

Estruturas de dados

Assunto de Hoje

Estruturas Compostas (structs)

Parte II

Professor Luciano Brum

lucianobrum@unipampa.edu.br

Roteiro



- Alocação dinâmica de estruturas.
- Vetores de ponteiros para estruturas.
- Resumo.

Alocação Dinâmica

- Até o momento, vimos como criar estruturas de forma estática.
- Exemplo: struct aluno estudante1;

Alocação Dinâmica

- Deste modo, estamos utilizando o espaço de memória necessário para essa estrutura.
- Ao usar `struct aluno estudantes[100]`, reservamos um espaço de memória para armazenar 100 estudantes, mesmo que não sejam utilizadas, de fato, todas essas estruturas.

Alocação Dinâmica

- Para evitar o desperdício de memória, podemos alocar estruturas de forma dinâmica.
- Como fazemos alocação dinâmica com variáveis de tipos básicos?

Alocação Dinâmica

```
int *variavel;
```

Alocação Dinâmica

```
int *variavel;  
variavel = (int*)malloc(sizeof(int));
```

Alocação Dinâmica

```
int *variavel;  
variavel = (int*)malloc(sizeof(int));  
int x = 10;
```


Alocação Dinâmica

```
int *variavel;  
variavel = (int*)malloc(sizeof(int));  
int x = 10;  
variavel = &x;
```

Alocação Dinâmica

```
int *variavel;  
variavel = (int*)malloc(sizeof(int));  
int x = 10;  
variavel = &x;  
*variável = 20;
```

Alocação Dinâmica

```
int *variavel;  
variavel = (int*)malloc(sizeof(int));  
int x = 10;  
variavel = &x;  
*variável = 20;  
printf("x = %d", x);
```

Alocação Dinâmica

```
int *variavel;  
variavel = (int*)malloc(sizeof(int));  
int x = 10;  
variavel = &x;  
*variável = 20;  
printf("x = %d", x);
```

Quanto vale x??

Alocação Dinâmica

- Como alocar uma struct dinamicamente?

Alocação Dinâmica

- Como alocar uma struct dinamicamente?

```
struct aluno{  
    int mat;  
    char nome[80];  
    char end[120];  
    char tel[20];  
};  
typedef struct aluno Aluno;
```

Alocação Dinâmica

- Como alocar uma struct dinamicamente?
- No main...

Alocação Dinâmica

- Como alocar uma struct dinamicamente?
- No main...

```
Aluno* alunos;  
inicializa(alunos);  
preenche(alunos);  
retira(alunos);
```


Alocação Dinâmica

➤ Procedimento inicializa:

```
void inicializa(Aluno* a){  
    a = NULL;  
}
```

Alocação Dinâmica

➤ Procedimento preenche:

```
void preenche(Aluno* a){  
    if(a!=NULL){  
        printf("Posição já cadastrada.\n");  
        return;  
    }  
    else{  
        ...  
    }  
}
```

Alocação Dinâmica

➤ Procedimento preenche:

```
void preenche(Aluno* a){  
    if(a!=NULL){  
        printf("Posição já cadastrada.\n");  
        return;  
    }  
    else{  
        ...  
    }  
}
```

Alocação Dinâmica

➤ Procedimento preenche:

```
void preenche(Aluno* a){  
    if(a[i]!=NULL){  
        printf("Posição já cadastrada.\n");  
        return;}  
  
    else{  
        a = (Aluno*)malloc(sizeof(Aluno));  
        printf("Matrícula:");  
        scanf("%d", &a->mat);  
        printf("Nome:");  
        scanf(" %80[^\n]", &a->nome);  
        printf("Endereço:");  
        scanf(" %120[^\n]", &a->end);  
        printf("Telefone:");  
        scanf(" %20[^\n]", &a->tel);  
    }  
}
```

Alocação Dinâmica

➤ Procedimento retira:

```
void retira(Aluno* a){  
    if(a!=NULL){  
        free(a);  
        a=NULL;  
    }  
}
```

Alocação Dinâmica

➤ Procedimento retira:

```
void retira(Aluno* a){  
    if(a!=NULL){  
        free(a);  
        a=NULL;  
    }  
}
```

Roteiro



- Alocação dinâmica de estruturas.
- Vetores de ponteiros para estruturas.
- Resumo.

Vetor de Ponteiros para Structs

- E se desejássemos cadastrar 100 alunos?
- E se o endereço fosse uma estrutura?

Vetor de Ponteiros para Structs

- E se desejássemos cadastrar 100 alunos?
- E se o endereço fosse uma estrutura?

```
struct end{  
    char rua[50];  
    int num;  
};  
typedef struct end End;  
struct aluno{  
    int mat;  
    char nome[80];  
    End endereco;  
    char tel[20];  
};  
typedef struct aluno Aluno;
```

Vetor de Ponteiros para Structs

- Quais alterações no inicializa?
- Quais alterações no preenche?
- Quais alterações no retira?

Vetor de Ponteiros para Structs

➤ No main:

```
int tam = 100;  
Aluno* alunos[tam];  
inicializa(tam,alunos);  
preenche(tam,alunos,pos);  
retira(tam,alunos,pos);
```

Vetor de Ponteiros para Structs

➤ Procedimento inicializa:

```
void inicializa(int tam, Aluno** a){  
    int i;  
    for(i=0;i<tam;i++){  
        a[i] = NULL;  
    }  
}
```

Vetor de Ponteiros para Structs

➤ Procedimento preenche:

```
void preenche(int n, Aluno** a, int i){  
    if(i<0 || i>n){  
        printf("Índice fora dos limites do vetor.\n");  
        exit(1);  
    }  
    if(a[i]!=NULL){  
        printf("Posição já cadastrada.\n");  
        return;  
    }  
    else {  
        a[i] = (Aluno*) malloc(sizeof(Aluno));  
        printf(" Matrícula:");  
        scanf(" %d", &a[i]->mat);  
        printf(" Nome:");  
        scanf(" %80[^\n]", &a[i]->nome);  
        printf(" Rua:");  
        scanf(" %50[^\n]", &a[i]->endereco.rua);  
        printf(" Numero:");  
        scanf(" %d", &a[i]->endereco.num);  
        printf(" Telefone:");  
        scanf(" %20[^\n]", &a[i]->tel);  
    }
```

Vetor de Ponteiros para Structs

➤ Procedimento preenche:

```
void preenche(int n, Aluno** a, int i){
    if(i<0 || i>n){
        printf("Índice fora dos limites do vetor.\n");
        exit(1);}
    if(a[i]!=NULL){
        printf("Posição já cadastrada.\n");
        return;}

    else{
        a[i] = (Aluno*)malloc(sizeof(Aluno));
        printf("Matrícula:");
        scanf("%d", &a[i]->mat);
        printf("Nome:");
        scanf(" %80[^\n]", &a[i]->nome);
        printf("Rua:");
        scanf(" %50[^\n]", &a[i]->endereco.rua);
        printf("Numero:");
        scanf("%d", &a[i]->endereco.num);
        printf("Telefone:");
        scanf(" %20[^\n]", &a[i]->tel);} }
```

Vetor de Ponteiros para Structs

➤ Procedimento retira:

```
void retira(int n, Aluno** a, int i){  
    if(i<0 || i>n){  
        printf("Índice fora dos limites do vetor.\n");  
        exit(1);  
    }  
    if(a[i]!=NULL){  
        free(a[i]);  
        a[i]=NULL;  
    }  
}
```

Vetor de Ponteiros para Structs

- E se o endereço fosse um ponteiro para a estrutura?

```
struct end{
    char rua[50];
    int num;
};
typedef struct end End;
struct aluno{
    int mat;
    char nome[80];
    End* endereco;
    char tel[20];
};
typedef struct aluno Aluno;
```


Vetor de Ponteiros para Structs

➤ Procedimento inicializa:

```
void inicializa(int n, Aluno** a){  
    int i;  
    for(i=0;i<n;i++){  
        a[i] = NULL;  
    }  
}
```

Vetor de Ponteiros para Structs

➤ Procedimento preenche:

```
void preenche(int n, Aluno** a, int i){
    if(i<0 || i>n){
        printf("Índice fora dos limites do vetor.\n");
        exit(1);}
    if(a[i]!=NULL){
        printf("Posição já cadastrada.\n");
        return;}

    else{
        a[i] = (Aluno*)malloc(sizeof(Aluno));
        a[i]->endereco = (End*)malloc(sizeof(End));
        printf("Matrícula:");
        scanf("%d", &a[i]->mat);
        printf("Nome:");
        scanf(" %80[^\n]", &a[i]->nome);
        printf("Rua:");
        scanf(" %50[^\n]", &a[i]->endereco->rua);
        printf("Numero:");
        scanf("%d", &a[i]->endereco->num);
        printf("Telefone:");
        scanf(" %20[^\n]", &a[i]->tel);}
```

Vetor de Ponteiros para Structs

➤ Procedimento retira:

```
void retira(int n, Aluno** a, int i){  
    if(i<0 || i>n){  
        printf("Índice fora dos limites do vetor.\n");  
        exit(1);  
    }  
    if(a[i]!=NULL){  
        free(a[i]->endereco);  
        free(a[i]);  
        a[i]=NULL;  
    }  
}
```

Roteiro



- Alocação dinâmica de estruturas.
- Vetores de ponteiros para estruturas.
- Resumo.

Resumo

- Vimos o que é uma estrutura (struct) e como podemos construir objetos com atributos heterogêneos na linguagem C.
- Vimos como utilizar o modificador de tipos typedef.
- Vimos como alocar memória dinamicamente para vetores de e para structs.

Dúvidas ?