



Programação de Computadores:

Introdução ao FORTRAN

Luis Martí

Instituto de Computação

Universidade Federal Fluminense

lmarti@ic.uff.br - <http://lmarti.com>

Introdução ao FORTRAN

- Cinco aspectos precisam ser considerados antes de escrever o primeiro programa em FORTRAN
 - Como um programa é **organizado**
 - Quais são os **tipos de dados** disponíveis
 - Como declarar **variáveis**
 - Como fazer a **entrada e saída** de dados
 - Como **atribuir valores** às variáveis

Organização de Programas

- O programa deve ser escrito utilizando um editor de código, como o **Force** (<http://force.lepsch.com>)
- O FORTRAN faz exigências quanto o uso das colunas no código fonte
 - As **colunas 1 a 6 são reservadas** para controle
 - As **colunas 7 a 72** são utilizadas **para o programa** em si
 - As **colunas 73 em diante** são ignoradas
- Você **DEVE** documentar o código fonte

Documentação do Código Fonte

- Sintaxe
 - Iniciar a coluna 1 da linha de comentário com C ou *, com isso o restante da linha será ignorada
- Boas práticas
 - Utilize linguagem natural (português)
 - Descreva a finalidade do programa nas primeiras linhas do código fonte
 - Descreva o que cada “pedaço” de código faz

Exemplo Simples de Programa FORTRAN

Comentário



C Este programa imprime a expressão

C "Hello World" na saída padrão

A palavra PROGRAM indica o início do código fonte do programa

Nome do programa

```
program p1hello
```

O comando PRINT escreve na saída padrão (tela do computador)

```
print *, 'Hello World!'
```

A palavra END indica o fim do código fonte do programa

6 colunas

```
end
```

Tipos de Dados

- Toda variável declarada **deve ter um tipo de dado**
- Lembre-se
 - O tipo define quais são os **valores possíveis**
 - Diz ao compilador **quanto de memória** deve ser reservado para a variável
 - Os tipos dividem-se em
 - Numéricos: inteiro, real e real com dupla precisão
 - Não numéricos: caractere e lógico

Tipos de Dados no FORTRAN (32bits)

- Tipo inteiro: **integer**
 - Valores **numéricos inteiros** de -2^{31} a 2^{31}
 - Exemplos: 6, 123, -45
- Tipo real com precisão simples: **real**
 - Valores **numéricos reais** com precisão de até **sete casas decimais**
 - Magnitude de 10^{-38} a 10^{38}
 - O ponto (".") separa a parte inteira da parte fracionária
 - Exemplo: $0.6023E24 = 0.6023 \times 10^{24}$

Tipos de Dados no FORTRAN (32bits)

- Tipo real com precisão dupla: **double precision**
 - Valores **numéricos reais** com precisão de até **quinze casas decimais**
 - Magnitude de 10^{-308} a 10^{308}
 - Exemplos: $0.54336D-94 = 0.54336 \times 10^{-94}$
- Tipo lógico: **logical**
 - Permite representar **valores lógicos**, ou seja, verdadeiro (.true.) ou falso (.false.)
 - Note que é preciso utilizar um ponto antes e outro depois dos valores

Tipos de Dados no FORTRAN

- Tipo caractere: **character**
 - Variáveis deste tipo permitem a **representação de texto** demarcado por aspas simples (‘)
 - Exemplo: ‘Isso é um texto’
 - Para utilizar o caractere ‘ no meio do texto, coloque duas aspas simples consecutivas
 - Exemplo: ‘Quero usar “aspas” aqui’

Declaração de Variáveis

- FORTRAN aceita declaração **implícita e explícita** de variáveis
- **Declaração Implícita**

```
program p2implic
print *, 'Informe A: '
read *, a

x = 2 + a
print *, 'Total: ', x

end
```

Declaração Explícita

```
program p3explic
real a, x
print *, 'Informe A: '
read *, a

x = 2 + a
print *, 'Total: ', x

end
```

Declaração Implícita

- **Declaração implícita** se baseia na primeira letra do identificador (nome) da variável
 - Variáveis que começam com a-h ou o-z são reais
 - Variáveis que começam com i-n são inteiras
- Contudo, isso pode **levar a erros!**
- É recomendado utilizar declaração explícita

Declaração Explícita

- A **declaração explícita** define qual é o tipo de cada variável que será utilizada no programa
- A declaração **deve vir antes** de todos os demais comandos
- Regras para identificadores (nomes) de variáveis
 - Ter somente letras de A a Z, números de 0 a 9, e _
 - Começar com uma letra
 - Maiúsculas e minúsculas são equivalentes
- Use o comando especial **implicit none** no início do programa para forçar o uso de declaração explícita

Exemplo de uso do “implicit none”

Veja o arquivo p4forced.f

Declaração Explícita

- Mais de uma variável pode ser declarada numa mesma lista

```
integer contador, n  
real X, Y, Z
```

caractere de tamanho 20

```
character*20 nome
```

nome é uma variável

nome tem tamanho 20 e endereco tamanho 30

```
character nome*20, endereco*30
```

```
logical teste, aberto
```

```
double precision raio, diametro
```

Entrada e Saída de Dados

- Entrada
 - Leitura de valores informados pelo usuário ou vindos de arquivo
 - Comando `read`
- Saída
 - Impressão na tela ou em arquivo
 - Comandos `write` e `print`
- Por simplicidade, utilizaremos apenas `read` e `print`

Sintaxe dos Comandos “read” e “print”

- Ambos os comandos esperam o formato do dado e a lista de parâmetros
 - `read (unidade, formato) lista_de_parâmetros`
 - `print (unidade, formato) lista_de_parâmetros`
- *Por enquanto utilizaremos formato livre*
 - `read *, altura, base`
 - `print *, 'A altura é ', altura, ' e a base é ', base`

O asterisco seguido de vírgula indica formato livre

Exemplos de Declaração, Entrada e Saída

Veja o arquivo `p5misc.f`

Atribuição de Valores

- No FORTRAN, o **símbolo de igual (=)** representa a **atribuição de valores** a variáveis
 - Exemplo: $x = 5 + b$ (a variável x recebe o valor de $5 + b$)
- Na escrita de algoritmos, o símbolo \leftarrow representa atribuição e o símbolo $=$ é o operador lógico de comparação de igualdade
 - Não confunda os símbolos $=$ da escrita de um algoritmo com o $=$ em FORTRAN!

Exemplo de Atribuição de Valores

Veja o arquivo p6atribu.f

Operadores Aritméticas no FORTRAN

Descrição	Álgebra	FORTRAN	Prioridade
Parênteses	(...)	(...)	1
Potenciação	A^B	A ** B	2
Multiplicação	$A \times B$	A * B	3
Divisão	$A \div B$	A / B	3
Adição	$A + B$	A + B	4
Subtração	$A - B$	A - B	4

Conflito de Prioridade

- Para duas operações com a **mesma prioridade**, a avaliação é feita da **esquerda para a direita**
 - $8 + 512 - 2 \rightarrow 520 - 2 \rightarrow 518$
 - $6 * 3 * 4 \rightarrow 18 * 4 \rightarrow 72$
- Contudo, no caso da **potência**, a avaliação é feita da **direita para a esquerda**
 - $4 ** 3 ** 2 \rightarrow 4 ** 9 \rightarrow 262144$
- Para evitar problemas, **sempre utilize parênteses**

Operação de Divisão com Inteiros

- Aritmética de inteiros
 - Na operação de divisão aplicada a valores inteiros, o resultado será **somente a parte inteira** da divisão
 - Ex.: $3 / 2 = 1$
- Aritmética em modo misto
 - Na operação de divisão aplicada a um valor real e um inteiro, os **inteiros serão automaticamente convertidos** para reais
- Exemplo
 - Veja o arquivo p7divide.f

Funções Matemáticas

Descrição	FORTTRAN	Exemplo
Valor absoluto	ABS(X)	ABS(-5.0) → 5.0
Resto da divisão	MOD(X, Y)	MOD(5, 2) → 1
Arredondamento	NINT(X)	NINT(5.7) → 6
Raiz quadrada	SQRT(X)	SQRT(2.0) → 1.41421354
Arco cosseno	ACOS(X)	ACOS(-1.0) → 3.14159274
Cosseno	COS(X)	COS(0.0) → 1.0
Arco seno	ASIN(X)	ASIN(1.0) → 1.57079637
Seno	SIN(X)	SIN(0.0) → 0.0
Arco tangente	ATAN(X)	ATAN(1.0) → 0.785398185
Tangente	TAN(X)	TAN(1.0) → 1.55740774
Exponencial	EXP(X)	EXP(1.0) → 2.71828175
Logaritmo natural	LOG(X)	LOG(2.71828175) → 1.0
Logaritmo na base 10	LOG10(X)	LOG10(100.0) → 2.0
Maior número de uma lista	MAX(...)	MAX(1, 5, 10) → 10
Menor número de uma lista	MIN(...)	MIN(1, 5, 10) → 1
Converter para inteiro	INT(X)	INT(5.7) → 5
Converter para real	REAL(X)	REAL(5) → 5.0
Converter para precisão dupla	DBLE(X)	DBLE(5) → 5.0

Estruturas de Decisão no FORTRAN

- SE ... ENTÃO ... SE NÃO ... FIM SE

```
IF (condição) THEN
    bloco de comandos
ELSE
    bloco de comandos
END IF
```

- SE ... ENTÃO ... FIM SE

```
IF (condição) THEN
    bloco de comandos
END IF
```


Operadores Relacionais no FORTRAN

Descrição	Operador	FORTRAN	Exemplo
Igual (equal)	=	.EQ.	3 .EQ. 2 → .FALSE.
Diferente (not equal)	≠	.NE.	3 .NE. 2 → .TRUE.
Menor que (less than)	<	.LT.	3 .LT. 2 → .FALSE.
Menor ou igual (less or equal)	≤	.LE.	3 .LE. 2 → .FALSE.
Maior que (greater than)	>	.GT.	3 .GT. 2 → .TRUE.
Maior ou igual (greater or equal)	≥	.GE.	3 .GE. 2 → .TRUE.

- Exemplo
 - Veja o arquivo p8oprela.f

Operadores Lógicos no FORTRAN

Operador	Condição para Verdadeiro	FORTRAN	Exemplo
E	Ambos os valores verdadeiros	.AND.	.TRUE. .AND. .TRUE. → .TRUE. .TRUE. .AND. .FALSE. → .FALSE. .FALSE. .AND. .TRUE. → .FALSE. .FALSE. .AND. .FALSE. → .FALSE.
OU	Um dos valores verdadeiro	.OR.	.TRUE. .OR. .TRUE. → .TRUE. .TRUE. .OR. .FALSE. → .TRUE. .FALSE. .OR. .TRUE. → .TRUE. .FALSE. .OR. .FALSE. → .FALSE.
Negação	O valor ser falso	.NOT.	.NOT. .TRUE. → .FALSE. .NOT. .FALSE. → .TRUE.

- Exemplo
 - Veja o arquivo p9oplogi.f

Estruturas de Repetição

- Permitem que um bloco de comandos seja executado **diversas vezes**
- **Dois tipos** principais de estruturas de repetição
 - Condicional
 - Contável

Repetição Condicional

```
DO WHILE (<condição>)  
    <bloco de comandos>  
END DO
```

- Significado

- *<bloco de comandos>* é executado repetidamente enquanto *<condição>* for verdadeira
- Verificação da *<condição>* é realizada antes de cada repetição
- Equivale na ENQUANTO ... REPETIR ...

Exemplo de Repetição Condicional

Veja o arquivo `p10loop1.f`

Simulando o REPETIR... ENQUANTO ...

- Existem casos em que é mais prático fazer a verificação no final do processamento
 - Ou seja, o bloco de comandos é executado ao menos uma vez
 - Nestes casos, utilizamos uma variável lógica de apoio

LOGICAL repete

```
repete = .TRUE.  
DO WHILE (repete)  
  <bloco de comandos>  
  repete = <condição>  
END DO
```

Exemplo de Repetição Condicional

Veja o arquivo `p1 1 loop2.f`

Repetição Contável

```
DO <variável> = <início>, <fim>,  
  <bloco de comandos>  
END DO
```

<incremento>

Argumento opcional.

Sua ausência significa
que é igual a 1.

Pode ser negativo.

- Significado

- A <variável> inicia com <início> e a cada repetição é incrementada de <incremento>
- Para cada repetição <bloco de comandos> é executado, enquanto a <variável> for menor ou igual que <fim>
- Equivale ao PARA ... ATÉ ... REPETIR ...

Exemplo de Repetição Contável

Veja o arquivo p12loop3.f

Material adaptado por Luis Martí a partir dos slides de Leandro Augusto Frata Fernandes.