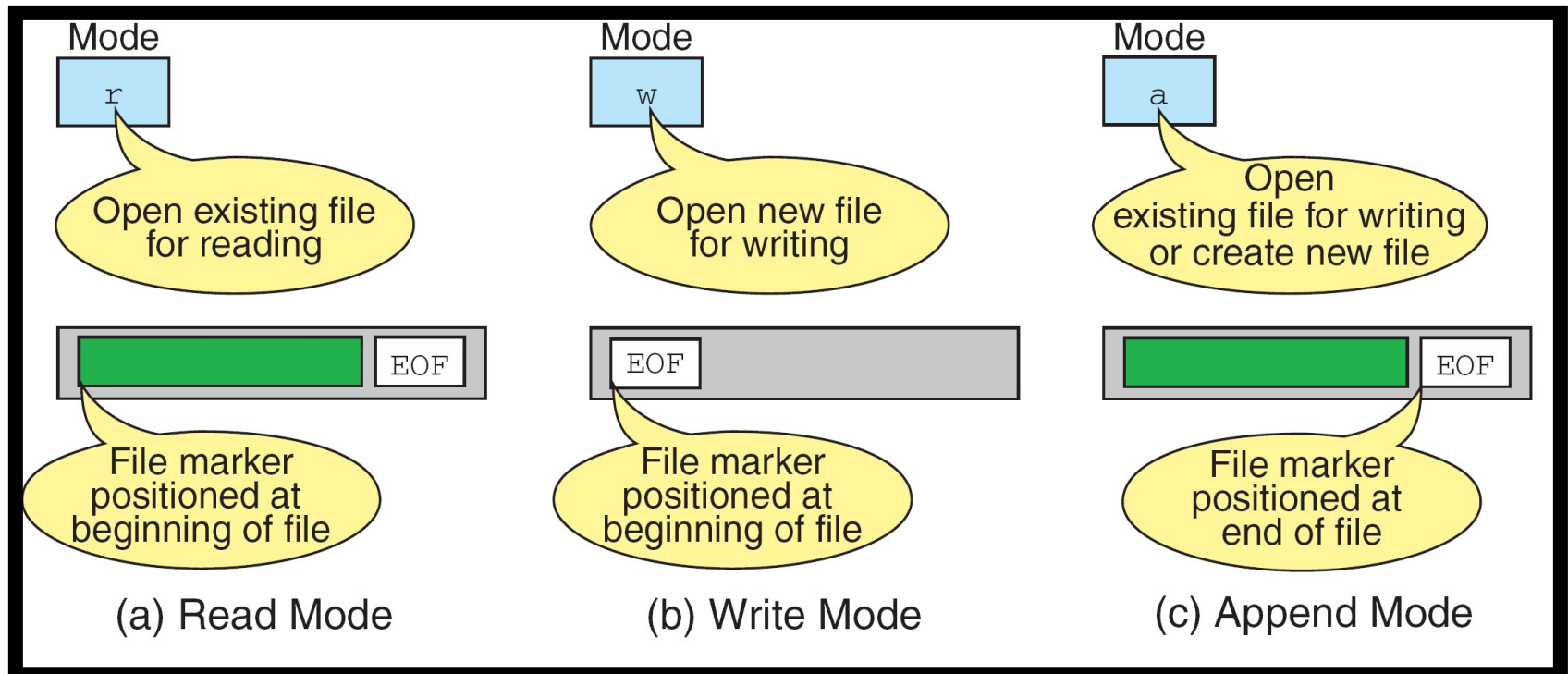
Manipulação Arquivos

- Os protótipos das funções de E/S para acesso a arquivos estão em `<stdio.h>`.
- Um ponteiro para o tipo `FILE` define a estrutura necessária para usar arquivos.
 - `FILE *fp;`
- Três tipos de modos de acesso dizem como será usado um arquivo:
 - “r” : acesso para leitura, devendo existir.
 - “w” : cria para escrita, apagando se existente.
 - “a” : permite anexar no final do arquivo, criando se não existe.

Modos de acesso

Modo	Significado
r	Abre arquivo texto para leitura
w	Cria arquivo texto para escrita
a	Anexa a um arquivo texto
rb	Abre um arquivo binário para leitura
wb	Cria um arquivo binário para escrita
ab	Anexa a um arquivo binário
r+	Abre um arquivo texto para leitura/escrita
w+	Cria um arquivo texto para leitura/escrita
a+	Anexa ou cria um arquivo texto para leitura/escrita
r+b	Abre um arquivo binário para leitura/escrita
w+b	Cria um arquivo binário para leitura/escrita
a+b	Anexa a um arquivo binário para leitura/escrita

Mais sobre modos de acesso



from Figure 7-4 in Forouzan & Gilberg, p. 401

Abrindo e fechando um arquivo binário

- `FILE * fopen (char * filename, char * mode);`
- `fclose (FILE * fp);`

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main() {
5      FILE *fp;
6
7      fp = fopen("exemplo.dat","wb");
8      if(fp == NULL) {
9          printf("\nErro na abertura do arquivo!");
10         exit(100);
11     }
12
13     //usando o arquivo ...
14
15     fclose(fp);
16
17     return 0;
18 }
```


Escrevendo em um arquivo binário

- `int fwrite (void *data, int tam, int num, FILE * fp);`

```
1      #include <stdio.h>
2      #include <stdlib.h>
3
4      int main() {
5          FILE *fp;
6          int i;
7
8          fp = fopen("exemplo.dat","wb");
9          if(fp == NULL) {
10             printf("\nErro na abertura do arquivo!");
11             exit(100);
12         }
13
14         for(i=0;i<10;i++)
15             fwrite(&i,sizeof(i),1,fp);
16
17         fclose(fp);
18
19         return 0;
20     }
```

\exemplos.arq.binario.slide>type exemplo.dat

Lendo de um arquivo binário

- `int fread (void *data, int tam, int num, FILE * fp);`

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main() {
5      FILE *fp;
6      int i, data;
7
8      fp = fopen("exemplo.dat","rb");
9      if(fp == NULL) {
10         printf("\nErro na abertura do arquivo!");
11         exit(100);
12     }
13
14     for(i=0;i<10;i++) {
15         fread(&data,sizeof(data),1,fp);
16         printf("%d ",data);
17     }
18
19     fclose(fp);
20
21     return 0;
22 }
```

0 1 2 3 4 5 6 7 8 9

Lendo de um arquivo binário sem quantidade de dados

- `int feof (FILE * fp);`

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main() {
5      FILE *fp;
6      int i, data;
7
8      fp = fopen("exemplo.dat","rb");
9      if(fp == NULL) {
10         printf("\nErro na abertura do arquivo!");
11         exit(100);
12     }
13
14     while( !feof(fp) ) {
15         if( fread(&data,sizeof(data),1,fp) > 0 )
16             printf("%d ",data);
17     }
18
19     fclose(fp);
20
21     return 0;
22 }
```

Ler um dado não seqüencial

- Uso de um marcador/cursor de posição;
- `int fseek (FILE * fp, long int deslocamento, int origem);`
 - **Origem**
 - **SEEK_SET:** a partir do início do arquivo
 - **SEEK_CUR:** a partir da posição atual
 - **SEEK_END:** a partir do final do arquivo

```
//acessando o 4º dado
fseek(fp,sizeof(int)*3,SEEK_SET);
fread(&i,sizeof(i),1,fp);
printf("%d ",i);

//acessando o penúltimo dado
fseek(fp,sizeof(int)*-2,SEEK_END);
fread(&i,sizeof(i),1,fp);
printf("%d ",i);
```

struct e arquivo

```
struct EMPRESA {
    int id;
    char nome[40];
    char cep[11];
};

FILE *abreArq(char *nome_arq, char *modo)
{
}

void fechaArq(FILE *f, char *nome_arq)
{
}

void leEmpresa(struct EMPRESA *reg) {
}

void Cadastro(FILE *f) {
}

void imprimeEmpresa(struct EMPRESA reg) {
}

void Relatorio(FILE *f) {
}

int main()
{
    FILE *pfileTemp;  char filename[]="c:\\temp\\empresa.dat";

    pfileTemp=abreArq(filename, "w+b");
    Cadastro(pfileTemp);
    Relatorio(pfileTemp);
    fechaArq(pfileTemp, filename);
    return 0;
}
```

struct e arquivo

```
void leEmpresa(struct EMPRESA *reg) {  
  
    printf("\nID   : "); scanf("%d",&reg->id);  
    printf("NOME: "); scanf("%s",reg->nome);  
    printf("CEP  : "); scanf("%s",reg->cep);  
}  
  
void Cadastro(FILE *f) {  
    struct EMPRESA emp;  
    char simnao;  
  
    printf("Cadastrando\n");  
    do {  
        fflush(stdin);  
        leEmpresa(&emp);  
        fwrite(&emp,sizeof(emp),1,f);  
        fflush(stdin);  
        printf("\nContinua (S/N) ?");  
        scanf("%c",&simnao);  
    } while( (simnao == 's') || (simnao == 'S') );  
}
```

```
Cadastrando  
ID   : 10  
NOME: Melo  
CEP  : 87.700-900  
Continua (S/N) ?s  
ID   : 5  
NOME: Dias_Almeida  
CEP  : 80.000-000  
Continua (S/N) ?s  
ID   : 65  
NOME: Manso  
CEP  : 05.367-100  
Continua (S/N) ?n
```

struct e arquivo

```
void imprimeEmpresa(struct EMPRESA reg) {  
  
    printf("\nID   : "); printf("%d", reg.id);  
    printf("\nNOME: "); printf("%s", reg.nome);  
    printf("\nCEP  : "); printf("%s\n", reg.cep);  
}  
  
void Relatorio(FILE *f) {  
    struct EMPRESA emp;  
  
    printf("Relatorio\n");  
    fseek(f, 0, SEEK_SET);  
    do {  
        if( fread(&emp, sizeof(emp), 1, f) > 0 )  
            imprimeEmpresa(emp);  
    } while( !feof(f) );  
}
```

Relatorio

ID : 10
NOME: Melo
CEP : 87.700-900

ID : 5
NOME: Dias Almeida
CEP : 80.000-000

ID : 65
NOME: Manso
CEP : 05.367-100

struct e arquivo

```
void imprimeEmpresa(struct EMPRESA reg) {  
  
    printf("\nID   : "); printf("%d", reg.id);  
    printf("\nNOME: "); printf("%s", reg.nome);  
    printf("\nCEP  : "); printf("%s\n", reg.cep);  
}  
  
void Relatorio(FILE *f) {  
    struct EMPRESA emp;  
  
    printf("Relatorio\n");  
    fseek(f, 0, SEEK_SET);  
    do {  
        if( fread(&emp, sizeof(emp), 1, f) > 0 )  
            imprimeEmpresa(emp);  
    } while( !feof(f) );  
}
```

Relatorio

ID : 10
NOME: Melo
CEP : 87.700-900

ID : 5
NOME: Dias Almeida
CEP : 80.000-000

ID : 65
NOME: Manso
CEP : 05.367-100

struct e arquivo

```
alteraCepEmpresa(pfileTemp,5,"99.999-999");
```

```
void alteraCepEmpresa(FILE *f, int qualIDEmpresa, char *cep) {  
    struct EMPRESA emp;  
    int achou=0;  
  
    printf("Alterar\n");  
    fseek(f,0,SEEK_SET);  
    do {  
        if( fread(&emp,sizeof(emp),1,f) > 0 )  
            if( emp.id == qualIDEmpresa )  
                achou=1;  
    } while( !feof(f) && achou == 0);  
    if( achou == 1 ) {  
        fseek(f,sizeof(emp) * -1,SEEK_CUR);  
        strcpy(emp.cep,cep);  
        fwrite(&emp,sizeof(emp),1,f);  
    }  
}
```

Relatorio - ANTES

ID : 10
NOME: Melo
CEP : 87.700-900

ID : 5
NOME: Dias_Almeida
CEP : 80.000-000

ID : 65
NOME: Manso
CEP : 05.367-100
Altera

Relatorio - DEPOIS

ID : 10
NOME: Melo
CEP : 87.700-900

ID : 5
NOME: Dias_Almeida
CEP : 99.999-999

ID : 65
NOME: Manso
CEP : 05.367-100