

Arquivos

MC102

Algoritmos e Programação de Computadores

Aula de Laboratório 13

Instituto de Computação
Primeiro Semestre de 2012

04 de junho de 2012



- Assim como em arquivos texto devemos criar um ponteiro:
Ex.: FILE *nome-variavel;
- Após a criação do arquivo podemos associá-lo com um arquivo real do computador usando a função **fopen**:
Ex.:
FILE *arq;
arq = fopen("teste.bin", "rb");



MC102 - Algoritmos e Programação de Computadores

3 / 15

Arquivo Binário

Recursividade

Arquivo Binário

Recursividade

Conteúdo

1 Arquivo Binário

2 Recursividade

Escrevendo dados - fwrite

- Para escrever dados utilizamos a função **fwrite**:
Ex.: size_t fwrite(void *pt-mem, size_t size, size_t num-items, FILE *pt-arq);

Escrevendo no formato binário

```
FILE *arq;
double aux=2.5;
double vet[]={2.3, 4.7, 9.1}
arq = fopen("arquivo.bin", "w+b");
fwrite(&aux, sizeof(double), 1, arq);
fwrite(vet, sizeof(double), 3, arq);
```



MC102 - Algoritmos e Programação de Computadores

2 / 15



MC102 - Algoritmos e Programação de Computadores

4 / 15

Lendo dados - fread

- Para escrever dados de um arquivo binário utilizamos a função `fread`:

Ex.: `size_t fread(void *pt-mem, size_t size, size_t num-items, FILE *pt-arq);`

Lendo arquivo

```

1 #include <stdio.h>
2 int main(){
3 FILE *arq;
4 double aux=2.5;
5 double aux2=0;
6 arq = fopen("teste.bin", "w+b");
7 fwrite(&aux, sizeof(double), 1, arq);
8 rewind(arq);
9 fread(&aux2, sizeof(double), 1, arq);
10 printf("Conteudo de aux2: %lf, \n", aux2);
11 fclose(arq);
12 }
```



Um programa para cadastro de alunos - Parte 1

Exemplo

```

1 #include <stdio.h>
2 #include <string.h>
3 #define TAM 5 //tamanho do vetor usado como cadastro
4 struct Aluno{
5     char nome[100];
6     int RA;
7 };
8
9 typedef struct Aluno Aluno;
10
11 //Esta funcao imprime todo o conteudo
12 void imprimeArquivo();
13
14 //Nome do arquivo que contem o cadastro
15 char nomeArq[] = "alunos.bin";
```



Arquivos Binários

Exemplo

```

1 #include <stdio.h>
2 int main(){
3 FILE *arq;
4 double aux[]={2.5, 1.4, 3.6}, aux2[3];
5 int i;
6 arq = fopen("teste.bin", "w+b");
7 fwrite(aux, sizeof(double), 3, arq);
8 rewind(arq);
9 fread(aux2, sizeof(double), 3, arq);
10 for(i=0; i<3; i++)
11     printf(" [%d]: %.2lf\n", i, aux2[i]);
12 rewind(arq);
13 i=0;
14 while(fread(&aux2[i], sizeof(double), 1, arq) != 0)
15     i++;
16 for(i=0; i<3; i++)
17     printf(" [%d]: %.2lf\n", i, aux2[i]);
18 fclose(arq);
19 }
```



Um programa para cadastro de alunos - Parte 2

Exemplo

```

1 int main(){
2     FILE *arq;
3     Aluno cadastro[TAM] = {
4         {"Joao", 1}, {"Batata", 2}, {"Ze", 3}, {"Malu",
5         4}, {"Maria", 5} };
6     arq = fopen(nomeArq, "w+b");
7     if(arq == NULL){
8         printf("Erro: Main!\n");
9         return 0;
10    }
11    fwrite(cadastro, sizeof(Aluno), TAM, arq);
12    fclose(arq);
13    imprimeArquivo();
14 }
```



Um programa para cadastro de alunos - Parte 3

Exemplo

```

1 void imprimeArquivo(){
2     Aluno cadastro[TAM];
3     FILE *arq = fopen(nomeArq, "r+b"); //Note que usamos
        r e nao w
4     int i;
5     if(arq == NULL){
6         printf("Erro: Imprime Arquivo!\n");
7         return;
8     }
9     fread(cadastro, sizeof(Aluno), TAM, arq);
10    printf(" ---- Imprimindo Dados ----\n");
11    for(i=0; i<TAM; i++)
12        printf("Nome: %s, RA: %d \n", cadastro[i].nome,
13            cadastro[i].RA);
14    printf("\n");
15    fclose(arq);
16 }

```



Lendo dados do Arquivo

- Para ler dados podemos utilizar a função **fscanf**:
Ex.: `int fscanf(ponteiro, string de formato, variáveis);`

Lendo dados do arquivo.txt

```

char aux;
FILE *arq = fopen("arquivo.txt", "r");
fscanf(arq, "%c", &aux);
printf("%c", aux);

```



Recursão

Recursão

É o processo de resolução de um problema, reduzindo-o em um ou mais subproblemas com as seguintes características:

- 1 São idênticos aos problemas originais;
- 2 São mais simples de resolver.



Recursão

Recursão

É o processo de resolução de um problema, reduzindo-o em um ou mais subproblemas com as seguintes características:

- 1 São idênticos aos problemas originais;
- 2 São mais simples de resolver.

- Para se codificar programas de modo recursivo usa-se uma função, a qual pode chamar a si mesma.



Fatorial - Exemplo

Problema: Calcula o Fatorial de um número ($n!$).

Ex.: $n = 4!$

- $4! = 4*3!$ \Leftarrow subdividimos o problema;



Fatorial - Exemplo

Problema: Calcula o Fatorial de um número ($n!$).

Ex.: $n = 4!$

- $4! = 4*3!$ \Leftarrow subdividimos o problema;
- $3! = 3*2!$ \Leftarrow subdividimos o problema;



Fatorial - Exemplo

Problema: Calcula o Fatorial de um número ($n!$).

Ex.: $n = 4!$

- $4! = 4*3!$ \Leftarrow subdividimos o problema;
- $3! = 3*2!$ \Leftarrow subdividimos o problema;
- $2! = 2*1!$ \Leftarrow subdividimos o problema;



Fatorial - Exemplo

Problema: Calcula o Fatorial de um número ($n!$).

Ex.: $n = 4!$

- $4! = 4*3!$ \Leftarrow subdividimos o problema;
- $3! = 3*2!$ \Leftarrow subdividimos o problema;
- $2! = 2*1!$ \Leftarrow subdividimos o problema;
- $1! = 1$ \Leftarrow Opa!! Temos a solução;
- Chegamos a menos parte do problema!
Esta é a **condição de parada!**



Recursão

Fatorial Não recursivo

```

1 #include <stdio.h>
2 long fatorial(long n){
3     long valor=1;
4     while(n>1){
5         valor = valor * n;
6         n--;
7     }
8     return valor;
9 }
10
11 int main(){
12     long n = 6;
13     printf("O fatorial de %ld e: %ld\n",n, fatorial(n));
14 }

```



Questões?

Obrigado!

Para informação:

Página dos Laboratórios (Tarefas): <http://susy.ic.unicamp.br:9999/mc102ab>

Página do Curso: <http://www.lrc.ic.unicamp.br/~geraldoms/mc102>

E-mail:

geraldoms@lrc.ic.unicamp.br

brhenrique.fischer@gmail.com



Recursão

Fatorial recursivo

```

1 #include <stdio.h>
2 long fatorial(long n){
3     int r,x;
4     if(n <= 1)
5         return 1;
6     else{
7         x = n-1;
8         r = fatorial(x);
9         return (n*r);
10    }
11 }
12
13 int main(){
14     long n = 5;
15     printf("O fatorial de %ld e: %ld\n",n, fatorial(n));
16 }

```

