

Tipos de Dados Definidos pelo Usuário

3. Uniões

Uma declaração `union` determina uma única localização de memória onde podem estar armazenadas várias variáveis de tipos diferentes em momentos distintos. A declaração de uma união é semelhante à declaração de uma estrutura:

```
union nome_do_tipo_da_union  
{  
    tipo_1 nome_1;  
    tipo_2 nome_2;  
    ...  
    tipo_n nome_n;  
} variaveis_union;
```

Tipos de Dados Definidos pelo Usuário

3. Uniões (continuação)

Exemplo:

```
union inteiro_real  
{  
    short int A;  
    float B;  
} x;  
  
...  
x.A=4;  
x.B=3.14;
```

Tipos de Dados Definidos pelo Usuário

Exercício:

Com base no programa construído no exercício do slide 341, implemente uma função para consultar a presença de um determinado registro com base no valor do campo especificado, fazendo as alterações que julgar necessárias no código do referido programa, desde que em concordância com as restrições especificadas no exercício mencionado.

```
enum campos
{
    nome, data,
    matricula,
    CPF, coeficiente
};
union campos_registro
{
    char nom[30];
    char dat[9];
    unsigned long int n_mat;
    char cpf[13];
    float cr;
};
```

```

int consulta_reg_valor (struct registro *v, enum campos campo,
union campos_registro chave)
{
    int i;
    switch (campo)
    {
        case nome:
            for (i=0; i<num_reg; i++)
                if(!strcmp(v[i].nom,chave.nom))
                    return i+1;
        case data:
            for (i=0; i<num_reg; i++)
                if(!strcmp(v[i].dat,chave.dat))
                    return i+1;
        case matricula:
            for (i=0; i<num_reg; i++)
                if(v[i].n_mat==chave.n_mat)
                    return i+1;
    }
}

```

```
case CPF:
    for (i=0; i<num_reg; i++)
        if(!strcmp(v[i].cpf,chave.cpf))
            return i+1;
case coeficiente:
    for (i=0; i<num_reg; i++)
        if(v[i].cr==chave.cr)
            return i+1;
default:
    printf("\n\nOpcao invalida!\n");
}
return 0;
}
```

Tipos de Dados Definidos pelo Usuário

4. Definição de tipo

O comando `typedef` é utilizado para definir um novo tipo de dado. Ele é utilizado da seguinte forma:

```
typedef tipo nome_do_tipo;
```

Tipos de Dados Definidos pelo Usuário

Exemplo:

```
typedef enum  
{seg,ter,qua,qui,sex,sab,dom} dias_semana;
```

```
typedef struct  
{  
    dias_semana dia;  
    int hora;  
    int minuto;  
} data;
```

Tipos de Dados Definidos pelo Usuário

Exercício:

Construa um programa que defina um novo tipo de dado, que é um união do tipo caractere com o tipo inteiro. O programa deve ler através da entrada padrão um número inteiro, e de acordo com a possibilidade de representação de cada um dos tipos envolvidos na união, armazená-lo no tipo adequado.