

# **Aula 12: Laboratório - Estruturas (parte 1)**

Prof. João Henrique Kleinschmidt

Parte do material elaborado pelo Prof. Jesús P. Mena-Chalco

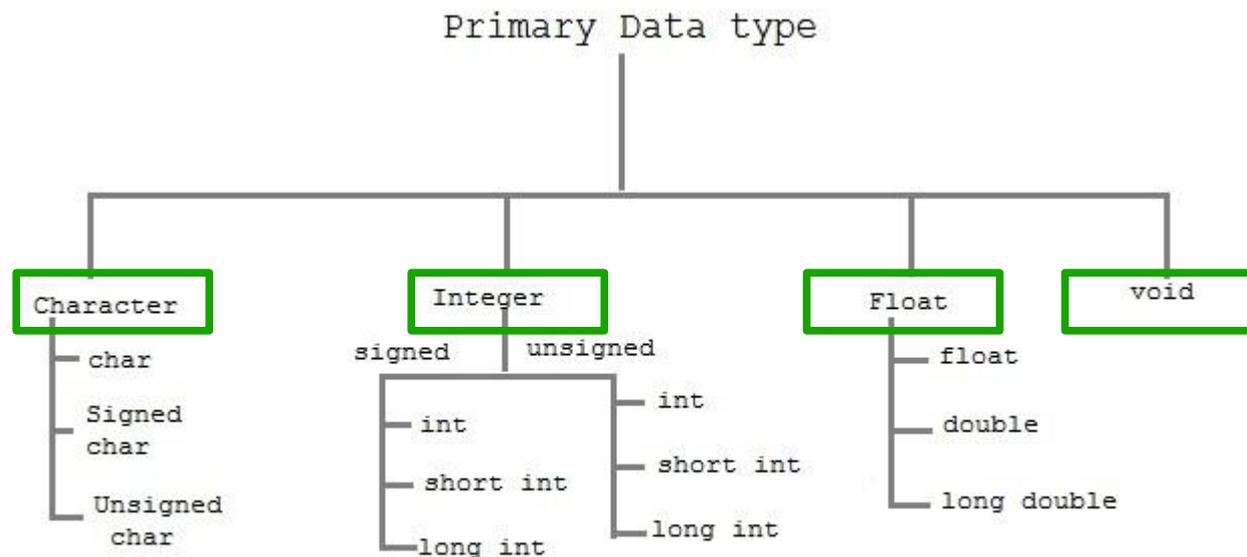
3Q-2018



# **Estruturas (=registros)**

# Linguagem C: Tipos de dados

- **Tipos de dados primários.**
- Tipos de dados derivados.
- Tipos definidos pelo usuário.



# Estruturas

```
1 #include<stdio.h>
2
3 int main()
4 {
5     struct ponto3D {
6         double x;
7         double y;
8         double z;
9     };
10
11     struct ponto3D p1;    /*um registro p1 do tipo ponto3D*/
12
13     printf("%ld\n", sizeof(p1));
14     printf("%f %f %f\n", p1.x, p1.y, p1.z);
15
16 }
```

# Estruturas

```
1 #include<stdio.h>
2
3 int main()
4 {
5     struct ponto3D {
6         double x;
7         double y;
8         double z;
9     };
10
11     struct ponto3D p1;    /*um registro p1 do tipo ponto3D*/
12
13     printf("%ld\n", sizeof(p1));
14     printf("%f %f %f\n", p1.x, p1.y, p1.z);
15
16 }
```

24

0.000000 0.000000 0.000000

# Estruturas

```
1 #include <stdio.h>
2
3 struct ponto3D {
4     double x;
5     double y;
6     double z;
7 };
8
9 int main() {
10     struct ponto3D *ptr; /* ptr é um ponteiro para registros ponto3D */
11     struct ponto3D p1;
12
13     ptr = &p1; /* ptr aponta a p1 */
14
15     (*ptr).x = 10; /* mesmo efeito que p1.x=10 */
16
17     ptr->x = 10; /* ptr->x é uma abreviatura (*ptr).x */
18 }
```

```
1 #include <stdio.h>
2 #include <math.h>
3
4 struct ponto3D {
5     double x;
6     double y;
7     double z;
8 };
9
10 double dEuclidiana(struct ponto3D p, struct ponto3D q) {
11     return sqrt(pow(p.x-q.x,2) + pow(p.y-q.y,2) + pow(p.z-q.z,2));
12 }
13
14 int main() {
15     struct ponto3D p1, p2;
16
17     p1.x = p1.y = p1.z = 0;
18     p2.x = p2.y = p2.z = 10;
19
20     printf("%f\n", dEuclidiana(p1,p2) );
21 }
```

```
/tmp/cciEjaWo.o: In function `dEuclidiana':
estrutura3.c:(.text+0x2a): undefined reference to `pow'
estrutura3.c:(.text+0x55): undefined reference to `pow'
estrutura3.c:(.text+0x89): undefined reference to `pow'
estrutura3.c:(.text+0x93): undefined reference to `sqrt'
collect2: error: ld returned 1 exit status
```

```
$ gcc distanciaE.c -lm -o distanciaE.exe
$ ./distanciaE.exe
17.320508
```

gcc: why the -lm flag is needed to link the math library? [duplicate]



▲  
36  
▼

This question already has an answer here:  
[Why do you have to link the math library in C?](#) 9 answers

# leituraPontos.c

```
1 #include <stdio.h>
2 #include <math.h>
3
4 struct ponto3D {
5     double x;
6     double y;
7     double z;
8 };
9
10
11 int main() {
12     int i, n;
13
14     scanf("%d", &n);
15     struct ponto3D v[n] ;
16
17     // Leitura dos pontos 3D
18     for (i=0; i<n; i++)
19         scanf("%lf %lf %lf", &(v[i].x), &(v[i].y), &(v[i].z) );
20
21     // Imprimimos os pontos 3D
22     for (i=0; i<n; i++)
23         printf("\n%lf %lf %lf", v[i].x, v[i].y, v[i].z );
24 }
```

```
$ gcc leituraPontos.c -o leituraPontos.exe
$ ./leituraPontos.exe < vetorDePontos3D.txt
```

# Exercício 1

Modifique o programa anterior para imprimir a maior distância Euclidiana entre quaisquer par de pontos 3d.

# Exercício 2

- Defina uma estrutura chamada *horario*, composta por hora, minuto e segundo.
- Crie uma função chamada *diferenca\_de\_horarios(struct horario inicio, struct horario fim)* para retornar a diferença entre dois horários (em segundos), com base na estrutura definida no exercício anterior.
- Por exemplo, para início = 20:05:10 (20 horas, 5 minutos e 10 segundos) e fim = 20:35:15, o resultado é: 1805 (segundos).

# Exercício 3

Crie uma estrutura representando os alunos de um determinado curso. A estrutura deve conter o RA do aluno, nome, nota da primeira prova, nota da segunda prova e nota da terceira prova.

- (a) Permita ao usuário entrar com os dados de 5 alunos.
- (b) Encontre o aluno com maior nota da primeira prova.
- (c) Encontre o aluno com maior média geral.
- (d) Encontre o aluno com menor média geral
- (e) Para cada aluno imprima se ele foi aprovado ou reprovado, considerando o valor 7 para aprovação.

Exemplo de leitura de string:

```
1  #include <stdio.h>
2
3  void main() {
4      char str[80];
5
6      printf("Digite um nome:");
7      scanf("%s", str);
8      printf("Você digitou: %s", str);
9  }
```

# Exercício 4

- Faça um programa que armazene em uma estrutura os dados de um funcionário de uma empresa, compostos de:
  - Nome (string de até 99 caracteres)
  - Idade
  - Sexo (M/F)
  - CPF
  - Setor onde trabalha (string de até 99 caracteres)
  - Cargo que ocupa (string de até 30 caracteres)
  - Salário
- Os dados devem ser digitados pelo usuário, armazenados na estrutura e exibidos na tela.

# maiorDistancia.c

```
1 #include <stdio.h>
2 #include <math.h>
3
4 struct ponto3D {
5     double x;
6     double y;
7     double z;
8 };
9
10 double dEuclidiana(struct ponto3D p, struct ponto3D q) {
11     return sqrt(pow(p.x-q.x,2) + pow(p.y-q.y,2) + pow(p.z-q.z,2));
12 }
13
14
15 int main() {
16     int i, j, n;
17     double maiorD = 0;
18
19     scanf("%d", &n);
20     struct ponto3D v[n] ;
21
22     // Leitura dos pontos
23     for (i=0; i<n; i++)
24         scanf("%lf %lf %lf", &(v[i].x), &(v[i].y), &(v[i].z) );
25
26     // Busca pela maior distancia entre quaisquer 2 pontos 3D
27     for (i=0; i<n-1; i++)
28         for (j=i+1; j<n; j++)
29             if ( maiorD < dEuclidiana(v[i], v[j]) )
30                 maiorD = dEuclidiana(v[i], v[j]);
31
32     printf("%lf", maiorD);
33 }
```