

Aula 5: Structs



DCC302-Estrutura de Dados I
Prof. Me. Acauan C. Ribeiro

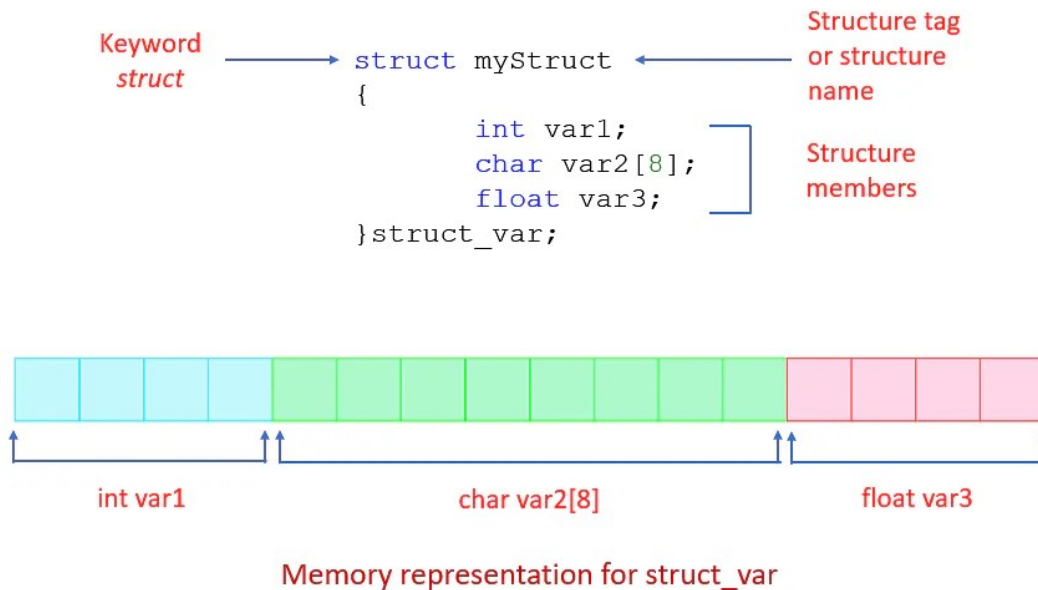
Structs

- Uma **struct** (ou **registro** ou **record**) é um "pacote" de variáveis, que podem ter tipos diferentes;
- Visa representar grupos de dados que resultem em algo mais “concreto”:
 - P. ex.: registro de alunos, automóveis, etc...
- Cada variável é um **campo** do registro;
- Em C, registros são conhecidos como **structs** (que vem de structure, do inglês);

Structs

Uso de Estruturas de Dados Heterogêneas

- A ideia de usar uma **struct** é permitir que, ao armazenar os dados de uma mesma entidade, isto possa ser feito com uma **única variável**. Por exemplo, se for preciso armazenar a altura, o peso e a idade de uma pessoa, pode-se criar uma **struct** chamada **Pessoa** e agrupar os dados em um único tipo de dado.
- Aos dados agrupados em uma **struct** dá-se o nome de **campos**(fields).



Struct (Como declarar?)

Definição:

→ FORMAS PARA **NÃO USAR** (mas temos que saber que existem)

```
struct [nome do registro]{  
    tipo campo1;  
    tipo campo2;  
    ...  
    tipo campoN;  
} [uma ou mais variáveis];
```

- *[nome do registro]* e *[uma ou mais variáveis]* são **opcionais**;
- *[nome do registro]* é simplesmente o nome (não um tipo) da classe de registros do mesmo tipo
- Você já pode **declarar variáveis**, referentes a esta *struct*, colocando-as no lugar de *[uma ou mais variáveis]*;

Struct (Como declarar?)

1ª opção

→ FORMAS PARA NÃO USAR (mas temos que saber que existem)

- Vamos declarar uma *struct* para armazenar **alunos**;

Opção 1: Declara a struct Aluno

```
struct Aluno {  
    char nome[100];  
    int idade;  
};  
  
// declaração de uma variável  
struct Aluno barney;
```

Struct (Como declarar?)

2ª opção

→ FORMAS PARA NÃO USAR (mas temos que saber que existem)

- Vamos declarar uma *struct* para armazenar **alunos**;

Opção 2: Define a struct Aluno e já declara variáveis deste tipo;

```
struct Aluno {  
    char nome[100];  
    int idade;  
} barney, ted;
```

Struct (Como declarar?)

3ª opção

→ FORMAS PARA NÃO USAR (mas temos que saber que existem)

- Vamos declarar uma *struct* para armazenar **alunos**;

Opção 3: Define a struct sem nome e declara variáveis deste tipo;

Aqui temos um problema →

```
struct {  
    char nome[100];  
    int idade;  
} barney, ted;
```

Struct (Como declarar?)

3ª opção

→ FORMAS PARA NÃO USAR (mas temos que saber que existem)

- Vamos declarar uma *struct* para armazenar **alunos**;

Opção 3: Define a struct sem nome e declara variáveis deste tipo;

Aqui temos um problema →

- Por não ter um nome, não conseguimos referenciar a struct dentro dela mesma (isso vai ser importante)
- Não conseguimos declarar outras structs somente "barney" e "ted"

```
struct {  
    char nome[100];  
    int idade;  
} barney, ted;
```


Struct (Como declarar?)

4ª opção

→ INDICO UTILIZAR DESTA FORMA!!



Define um tipo novo

```
typedef struct aluno {  
    char nome[64];  
    int idade;  
    float media;  
} Aluno;
```

Esse é o tipo

Podemos declarar

```
int main() {  
    Aluno a1;
```

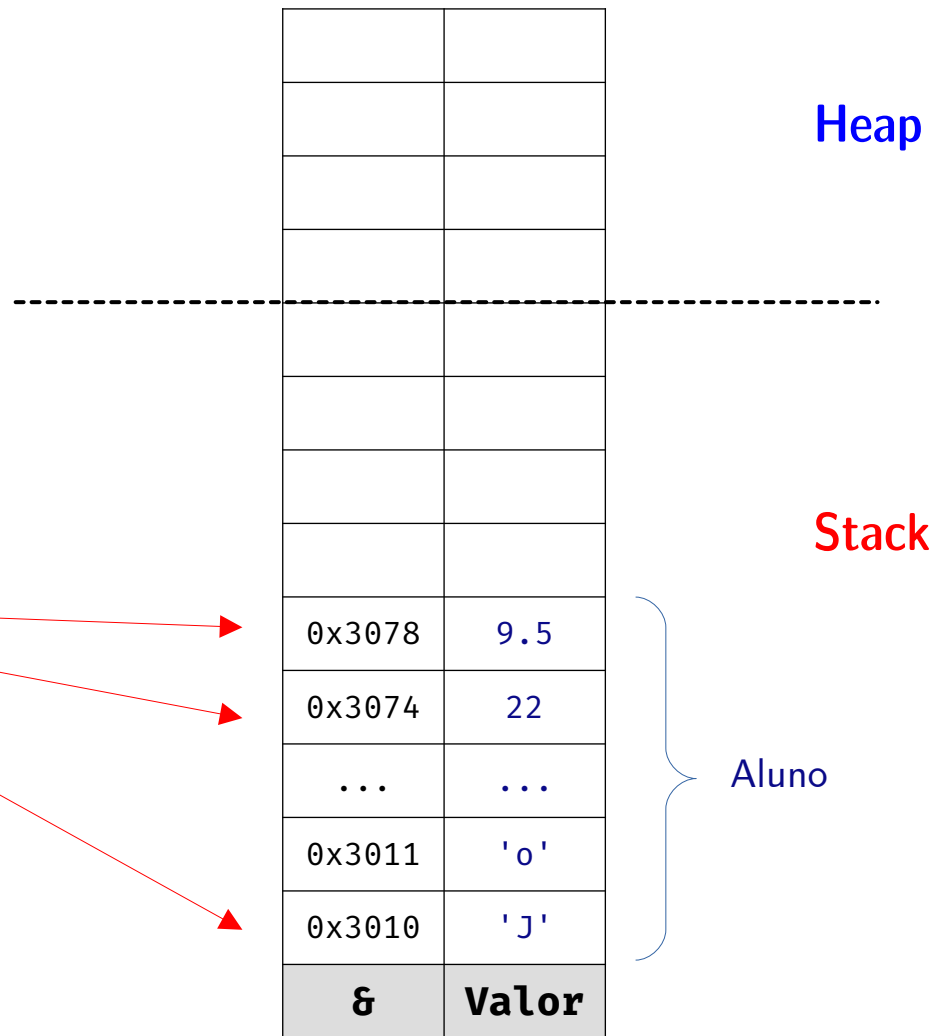
Structs

Let's code...



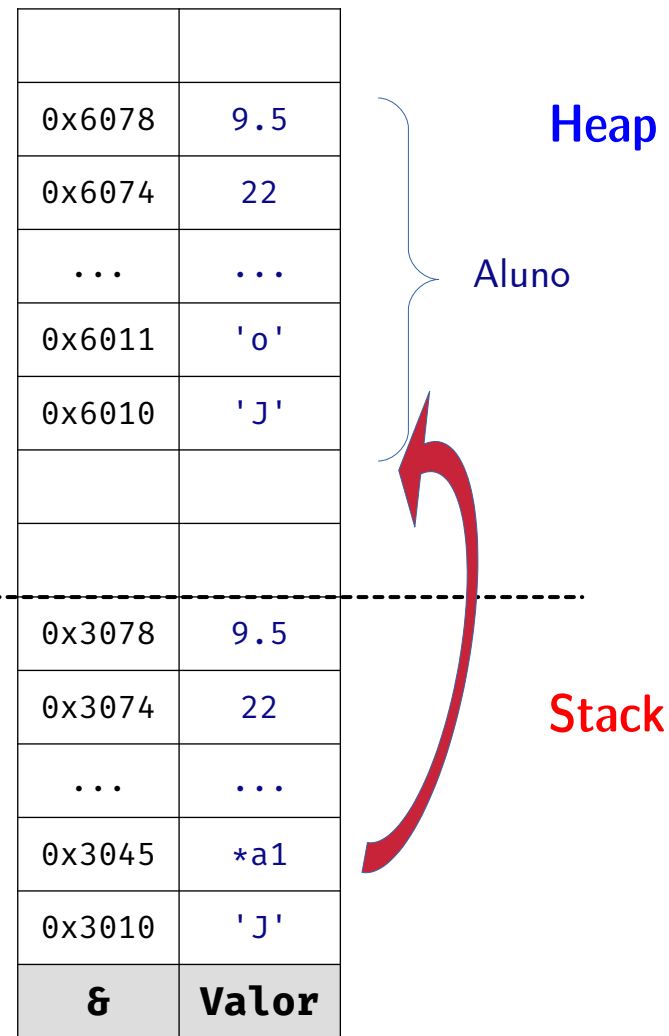
Structs: Alocação Estática

```
1  #include <stdio.h>
2  #include <string.h>
3
4  typedef struct aluno {
5      char nome[64];
6      int idade;
7      float media;
8  } Aluno;
9
10 int main() {
11     Aluno al;
12     strcpy(al.nome, "João");
13     al.idade = 22;
14     al.media = 9.5;
15
16     puts("--- Dados aluno ---");
17     printf("Nome: %s\n", al.nome);
18     printf("Idade: %d\n", al.idade);
19     printf("Média: %.2f\n\n", al.media);
20
21     puts("--- Tamanho tipo de dado 'Aluno' ---");
22     printf("size(Aluno): %lu\n", sizeof(Aluno));
23     printf("&al: %p\n", &al);
24     printf("&al.nome[0]: %p\n", &al.nome[0]);
25
26     return 0;
27 }
```



Structs: Alocação Dinâmica

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  typedef struct aluno {
6      char nome[64];
7      int idade;
8      float media;
9  } Aluno;
10
11 int main() {
12     Aluno *a1 = malloc(1*sizeof(Aluno));
13     strcpy(a1->nome, "João");
14     a1->idade = 22;
15     a1->media = 9.5;
16
17     puts("--- Dados aluno ---");
18     printf("Nome: %s\n", a1->nome);
19     printf("Idade: %d\n", a1->idade);
20     printf("Média: %.2f\n\n", a1->media);
21
22     puts("--- Tamanho tipo de dado 'Aluno' ---");
23     printf("size(Aluno): %lu\n", sizeof(Aluno));
24     printf("&a1: %p\n", &a1);
25     printf("&a1->nome[0]: %p\n", &a1->nome[0]);
26
27     return 0;
28 }
```



Structs

Let's code...

A game



Referências

- <https://www.inf.pucrs.br/~pinho/Laprol/Structs/Structs.htm>
- <https://www.simplilearn.com/tutorials/c-tutorial/structure-in-c>