



Exercícios

Organização de Arquivos

SCE-183 – Algoritmos e Estruturas de Dados II



Arquivos

- Organização dos dados nos arquivos viabiliza a obtenção de informações corretas de maneira eficiente
- Má organização
 - Impossibilidade de recuperar os dados
 - Desperdício de espaço
 - Grande esforço no processamento dos dados



Exemplo

- Faça um programa em C que
 1. Leia do usuário os seguintes dados de 2 alunos: nome, endereço e curso
 2. Escreva em um arquivo os dados lidos
 3. Leia do arquivo os dados escritos e imprima na tela

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define TAM 2

/*
 *
 */

typedef struct {
    char nome[50];
    char endereco[50];
    char curso[20];
} Aluno;

int main(int argc, char** argv) {

    int i;
    Aluno a;
    FILE * arq = fopen("saida.txt", "w");

    printf(" ===== LENDO DADOS DA ENTRADA =====\n");

    for(i = 0; i < TAM; i++){
        printf("Digite o nome do aluno:\n");
        scanf("%s", a.nome);
        printf("Digite o endereco do aluno:\n");
        scanf("%s", a.endereco);
        printf("Digite o curso do aluno:\n");
        scanf("%s", a.curso);

        fprintf(arq, "%s", a.nome);
        fprintf(arq, "%s", a.endereco);
        fprintf(arq, "%s", a.curso);

        printf("\n");
    }
}

```

```

fclose (arq);

printf(" ===== LENDO DADOS DO ARQUIVO =====\n");

arq = fopen("saida.txt", "r");

for(i = 0; i < TAM; i++){

    fscanf(arq, "%s", a.nome);
    printf("Nome do aluno: %s\n", a.nome);
    fscanf(arq, "%s", a.endereco);
    printf("Endereço do aluno: %s\n", a.endereco);
    fscanf(arq, "%s", a.curso);
    printf("Curso do aluno: %s\n", a.curso);

    printf("\n");

}

fclose (arq);
return (EXIT_SUCCESS);
}

```

O que esse programa vai imprimir?

Digite o nome do aluno:
Bruno
Digite o endereco do aluno:
S.Carlos
Digite o curso do aluno:
Computacao

Digite o nome do aluno:
Vanessa
Digite o endereco do aluno:
S.Paulo
Digite o curso do aluno:
Computacao

===== LENDO DADOS DO ARQUIVO =====

Nome do aluno: BrunoS.CarlosComputacaoVanessaS.PauloComputacao
Endereco do aluno: S.Paulo
Curso do aluno: Computacao

Nome do aluno: BrunoS.CarlosComputacaoVanessaS.PauloComputacao
Endereco do aluno: S.Paulo
Curso do aluno: Computacao

saida.txt



Last Saved: 23/04/12 10:50:53

File Path: ~/Documents/USP/PAE - 1o 2012/Material Aulas/Aula Arquivos/saida.txt

saida.txt

1

BrunoS.CarlosComputacaoVanessaS.PauloComputacao





Organização de Arquivos

- Informações em arquivos são, em geral, organizadas em **campos** e **registros**
- **Campo**: menor unidade lógica de informação em um arquivo
- **Registro**: conjunto de campos agrupados
- Se pensarmos em uma **struct**, qual seria o conceito correspondente a um **campo**? E a um **registro**?



Métodos para organização em campos

- Comprimento fixo
- Indicador de comprimento
- Delimitadores
- Uso de *tags* (etiquetas)



Métodos para organização em registros

- Tamanho fixo
- Número fixo de campos
- Indicador de tamanho
- Uso de índice
- Utilizar delimitadores



Exercício

- Considere a seguinte estrutura

```
typedef struct {  
    char * nome;  
    char * endereco;  
    char * curso;  
} Aluno;
```

```

#define TAM 20
int main(int argc, char** argv) {

    int i;
    Aluno a[TAM];
    FILE * arq = fopen("saida.txt", "wb");
    char buffer [100];

    printf(" ===== LENDO DADOS DA ENTRADA =====\n");

    for(i = 0; i < TAM; i++){
        printf("Digite o nome do aluno:\n");
        scanf("%s", buffer);
        a[i].nome = (char*) malloc(strlen(buffer));
        strcpy(a[i].nome, buffer);

        printf("Digite o endereco do aluno:\n");
        scanf("%s", buffer);
        a[i].endereco = (char*) malloc(strlen(buffer));
        strcpy(a[i].endereco, buffer);

        printf("Digite o curso do aluno:\n");
        scanf("%s", buffer);
        a[i].curso = (char*) malloc(strlen(buffer));
        strcpy(a[i].curso, buffer);

        fwrite(a[i].nome, sizeof(char[50]), 1, arq);
        fwrite(a[i].endereco, sizeof(char[50]), 1, arq);
        fwrite(a[i].curso, sizeof(char[20]), 1, arq);

        printf("\n");
    }

    fclose (arq);
}

```

O código ao lado armazena, em arquivo, registros da estrutura descrita



Exercício

1) Considerando o código anterior, escreva um programa que leia corretamente os registros contidos em um arquivo "entrada.txt"¹, os imprima na tela e os grave em outro arquivo "saida_1.txt".

Neste novo arquivo, deve-se gravar os arquivos de maneira que os REGISTROS possuam número fixo de campos e os CAMPOS sejam de comprimento variável, precedidos por indicador de tamanho.

```
05Bruno08S.Carlos10Computação07Vanessa07S.Paulo  
10Computação09Alexandre11B.Horizonte09Eng.Civil
```

1 - <http://sites.labic.icmc.usp.br/bmnogueira/entrada.zip>

```

printf(" ===== LENDO DADOS DO ARQUIVO =====\n");

arq = fopen("saida.txt", "rb");

for(i = 0; i < TAM; i++){

    fread ( buffer , sizeof(char[50]),1,arq);
    a[i].nome = (char*) calloc (1,strlen (buffer));
    strcpy (a[i].nome, buffer);
    printf("Nome do aluno: %s\n", a[i].nome);

    fread ( buffer , sizeof(char[50]),1,arq);
    a[i].endereco = (char*) calloc (1,strlen (buffer));
    strcpy (a[i].endereco, buffer);
    printf("Endereco do aluno: %s\n", a[i].endereco);

    fread ( buffer , sizeof(char[20]),1,arq);
    a[i].curso = (char*) calloc (1,strlen (buffer));
    strcpy (a[i].curso, buffer);
    printf("Curso do aluno: %s\n", a[i].curso);

    printf("\n");

}

fclose (arq);

```

Dica para leitura
do arquivo de
entrada para o
exercício 1.



Exercício

2) Leia os registros armazenados no arquivo "saida1.txt", imprima-os na tela e grave-os em um outro arquivo "saida2.txt".

Neste novo arquivo, deve-se gravar os arquivos de maneira que os REGISTROS sejam delimitados pelo caractere "\n" e os CAMPOS sejam de comprimento variável, sendo delimitados pelo caractere "|".

```
Bruno|S.Carlos|Computação|  
Vanessa|S.Paulo|Computação|  
Alexandre|B.Horizonte|Eng.Civil|
```



Exercício

3) Leia os registros armazenados no arquivo "saida2.txt", imprima-os na tela e grave-os em um outro arquivo "saida3.txt".

Neste novo arquivo, deve-se gravar os arquivos de maneira que os REGISTROS sejam delimitados pelo caractere "\n" e os CAMPOS sejam precedidos por marcadores (Nome=, Endereco= e Curso=) e sucedidos pelo caractere "|".

```
Nome=Bruno|Endereco=S.Carlos|Curso=Computação|
Nome=Vanessa|Endereco=S.Paulo|Curso=Computação|
Nome=Alexandre|Endereco=B.Horizonte|Curso=Eng.Civil|
```



Exercício

Implementações das respostas e os respectivos arquivos de saída devem ser compactados e postados no Moodle.