

Estruturas

Uma estrutura no IDL, são diversas variáveis com tipos diferentes. Existem dois tipos de variáveis de estrutura no IDL: estruturas nomeadas e estruturas anônimas.

Estruturas Nomeadas

Uma estrutura nomeada é a estrutura que armazena o layout da estrutura sob um único nome no IDL. Os componentes de uma estrutura nomeada são o nome, os campos e os valores. As chaves são usadas exclusivamente para criação das estruturas.

```
IDL> p1 = { point, x: 1.0, y: 3.0, color: bytarr(3) }
```

A instrução define o nome da estrutura, *point*, este contém três campos, *x*, *y*, e *color*. A estrutura é armazenada na variável *p1*. O tipo de variável dos campos de uma estrutura é determinado pelo valor utilizado no campo. No caso da estrutura *point*, os campos *x* e *y* são flutuantes e o *color* é um vetor de bytes. Para ter mais informações sobre a variáveis da estrutura, use a palavra-chave *STRUCTURE* para a rotina *HELP*.

```
IDL> help, p1, /structure
** Structure POINT, 3 tags, length=12, data length=11:
X          FLOAT          1.00000
Y          FLOAT          3.00000
COLOR BYTE   Array[3]
```

Os campos de uma estrutura podem ser inicializados por variáveis. O tipo da variável determinará o tipo dos campos. Observe o exemplo abaixo.

```
IDL> a = 0UL
IDL> b = 0.9
IDL> c = 23.4D
IDL> my1 = { mystructure, f1: a, f2: b, f3: c }
```

A estrutura nomeada *mystruct* contém três tipos de campos: uma *unsigned long*, uma *float* e uma *double*.

Uma vez que uma estrutura nomeada esteja definida no IDL, os campos não podem mudar o tamanho ou o tipo, nem podem ser adicionados ou removido da estrutura. As definições de uma estrutura nomeada ficam armazenadas na memória do IDL até que a sessão seja encerrada ou reiniciada.

Para criar um novo exemplo da estrutura nomeada, para atribuir a estrutura para uma variável, use chaves e entre elas insira o nome das estruturas nomeadas.

```
IDL> p2 = { point }
IDL> my2 = { mystruct }
```

A variável *p2* contém uma cópia fiel da estrutura *point*. Quando uma nova estrutura nomeada é criada desta maneira, todos os campos serão zero ou nulos (*strings*, *pointers*, *objects*), os valores utilizados para definir a estrutura não serão carregados.

```
IDL> help, p2, /structure
** Structure POINT, 3 tags, length=12, data length=11:
```

```
X      FLOAT    0.000000
Y      FLOAT    0.000000
COLOR  BYTE     Array[3]
```

Acessando campos das estruturas

Para acessar um campo de uma estrutura, use o nome da estrutura com o nome do campo,

```
IDL> print, p2.x
      0.000000
IDL> p2.y = 9.5
IDL> p2.color = [255, 0, 255]
IDL> my2.f1 = 44L
```

Os campos das estruturas podem ser acessados por números.

```
IDL> print, p2. (0)                ; da mesma forma que p2.x
      0.000000
IDL> p2. (1) = 9.5                  ; da mesma forma que p2.y
IDL> p2. (2) = [255, 0, 255]       ; da mesma forma que p2.color
IDL> my2.(0) = 44L                  ; da mesma forma que my2.f1
```

Estruturas Anônimas

Estruturas anônimas não tem um nome associado com elas.

```
IDL> an1 = { lat : 0.0, lon: 0.0, ht: 0L, pitch: 0.0 }
IDL> an1.lat = 12.4
IDL> an1.pitch = 45.67
```

Veja que não tem nenhum nome associado a ela, uma nova estrutura anônima pode ser definida adicionando ou removendo campos.

```
IDL> an1 = { lat : 0.0, lon: 0.0, ht: 0L, pitch: 0.0, attitude: 0U }
IDL> an2 = an1
IDL> an2.pitch = 79.0D
```

Trabalhando com Estruturas

Matrizes ou Vetores de Estruturas

Para criar uma matriz ou vetor de uma estrutura use a função *REPLICATE*. Esta função pode ser usada para copiar dados de qualquer tipo mas é usado freqüentemente para replicar matrizes ou vetores de estrutura. A matriz ou vetor criada pode ser replicada pelo nome da estrutura ou pelas suas variáveis.

```
IDL> ptarr = replicate ( { point }, 5 )
IDL> ptarr[0].x = 5.6
IDL> ptarr[0].y = 9.3
IDL> print, ptarr[0].x
      5.60000
```

```
IDL> print, ptarr[1].x
0.000000
```

Cada campo da matriz ou vetor de estrutura pode ser tratado como se fosse uma estrutura normal.

```
IDL> ptarr.x = 6.7
IDL> print, ptarr[0].x
6.70000
IDL> print, ptarr[1].x
6.700000
```

A instrução acima fez com que todas as posições do vetor de estruturas *ptarr* do campo *x* passasse a ser 6.7.

```
IDL> ptarr.y = findgen ( 5 )
IDL> print, ptarr.y
0.000000  1.00000  2.00000  3.00000  4.00000
```

A instrução acima mostra uma maneira de inicializar um vetor ou matriz com uma seqüência de números que começa em 0.000000.

Tipos e Tamanhos de Campos de uma Estrutura

Quando uma estrutura anônima ou nomeada é criada, o tipo e tamanho dos campos são definidos e não podem mais ser alterados. Quando você insere um valor no campo da estrutura, o IDL converte o valor ou variável para o mesmo tipo do campo.

```
IDL> anon = { nome : 'João', idade : 34, altura : 1.70, categoria : fltarr(10) }
IDL> anon.idade = 45.5 ; o valor é convertido para integer
IDL> anon.altura = 70 ; o valor é convertido para float
IDL> anon.categoria[9] = 78.54D ; o valor é convertido para float
IDL> anon.categoria = fltarr (8) ; tudo bem mas o campo ainda contém 10 valores
IDL> anon.categoria = intarr (10) ; tudo bem mas o campo ainda é fltarr
IDL> help, anon, /structure
** Structure <10e8918>, 4 tags, length=60, data length=58, refs=1:
NOME          STRING 'João'
IDADE         INT          45
ALTURA       FLOAT       70.0000
CATEGORIA     FLOAT       Array[10]
IDL> anon.categoria = fltarr (15) ; ERRO! Na tentativa de redimensionar o campo
```

O IDL é muito flexível com conversões, mas quando tratamos de um campo de matriz ou vetor ele não pode ser expandido.

Funções Variadas de uma Estrutura

Os nomes dos campos de uma estrutura podem ser retornados na forma de um vetor de *strings* usando a função *TAG_NAMES*.

```
IDL> fnames = tag_names (p2)
IDL> print, fnames
X Y COLOR
```

A função *TAG_NAMES* pode ser usada também para retornar o nome da estrutura.

```
IDL> sname = tag_names (p2, /structure_name)
IDL> print, sname
      POINT
```

Para saber o número de campos em uma estrutura use a função *N_TAGS*.

```
IDL> ncampos = n_tags(p2)
IDL> print, ncampos
      3
```

A função *N_TAGS* pode também devolver o tamanho em bytes de uma estrutura.

```
IDL> slen = n_tags (p2, /length)
IDL> print, slen
      12
```