

# Estruturas de Dados II

## Registros (structs)

## Manipulação de Arquivos Binários

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello world!" << endl;
    return 0;
}
```

# STRUCT - REGISTROS

As estruturas de dados consistem em criar apenas um dado que contém vários membros, que nada mais são do que outras variáveis.

De uma forma mais simples, é como se uma variável tivesse outras variáveis dentro dela.

A vantagem em se usar estruturas de dados é que podemos agrupar de forma organizada vários tipos de dados diferentes, por exemplo, dentro de uma estrutura de dados podemos ter juntos tanto um tipo float, um inteiro, um char ou um double.

As variáveis que ficam dentro da estrutura de dados são chamadas de membros.

```
struct nome
{
    tipo1 variavel1;
    tipo2 variavel2;
    tipo3 variavel3;
    ...
};
```



# STRUCT - REGISTROS

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

```
typedef struct Cpessoa
{
    char nome[20];
    int idade;
};
```

Definição do Registro (struct)

```
int main(void)
{
```

```
    Cpessoa aluno[4]; <- Definição do Vetors de Registro (4 posições)
```

```
    strcpy(aluno[0].nome, "Celso de Melo");
    aluno[0].idade = 22;
```

```
    strcpy(aluno[1].nome, "Augusto Nunes");
    aluno[1].idade = 22;
```

```
    strcpy(aluno[2].nome, "Julia Cleria");
    aluno[2].idade = 23;
```

```
    strcpy(aluno[3].nome, "Lucas Barroso");
    aluno[3].idade = 24;
```

Manipulação

```
    for (int i = 0; i<4 ; i++)
    {
```

```
        printf("%s \t %d \n", aluno[i].nome, aluno[i].idade);
```

```
    }
```

```
    getchar();
```

```
}
```

Impressão

# STRUCT - REGISTROS

## Ponteiro de Struct

Um struct consiste em vários dados agrupados em apenas um. Para acessarmos cada um desses dados, usamos um ponto (.) para indicar que o nome seguinte é o nome do membro.

Exemplo:

```
aluno[1].idade
```

Um ponteiro guarda o endereço de memória que pode ser acessado diretamente.

```
paluno->idade
```

# STRUCT - REGISTROS

```
...  
typedef struct Cpessoa  
{  
    char nome[20];  
    int idade;  
};
```

Criação do Registro (struct)

```
int main(void)  
{
```

```
    Cpessoa aluno[4];  
    Cpessoa *paluno = aluno;  
    strcpy(paluno->nome, "Celso de Melo");  
    paluno->idade = 22;  
    paluno++;
```

Criação do ponteiro para a struct

```
    strcpy(paluno->nome, "Augusto Nunes");  
    paluno->idade = 22;  
    paluno++;
```

Avança uma posição de memória

```
    strcpy(paluno->nome, "Julia Cleria");  
    paluno->idade = 23;  
    paluno++;
```

```
    strcpy(paluno->nome, "Lucas Barroso");  
    paluno->idade = 24;  
    paluno++;
```

Volta para a Posição inicial do ponteiro

```
    paluno = aluno;
```

```
    for (int i = 0; i < 4 ; i++)  
    {  
        printf("%s \t %d \n", paluno->nome, paluno->idade);  
        paluno++;  
    }
```

```
    getchar();
```

```
}
```

# STRUCT - REGISTROS



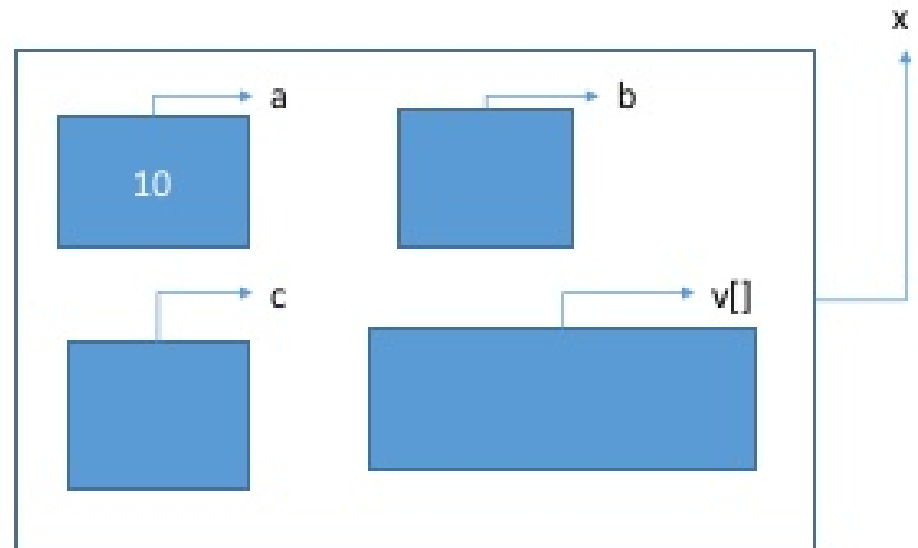
Uma possível representação visual para uma struct seria:

```
struct {  
  
    int a;  
    char b;  
    float c;  
    int v[5];  
  
} x;
```

Se fizermos:

`x.a = 10;`

A caixinha de a  
receberá o  
valor 10!



# Manipulação de Arquivos Binários

```
1  #include <stdio.h>
2  #include <string.h>
3
4  typedef struct Cpessoa
5  {
6      char nome[20];
7      int idade;
8  };
9
10 int main(void)
11 {
12     char condicao = 's';
13     Cpessoa aluno;
14     FILE *bin;
15
16     if((bin = fopen("arquivo_binario.txt", "ab")) == NULL)
17     {
18         printf("Erro ao abrir arquivo");
19     }
20     else
21     {
22         while (condicao == 's' || condicao == 'S')
23         {
24             printf("Informe o nome:");
25             scanf("%s", aluno.nome);
26
27             printf("Informe a idade:");
28             scanf("%d", &aluno.idade);
29
30             printf("Continuar S/N?:");
31
32             fwrite(&aluno, 1, sizeof(aluno), bin);
33
34             fflush(stdin);
35             condicao = getchar();
36         }
37     }
```

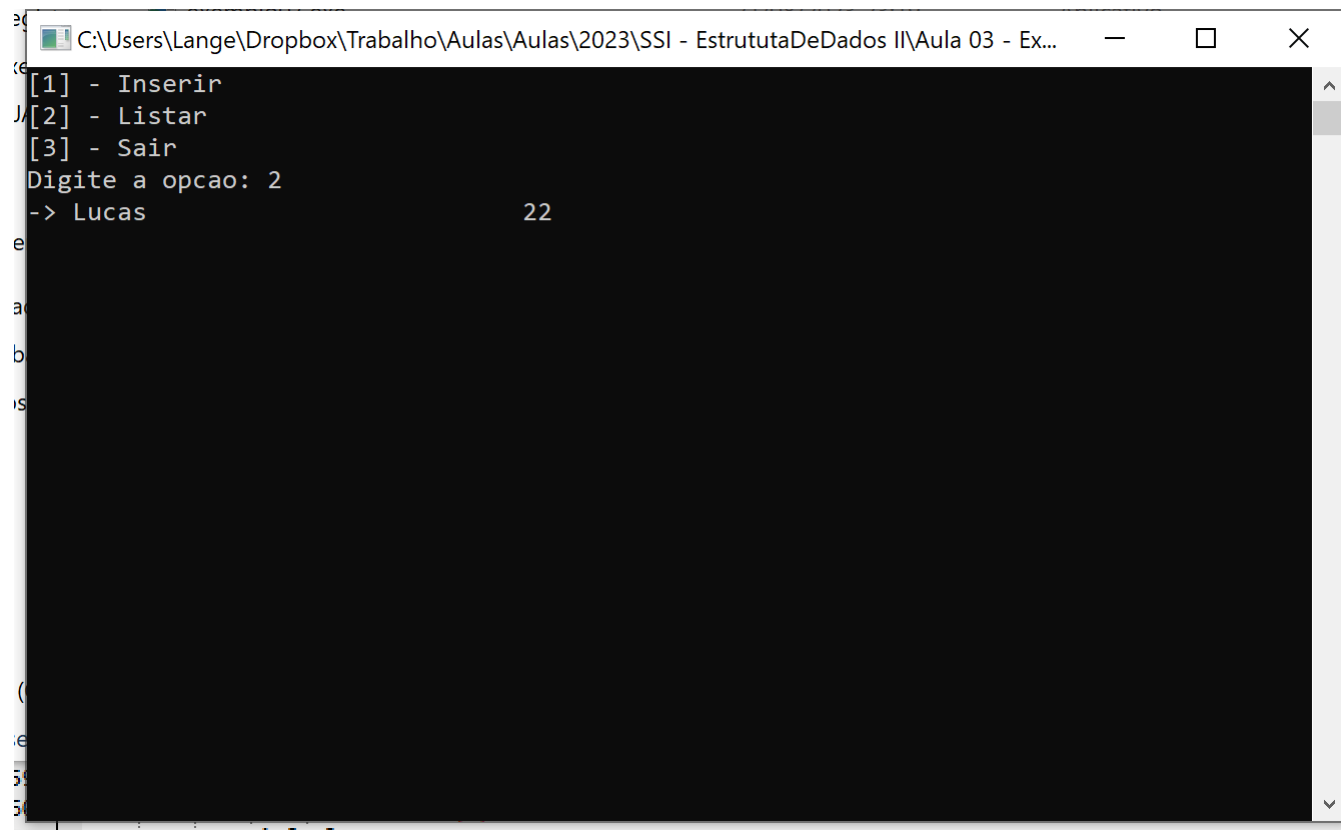
# Manipulação de Arquivos Binários

```
1  #include <stdio.h>
2  #include <string.h>
3
4  typedef struct Cessoa
5  {
6      char nome[20];
7      int idade;
8  };
9
10 int main(void)
11 {
12     char condicao = 's';
13     Cessoa aluno[200];
14     FILE *bin;
15     int tamanho = 0;
16
17     if((bin = fopen("arquivo_binario.txt", "rb")) == NULL)
18     {
19         printf("Erro ao abrir arquivo");
20     }
21     else
22     {
23         //Lendo o Arquivo
24         while (!feof(bin))
25         {
26             fread(&aluno[tamanho], sizeof(Cessoa), 1, bin);
27             tamanho++;
28         }
29
30         //Mostrando na tela
31         for (int i = 0; i < tamanho-1; i++)
32         {
33             printf("%s \t %d \n", aluno[i].nome, aluno[i].idade);
34         }
35         getchar();
36     }
37 }
```



# Manipulação de Arquivos Binários

Implemente um programa em C que manipule os registros e salve em um arquivo binário conforme o programa exemplo.



```
C:\Users\Lange\Dropbox\Trabalho\Aulas\Aulas\2023\SSI - EstruturaDeDados II\Aula 03 - Ex...
[1] - Inserir
[2] - Listar
[3] - Sair
Digite a opcao: 2
-> Lucas                22
```