

# PROGRAMAÇÃO IMPERATIVA

Prof. Dr. Edson J. R. Justino

Prof. Dr. Alcides Calsavara

Arquivo Binário

Pontifícia Universidade Católica do Paraná – PUCPR  
Escola Politécnica

# Arquivo binário contendo apenas um caracter

um byte

abre arquivo binário  
para escrita

```
char ch = 'Z';  
  
FILE* arq = fopen("teste.bin", "wb");  
fwrite(&ch, 1, 1, arq);  
fclose(arq);
```

um byte

uma vez

Escreve um char no arquivo  
arquivo\_binario\_0w.c

abre arquivo binário  
para leitura

```
char ch;  
  
FILE* arq = fopen("teste.bin", "rb");  
fread(&ch, 1, 1, arq);  
fclose(arq);  
  
printf("ch = %c\n", ch);
```

Lê um char do arquivo  
arquivo\_binario\_0r.c

# Arquivo binário contendo apenas um inteiro

quatro bytes

abre arquivo binário  
para escrita

```
int k = 999;

FILE* arq = fopen("teste.bin", "wb");
fwrite(&k, 4, 1, arq);
fclose(arq);
```

quatro bytes

uma vez

Escreve um inteiro no arquivo  
arquivo\_binario\_1w.c

abre arquivo binário  
para leitura

```
int k;

FILE* arq = fopen("teste.bin", "rb");
fread(&k, 4, 1, arq);
fclose(arq);

printf("k = %d\n", k);
```

Lê um inteiro do arquivo  
arquivo\_binario\_1r.c

# Arquivo binário contendo uma sequência de caracteres

escrita

```
#define MAX 10
```

MAX bytes

```
int main()  
{
```

sequência de  
caracteres

```
    char palavra[MAX];
```

```
    for (int i = 0; i < MAX; i++)
```

```
        palavra[i] = i + 'a';
```

```
    FILE* arq = fopen("teste.bin", "wb");
```

```
    fwrite(palavra, 1, MAX, arq);
```

```
    fclose(arq);
```

um byte

MAX vezes

Escreve MAX caracteres no arquivo  
arquivo\_binario\_2w.c

leitura

```
#define MAX 10
```

```
int main()  
{
```

```
    char palavra[MAX];
```

```
    FILE* arq = fopen("teste.bin", "rb");
```

```
    fread(palavra, 1, MAX, arq);
```

```
    fclose(arq);
```

```
    for (int i = 0; i < MAX; i++)
```

```
        printf("%c", palavra[i]);
```

```
    putchar('\n');
```

LÊ MAX caracteres do arquivo  
arquivo\_binario\_2r.c

# Arquivo binário contendo uma sequência de inteiros - escrita

```
#define QUANTIDADE 10
```

```
int main()
```

```
{
```

```
    int numeros[QUANTIDADE];
```

```
    for (int i = 0; i < QUANTIDADE; i++) numeros[i] = i * 2;
```

```
    const int ESPACO_UNITARIO = sizeof(int);
```

```
    FILE* arq = fopen("teste.bin", "wb");
```

```
    fwrite(numeros, ESPACO_UNITARIO, QUANTIDADE, arq);
```

```
    fclose(arq);
```

sequência de inteiros

espaço ocupado por cada elemento da sequência

Escreve QUANTIDADE inteiros no arquivo

arquivo\_binario\_3w.c

## Arquivo binário contendo uma sequência de inteiros - leitura

```
#define QUANTIDADE 10

int main()
{
    int numeros[QUANTIDADE];

    const int ESPACO_UNITARIO = sizeof(int);
    FILE* arq = fopen("teste.bin", "rb");
    fread(numeros, ESPACO_UNITARIO, QUANTIDADE, arq);
    fclose(arq);

    for (int i = 0; i < QUANTIDADE; i++) printf("%d\n", numeros[i]);
}
```

Lê QUANTIDADE inteiros do arquivo

arquivo\_binario\_3r.c



Arquivo binário contendo  
uma instância  
de uma estrutura - escrita

Escreve uma estrutura no arquivo

```
typedef struct {  
    char titulo[50];  
    char autor[40];  
    int numero_paginas;  
    int ano_publicacao;  
} Livro;  
  
int main()  
{  
    Livro L;  
    strcpy(L.titulo, "Ensaio sobre a cegueira");  
    strcpy(L.autor, "José Saramago");  
    L.numero_paginas = 310;  
    L.ano_publicacao = 1995;  
  
    FILE* arq = fopen("teste.bin", "wb");  
    fwrite(&L, sizeof(Livro), 1, arq);  
    fclose(arq);  
}
```

Arquivo binário contendo  
uma instância  
de uma estrutura - leitura

Lê uma estrutura do arquivo

arquivo\_binario\_4r.c

```
typedef struct {
    char titulo[50];
    char autor[40];
    int numero_paginas;
    int ano_publicacao;
} Livro;

int main()
{
    Livro L;

    FILE* arq = fopen("teste.bin", "rb");
    fread(&L, sizeof(Livro), 1, arq);
    fclose(arq);

    printf("%s\n", L.titulo);
    printf("%s\n", L.autor);
    printf("%d\n", L.numero_paginas);
    printf("%d\n", L.ano_publicacao);
}
```



Arquivo binário contendo  
uma sequência de instâncias  
de uma estrutura - escrita

```
#define QUANTIDADE 3

typedef struct {
    char titulo[50];
    char autor[40];
    int numero_paginas;
    int ano_publicacao;
} Livro;
```

Escreve QUANTIDADE  
instâncias de  
uma estrutura no arquivo

arquivo\_binario\_5w.c

```
Livro biblioteca[QUANTIDADE];
```

```
strcpy(biblioteca[0].titulo, "Ensaio sobre a cegueira");
strcpy(biblioteca[0].autor, "José Saramago");
biblioteca[0].numero_paginas = 310;
biblioteca[0].ano_publicacao = 1995;
```

```
strcpy(biblioteca[1].titulo, "Cem sonetos de amor");
strcpy(biblioteca[1].autor, "Pablo Neruda");
biblioteca[1].numero_paginas = 128;
biblioteca[1].ano_publicacao = 1959;
```

```
strcpy(biblioteca[2].titulo, "Sentimento do mundo");
strcpy(biblioteca[2].autor, "Carlos Drummond de Andrade");
biblioteca[2].numero_paginas = 101;
biblioteca[2].ano_publicacao = 1940;
```

```
FILE* arq = fopen("teste.bin", "wb");
fwrite(&biblioteca, sizeof(Livro), QUANTIDADE, arq);
fclose(arq);
```

quantidade de  
instâncias

tamanho de uma instância

Arquivo binário contendo  
uma sequência de instâncias  
de uma estrutura - leitura

```
#define QUANTIDADE 3

typedef struct {
    char titulo[50];
    char autor[40];
    int numero_paginas;
    int ano_publicacao;
} Livro;
```

Lê QUANTIDADE  
instâncias de  
uma estrutura do arquivo

```
Livro biblioteca[QUANTIDADE];

FILE* arq = fopen("teste.bin", "rb");
fread(biblioteca, sizeof(Livro), QUANTIDADE, arq);
fclose(arq);

for (int i = 0; i < QUANTIDADE; i++)
{
    printf("%s\n", biblioteca[i].titulo);
    printf("%s\n", biblioteca[i].autor);
    printf("%d\n", biblioteca[i].numero_paginas);
    printf("%d\n", biblioteca[i].ano_publicacao);
    putchar('\n');
}
```

Arquivo Binário

Leitura por repetição



# Arquivo binário contendo uma sequência de caracteres

escrita

```
#define MAX 10

int main()
{
    char palavra[MAX];
    for (int i = 0; i < MAX; i++)
        palavra[i] = i + 'a';

    FILE* arq = fopen("teste.bin", "wb");
    fwrite(palavra, 1, MAX, arq);
    fclose(arq);
}
```

MAX bytes

sequência de caracteres

um byte

MAX vezes

arquivo\_binario\_2w.c

leitura sem repetição

```
#define MAX 10

int main()
{
    char palavra[MAX];

    FILE* arq = fopen("teste.bin", "rb");
    fread(palavra, 1, MAX, arq);
    fclose(arq);

    for (int i = 0; i < MAX; i++)
        printf("%c", palavra[i]);
    putchar('\n');
}
```

leitura  
única  
de MAX  
caracteres

arquivo\_binario\_2r.c

Arquivo binário –  
leitura por repetição –  
volume de dados fixo

número de  
caracteres lidos  
por vez

MAX leituras  
de um  
caracter

```
#define MAX 10

int main()
{
    char palavra[MAX];

    FILE* arq = fopen("teste.bin", "rb");
    for (int i = 0; i < MAX; i++)
    {
        int lido = fread(&palavra[i], 1, 1, arq);
        printf("lido = %d\n", lido);
    }

    fclose(arq);

    for (int i = 0; i < MAX; i++)
        printf("%c", palavra[i]);
    putchar('\n');
```

endereço do  
i-ésimo  
caracter



Arquivo binário –  
leitura por repetição –  
volume de dados variável

Lê um caracter por vez

número de  
caracteres lidos  
por vez

número total de  
caracteres lidos

```
char ch;

FILE* arq = fopen("teste.bin", "rb");
int lido = 0;
int total = 0;
do
{
    lido = fread(&ch, 1, 1, arq);
    printf("lido = %d\n", lido);
    total = total + lido;
}
while (lido > 0);
fclose(arq);
printf("total = %d\n", total);
```

endereço de  
um caracter

lê até o fim  
do arquivo  
(lido == 0)

Arquivo binário –  
leitura por repetição –  
volume de dados variável –  
com buffer

Lê BUFFER\_SIZE  
caracteres por vez

lê até o fim  
do arquivo  
(lido < BUFFER\_SIZE)

arquivo\_binario\_2rC.c

```
#define BUFFER_SIZE 3
```

```
int main()  
{
```

```
    char buffer[BUFFER_SIZE];
```

```
    FILE* arq = fopen("teste.bin", "rb");
```

```
    int lido = 0;
```

```
    int total = 0;
```

```
    do
```

```
    {
```

```
        lido = fread(buffer, 1, BUFFER_SIZE, arq);
```

```
        printf("lido = %d\n", lido);
```

```
        total = total + lido;
```

```
    }
```

```
    while (lido == BUFFER_SIZE);
```

```
    fclose(arq);
```

```
    printf("total = %d\n", total);
```

sequência  
de caracteres

endereço de  
uma sequência  
de caracteres

leitura de uma  
sequência de  
caracteres

Arquivo binário contendo  
uma sequência de instâncias  
de uma estrutura - escrita

```
#define QUANTIDADE 3

typedef struct {
    char titulo[50];
    char autor[40];
    int numero_paginas;
    int ano_publicacao;
} Livro;
```

Escreve de uma só vez  
QUANTIDADE  
ocorrências da estrutura  
Livro no arquivo

arquivo\_binario\_5w.c

```
Livro biblioteca[QUANTIDADE];
```

```
strcpy(biblioteca[0].titulo, "Ensaio sobre a cegueira");
strcpy(biblioteca[0].autor, "José Saramago");
biblioteca[0].numero_paginas = 310;
biblioteca[0].ano_publicacao = 1995;
```

```
strcpy(biblioteca[1].titulo, "Cem sonetos de amor");
strcpy(biblioteca[1].autor, "Pablo Neruda");
biblioteca[1].numero_paginas = 128;
biblioteca[1].ano_publicacao = 1959;
```

```
strcpy(biblioteca[2].titulo, "Sentimento do mundo");
strcpy(biblioteca[2].autor, "Carlos Drummond de Andrade");
biblioteca[2].numero_paginas = 101;
biblioteca[2].ano_publicacao = 1940;
```

```
FILE* arq = fopen("teste.bin", "wb");
fwrite(&biblioteca, sizeof(Livro), QUANTIDADE, arq);
fclose(arq);
```

quantidade de  
elementos

tamanho de um elemento



Arquivo binário contendo  
uma sequência de instâncias  
de uma estrutura –  
leitura por repetição –  
volume de dados variável –  
com buffer

Lê do arquivo QUANTIDADE ocorrências da  
estrutura Livro por vez

```
Livro buffer[BUFFER_SIZE];

FILE* arq = fopen("teste.bin", "rb");
size_t lido = 0;
size_t total = 0;
do
{
    lido = fread(buffer, sizeof(Livro), BUFFER_SIZE, arq);
    printf("lido = %zu\n", lido);
    total = total + lido;
}
while (lido == BUFFER_SIZE);

fclose(arq);
printf("total = %zu\n", total);
```

número de elementos  
lidos por vez

lê até o fim  
do arquivo  
(lido < BUFFER\_SIZE)

tamanho em bytes de  
cada elemento no buffer

número máximo de  
elementos no buffer