IDL Tutorial

AULA 10

COMMON
ARQUIVOS
LEITURA
ESCRITA

Antonio Paulo V. Pinto - Eng. Elétrica - UFRN

Revisão

- Funções
  - Realiza um cálculo matemática e retorna um valor. Sua chamada pode ser utilizada dentro de equações.
- Sub-rotinas ou Procedimentos
  - Algoritmo completo, realiza alguma ação e pode ter diversas saídas. Sua chamada deve ser em linha de comando própria.
- Scripts
  - Algoritmo de comandos de linha, funciona como Ctrl+C e Ctrl+V.

Revisão de definições.

- Funções:

```
FUNCTION nome, in1, in2
    out = in1 + in2
    RETURN, out
END
```

- Sub-Rotinas

```
PRO nome, var1, var2
    var2 = var1^2
END
```

- Scripts
  - Não há estrutura básica, nem variáveis de entrada, nem de saída.

Chamadas das funções

- Funções
  - Y = funcao(in1, in2)
- Sub-Rotinas
  - subrotina, in1, out, in2
- Scripts
  - @script

Compartilhamento de Variáveis

- Bloco de variáveis que podem ser compartilhadas em diversas funções.
- Finalidade: Criar um conjunto de variáveis quase-globais.
- Sub-Rotina: **COMMON**
- Definição: **COMMON**, nome, var1, var2, varN

Exemplo

```
; Arquivo: Teste.pro

PRO teste2

  COMMON buga, a, b
  a=[5,2]
  b=[6,3]

END

PRO teste
  teste2
  COMMON buga, x, y
  PLOT, x, y

END
```

Utilidade

- Criar um conjunto grande de variáveis que serão passadas para as sub-rotinas ou funções e podem ser alteradas nelas.
- Evitar a criação de uma quantidade imensa de variáveis de entrada.
- Compartilhar um conjunto de Constantes do seu algoritmo.

Arquivos e Banco de Dados

- Banco de dados: Arquivo onde serão armazenados uma quantidade imensa de informações, como por exemplo, inúmeros valores de variáveis, ou dados.

- Tipos de banco de dados: SQL, MySQL, ACCESS...
  - Suportado naturalmente pelo IDL: SQL

- Arquivo '.txt'
  - Arquivo de texto onde podemos armazenar qualquer tipo de informação. Pode ser usado como um banco de dados rudimentar.

Arquivos

- OpenR
  - Abre um arquivo somente para leitura, dessa forma você estará protegido contra fazer alterações indesejadas no arquivo.
- OpenW
  - Abre o arquivo para escrita, se o arquivo já existe, ele apaga e cria um novo para gravar as informações que lhe forem ordenadas.
- OpenU
  - Abre um arquivo para leitura e escrita. Caso o arquivo já exista, ele simplesmente adiciona as informações ao final do arquivo.

Exemplo

```
PRO abre_arquivo
    loc = 1
    OPENR, loc, 'arquivo.txt'
    ; Trabalhar com este arquivo
END
```

Arquivo na Memória

- Quando se abre um arquivo no código fonte ele fica armazenado em um local na memória. (loc)
- Pode-se definir um local ou solicitar um local automaticamente (recomendável).
- Ao se concluir a execução é **altamente recomendável** liberar aquele local da memória onde o arquivo foi armazenado.

Solicitar e Liberar

- GET_LUN
  - Procedimento usado para solicitar um local vazio na memória.
- GET_LUN, loc
  - OPENR, loc, ‘arquivo.txt’, /get_lun
- FREE_LUN, loc
  - Libera o local da memória.

Ler e imprimir em arquivo

- READF
  - Ler variáveis de determinado arquivo sequencialmente.
  - READF, loc, X
- PRINTF
  - Imprime variáveis em arquivo sequencialmente.
  - PRINTF, loc, X

Exemplo (estrutural)

```
PRO ler_arquivo

   OPENR, loc, 'arquivo.txt', /get_lun
   i = 0
   WHILE ~( EOF(loc) ) DO BEGIN
      READF, loc, x, y
      i++
   ENDWHILE

   x = dindgen(i)*0
   y = x

   FOR j=0, i DO READF, loc, x[j], y[j]
   CLOSE, loc
   FREE_LUN, loc

END
```

Leitura e Escrita

- Sequencial
  - Segue um indicador que percorre o arquivo a medida que se lê ou escreve-o.
- EOF(loc)
  - Indica se o final do arquivo fora atingido. Retorna ‘1’ para afirmativo, e ‘0’ para o caso negativo.
- **CLOSE**, loc
  - Fecha o arquivo aberto, salvando-o definitivamente.

Exemplo (vetorial)

```
PRO ler_arquivo

    OPENR, loc, 'arquivo.txt', /get_lun
    READF, loc, dados
    x = dados[0,*]
    y = dados[1,*]
    CLOSE, loc
    FREE_LUN, loc
END
```

Atividade (monstro)

- Montar um **projeto** de um simulador do modelo Okumura Hata dividido em pequenos blocos.
- O projeto deverá ser organizado em um **Script**.
- Deverá ser dividido em algumas sub-rotinas, e algumas funções obedecendo as divisões solicitadas a seguir.

Projeto

- Subrotina 1:
  - Simula o modelo Okumura Hata e chama subrotina que salva os resultados.
- Subrotina 2:
  - Salva os valores em um arquivo ‘dados.txt’
- Subrotina 3:
  - Ler arquivo ‘dados.txt’ e compartilha as variáveis dos dados.

Projeto

- Subrotina 4:
  - Ler arquivos compartilhados e calcula média, número de positivos e número de negativos através de funções, e chama subrotina para gerar mini-relatório.
- Subrotina 5:
  - Salva arquivo com mini-relatório, com os dados da simulação, a média, o número de negativos e o número de positivos.
- Subrotina 6:
  - Gera gráfico dos dados simulados.

Projeto

- Função 1:
  - Calcula a perda de potencial seguindo o modelo Okumura Hata.
- Função 2:
  - Calcula a média de uma série de dados.
- Função 3:
  - Calcula o número de positivos de uma série de números.
- Função 4:
  - Calcula o número de negativos de uma série de números.

Dúvidas?

HTTP://IDLTUTORIAL.BLOGSPOT.COM

ANTONIOPAULOVP@GMAIL.COM

LUCIOMARASSI@GMAIL.COM