Resumo: Structs em C (Registros)

Este resumo foi feito para estudantes que nunca tiveram contato com o conceito de 'structs' em C, também conhecidas como registros. O conteúdo a seguir explica de forma simples e com exemplos claros.

# O que são structs?

Structs (ou registros) são tipos de variáveis compostas heterogêneas, ou seja, capazes de armazenar vários dados de tipos diferentes dentro de uma mesma estrutura. Elas servem para representar melhor objetos do mundo real, como uma data ou os dados de uma pessoa.

# Exemplo básico de struct

Vamos imaginar que queremos guardar uma data. Podemos usar uma struct assim:

struct {  
 int dia;  
 int mes;  
 int ano;  
} data;  
  
data.dia = 31;  
data.mes = 12;  
data.ano = 2024;

# Criando um tipo com nome

É uma boa prática nomear structs, o que facilita seu uso repetido:

struct Data {  
 int dia;  
 int mes;  
 int ano;  
};  
  
struct Data hoje;

# Simplificando com typedef

Podemos usar typedef para criar um nome mais simples para a struct:

typedef struct {  
 int dia, mes, ano;  
} Data;  
  
Data hoje, amanha;

# Campos de diferentes tipos

Os campos de uma struct podem ser de tipos variados. Por exemplo, para guardar dados de uma pessoa:

typedef struct {  
 char nome[30];  
 int idade;  
} Pessoa;

# Structs dentro de structs

Uma struct pode conter outra struct como campo. Veja um exemplo de uma venda com data:

typedef struct {  
 int dia, mes, ano;  
} Data;  
  
typedef struct {  
 char nome\_peca[50];  
 float preco;  
 Data dataVenda;  
} Venda;

# Vetores de structs

Também é possível criar vetores (listas) de structs. Exemplo:

Venda vendas[50];  
  
strcpy(vendas[0].nome\_peca, "Parafuso");  
vendas[0].preco = 5.50;  
vendas[0].dataVenda.dia = 10;

# Conclusão

Structs são muito úteis para organizar dados complexos. Elas tornam o código mais claro, organizado e mais próximo de como pensamos no mundo real. Entender e usar structs é essencial para quem está aprendendo C.