



UNIVERSIDADE DO ESTADO DO RIO DE JANEIRO
CENTRO DE TECNOLOGIA E CIÊNCIAS
FACULDADE DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA CARTOGRÁFICA

Computação Aplicada à Cartografia II

Prof. Luiz Paulo Souto Fortes, PhD

fortes@pq.cnpq.br

1

Página em branco

Plano de Curso

1. Considerações iniciais. Plano de curso
2. Introdução ao ambiente de desenvolvimento Open Watcom
3. Introdução ao FORTRAN
4. Introdução ao FORTRAN
5. Introdução ao FORTRAN
6. Exercícios práticos
7. Primeira Avaliação
8. Introdução ao FORTRAN
9. Introdução ao FORTRAN
10. Introdução ao FORTRAN
11. Exercícios práticos em Cartografia
12. Segunda avaliação
13. Exame final

Prof. Luiz Paulo S. Fortes Computação Aplicada à Cartografia II

3

Bibliografia

- Open Watcom FORTRAN 77 Guides

Prof. Luiz Paulo S. Fortes Computação Aplicada à Cartografia II

4

Introdução ao FORTRAN (1/18)

- Introdução
 - FORTRAN: **FOR**mula **TRAN**slation
 - Desenvolvida a partir de 1954
 - Várias versões
 - FORTRAN I (1954-1957)
 - FORTRAN II (1958)
 - FORTRAN III (1958) – nunca liberada para o público
 - FORTRAN IV (1961)
 - FORTRAN 66 standard (1962-1966) – primeira versão padronizada para qualquer computador
 - FORTRAN 77 standard
 - FORTRAN 90 standard
 - FORTRAN 95 standard
 - Um das linguagens mais usadas de todos os tempos (ainda hoje...) !!

Prof. Luiz Paulo S. Fortes Computação Aplicada à Cartografia II

5

Introdução ao FORTRAN (2/18)

- O conjunto de caracteres FORTRAN consiste de 26 letras, 10 dígitos e 13 caracteres especiais
 - Letras
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
 - Dígitos
0 1 2 3 4 5 6 7 8 9
 - Caracteres especiais
branco = + - * / () , . \$ ' :
- É um subconjunto do conjunto de caracteres do sistema computacional que está sendo usado
- OpenWatcom também inclui os seguintes caracteres
! \ %

Prof. Luiz Paulo S. Fortes Computação Aplicada à Cartografia II

6

Introdução ao FORTRAN (3/18)

- Um programa é formado por linhas
 - Cada linha pode ser de um dos três tipos:
 1. Linha de comentário
 - Possui um “C”, “c” ou “*” na coluna 1
 - Podem estar situadas em qualquer posição do programa fonte
 - Não existe restrição quanto ao número de linhas de comentário no programa (quanto mais, melhor!!)
 - Além disso, a partir de um “!” em qualquer posição de uma linha (exceto na coluna 6 ou entre apóstrofes), o conteúdo do resto da linha é interpretado como comentário
 - Qualquer caracter do conjunto do sistema computacional pode ser utilizado
 - Exemplos
- ```
C Início do programa
c Escrever resultados
* Cálculo da média
 a = soma/n ! Cálculo da média
```

Prof. Luiz Paulo S. Fortes      Computação Aplicada à Cartografia II

7

### Introdução ao FORTRAN (4/18)

- Um programa é formado por linhas
  - Cada linha pode ser de um dos três tipos:
    2. Linha inicial
      - Primeira linha de um comando FORTRAN
      - Coluna 6 deve ser branco (ou 0)
      - Colunas 1 a 5 podem ter um número de comando (**statement label**)
      - Um número de comando não pode aparecer repetido em um módulo de programa (**program unit**)
      - Um módulo de programa é uma sequência de comandos e de linhas de comentário, terminando em um comando END
      - O corpo de um comando pode ser escrito da coluna 7 à 72
      - As colunas 73 e além são campos de sequência (**Sequence field**), sendo ignorados pelo compilador
      - Dependendo de uma opção do compilador Open Watcom FORTRAN 77, o comando pode ir até a coluna 132 (melhor não usar...)

Prof. Luiz Paulo S. Fortes      Computação Aplicada à Cartografia II

8

## Introdução ao FORTRAN (5/18)

- Um programa é formado por linhas
  - Cada linha pode ser de um dos três tipos:
    3. Linha de continuação
      - Um comando FORTRAN pode continuar numa nova linha
      - Coluna 6 deve ser diferente de branco (ou 0), usando-se um caracter válido do FORTRAN ou do sistema computacional (extensão Open Watcom)
      - Colunas 1 a 5 devem estar em branco
      - FORTRAN 77 permite até 19 linhas de continuação
      - O compilador Open Watcom FORTRAN 77 permite até 61 linhas de continuação quando ocupa-se até a coluna 72, ou até 31 linhas, quando ocupa-se até a coluna 132

Prof. Luiz Paulo S. Fortes      Computação Aplicada à Cartografia II

9

## Introdução ao FORTRAN (6/18)

- Exemplos

```
C This and the following five lines are comment lines
c The following statement "INDEX = INDEX + 2" has a
c statement number and is continued by placing a "$"
c in column 6.
* Column Numbers
*234567890

10 INDEX = INDEX
 $ + 2

* The above blank lines are treated like comment lines.
```

Prof. Luiz Paulo S. Fortes      Computação Aplicada à Cartografia II

10

## Introdução ao FORTRAN (7/18)

- Exemplos

```
* From the quadratic equation
*
* 2
* ax + bx + c = 0
*
* we derive the following two equations:
*
* + / 2
* -b - \ / b - 4ac
* x = -----
* 2a
*
* and express these equations in FORTRAN as:
*234567890
*
* X1 = (-B + SQRT(B**2 - 4 * A * C))
* $ / (2 * A)
* X2 = (-B - SQRT(B**2 - 4 * A * C))
* $ / (2 * A)
```

Prof. Luiz Paulo S. Fortes      Computação Aplicada à Cartografia II

11

## Introdução ao FORTRAN (8/18)

- Ordem das linhas e comandos
  - O primeiro comando de um módulo de programa é um dos seguintes comandos:
    - PROGRAM, FUNCTION, SUBROUTINE ou BLOCK DATA
    - O comando PROGRAM identifica o início do programa principal e só pode haver um deles em um programa FORTRAN executável
    - A execução de um programa FORTRAN inicia no primeiro comando executável do programa principal
    - Os comandos FUNCTION, SUBROUTINE e BLOCK DATA identificam o início de subprogramas
    - Se o primeiro comando de um módulo de programa não é um dos comandos acima, ele é interpretado como um programa principal

Prof. Luiz Paulo S. Fortes      Computação Aplicada à Cartografia II

12

## Introdução ao FORTRAN (9/18)

- Ordem das linhas e comandos

|                      |                                                           |            |                                      |
|----------------------|-----------------------------------------------------------|------------|--------------------------------------|
|                      | PROGRAM, FUNCTION, SUBROUTINE, or<br>BLOCK DATA Statement |            |                                      |
|                      |                                                           | PARAMETER  | IMPLICIT<br>Statements               |
| Comment<br>Lines     | ENTRY                                                     | Statements | Other<br>Specification<br>Statements |
|                      | and                                                       |            |                                      |
|                      | FORMAT                                                    |            | Statement                            |
| INCLUDE<br>Statement | Statements                                                | DATA       | Function<br>Statements               |
|                      |                                                           | Statements | Executable<br>Statements             |
|                      | END Statement                                             |            |                                      |

Prof. Luiz Paulo S. Fortes

Computação Aplicada à Cartografia II

13

## Introdução ao FORTRAN (10/18)

- Nomes simbólicos
  - Nomes simbólicos são nomes que representam variáveis, matrizes, funções, etc.
  - Nomes são formados por letras maiúsculas A-Z e dígitos 0-9, sendo o primeiro obrigatoriamente uma letra
  - Nomes simbólicos são restritos a 6 caracteres
  - Exemplos: AMOUNT  
AGE  
CUST73
  - Extensões do Open Watcom FORTRAN 77:
    - Letras minúsculas a-z, \$ e \_, podendo ser o primeiro caracter do nome
    - Nomes de até 32 caracteres
    - Exemplos: Evaluate  
\$Cheque  
ComputeAverage  
\_device  
IO\$ERROR  
student\_total
    - Maiúsculas e minúsculas significam a mesma coisa e espaços são ignorados
      - Account, ACCount, ACCOUNT; C R E D I T, C R E D I T

Prof. Luiz Paulo S. Fortes

Computação Aplicada à Cartografia II

14

### Introdução ao FORTRAN (11/18)

- 6 Tipos de dados em FORTRAN: logical, integer, real, double precision, complex e character (ou seja, logical, numeric ou character)

| Data Type        | Size<br>(bytes) | Standard<br>FORTRAN |
|------------------|-----------------|---------------------|
| LOGICAL          | 4               | yes                 |
| LOGICAL*1        | 1               | extension           |
| LOGICAL*4        | 4               | extension           |
| INTEGER          | 4               | yes                 |
| INTEGER*1        | 1               | extension           |
| INTEGER*2        | 2               | extension           |
| INTEGER*4        | 4               | extension           |
| REAL             | 4               | yes                 |
| REAL*4           | 4               | extension           |
| REAL*8           | 8               | extension           |
| DOUBLE PRECISION | 8               | yes                 |
| COMPLEX          | 8               | yes                 |
| COMPLEX*8        | 8               | extension           |
| DOUBLE COMPLEX   | 16              | extension           |
| COMPLEX*16       | 16              | extension           |
| CHARACTER        | 1               | yes                 |
| CHARACTER*n      | n               | yes                 |

Prof. Luiz Paulo S. Fortes      Computação Aplicada à Cartografia II

15

### Introdução ao FORTRAN (12/18)

- Faixa de variação dos valores para cada tipo

| Data Type        | Valores possíveis                                       |
|------------------|---------------------------------------------------------|
| LOGICAL          | .true. Ou .false.                                       |
| LOGICAL*1        | .true. Ou .false.                                       |
| LOGICAL*4        | .true. Ou .false.                                       |
| INTEGER          | $-2147483648 \leq n \leq 2147483647$                    |
| INTEGER*1        | $-128 \leq n \leq 127$                                  |
| INTEGER*2        | $-32768 \leq n \leq 32767$                              |
| INTEGER*4        | $-2147483648 \leq n \leq 2147483647$                    |
| REAL             | $1.175494e-38 \leq m \leq 3.402823e38$                  |
| REAL*4           | $1.175494e-38 \leq m \leq 3.402823e38$                  |
| REAL*8           | $2.2250738585072e-308 \leq m \leq 1.79769313486232e308$ |
| DOUBLE PRECISION | $2.2250738585072e-308 \leq m \leq 1.79769313486232e308$ |
| COMPLEX          | (real, real)                                            |
| COMPLEX*8        | (real*4, real*4)                                        |
| DOUBLE COMPLEX   | (real*8, real*8)                                        |
| COMPLEX*16       | (real*8, real*8)                                        |
| CHARACTER        | 1 character                                             |
| CHARACTER*n      | n caracteres                                            |

- OBS: tipos adicionais podem ser criados com o comando STRUCTURE

Prof. Luiz Paulo S. Fortes      Computação Aplicada à Cartografia II

16



### Introdução ao FORTRAN (13/18)

- Tipo de dado em função do nome da variável
  - Por *default*, o FORTRAN assume que as variáveis iniciadas em I, J, K, L, M, N são do tipo inteiro, sendo as restantes do tipo real
  - Esse *default* pode ser alterado com o comando IMPLICIT
  - A programação estruturada recomenda que não se use o *default* pois erros podem ocorrer com facilidade
    - Por exemplo, erro de digitação dos nomes de variáveis

```
Real*8 lat, long, lat_rad, lon_rad
...
lat = -22.
long = -47.
...
lon_rad = lon * pi / 180. ! lon_rad = ZERO !!
```

- No início de todo o programa FORTRAN usar sempre o comando **IMPLICIT NONE**

Prof. Luiz Paulo S. Fortes

Computação Aplicada à Cartografia II

17

### Introdução ao FORTRAN (14/18)

- Constantes

#### Exemplos

#### LOGICAL

```
.true.
.false.
```

#### INTEGER

```
1423
+345
-34565788
```

#### CHARACTER

```
'ABCDEFG1234567'
'There's always tomorrow'
```

Prof. Luiz Paulo S. Fortes

Computação Aplicada à Cartografia II

18

## Introdução ao FORTRAN (15/18)

- Constantes (continuação)

### Exemplos

#### REAL

123.764  
 .4352344  
 1423.34E12  
 +345.E-4  
 -.4565788E3  
 2E6  
 1234.

#### REAL\*8

1423.34D12  
 +345.D-4  
 -.4565788D5  
 2D6

- É possível atribuir um nome simbólico a uma constante com o uso do comando PARAMETER (o valor da constante não pode mudar no pgm!)  
 PARAMETER (PI=3.141592653589793D0)

Prof. Luiz Paulo S. Fortes

Computação Aplicada à Cartografia II

19

## Introdução ao FORTRAN (16/18)

- Vetores e matrizes (*Arrays*)
  - Podem ter até 7 dimensões
  - Normalmente são usadas até 3 dimensões
    - Uma dimensão: vetor
    - Duas dimensões: matriz
    - Três dimensões: array de matrizes paralelas
  - Forma genérica:
    - a (d [,d] ...), onde d é declarador de dimensão do tipo
      - d = [limite\_inferior:] limite superior
  - Exemplos  
 DIMENSION A(10), B(-5:5,-10:10)  
 INTEGER C(10,20)  
 COMMON /DATA/ X, Y(30,30), Z
  - O tamanho máximo de um *array* é limitado pela memória (RAM) disponível
  - Elementos de um *array*
    - A(5), B(-3, 9), C(7, 16), Y(21, 13)
  - Todos os elementos de um array são do mesmo tipo

Prof. Luiz Paulo S. Fortes

Computação Aplicada à Cartografia II

20

## Introdução ao FORTRAN (17/18)

- Comando de atribuição:  $V = E$   
Onde V é uma variável numérica, lógica ou caracter e E uma expressão do tipo aritmética, relacional/lógica ou de caracter, respectivamente
- Expressões
  - Aritméticas
    - Operações entre valores de tipo numérico e que têm como resultado um valor numérico
    - Operadores aritméticos: + - \* / \*\* (em ordem **inversa** de precedência)
    - Exemplo:  $B + (C - D) * 4 / 2$
  - Relacionais
    - Operações entre dois valores do mesmo tipo e que têm como resultado um valor lógico
    - Operadores relacionais: **.EQ. .NE. .GT. .LT. .GE. .LE.**
    - Exemplo:  $A .LT. B$
  - Lógicas
    - Operações entre valores lógicos e que têm como resultado valores lógicos
    - Operadores lógicos **.XOR. .OR. .AND. .NOT.** (em ordem inversa de precedência)
    - Exemplo:  $D .OR. E .AND. F \{ = D .OR. (E .AND. F) \}$

Prof. Luiz Paulo S. Fortes      Computação Aplicada à Cartografia II

21

## Introdução ao FORTRAN (18/18)

- Operadores lógicos

| A       | B       | A .AND. B | A .OR. B | A .XOR. B | .NOT. A |
|---------|---------|-----------|----------|-----------|---------|
| .TRUE.  | .TRUE.  | .TRUE.    | .TRUE.   | .FALSE.   | .FALSE. |
| .TRUE.  | .FALSE. | .FALSE.   | .TRUE.   | .TRUE.    | .FALSE. |
| .FALSE. | .TRUE.  | .FALSE.   | .TRUE.   | .TRUE.    | .TRUE.  |
| .FALSE. | .FALSE. | .FALSE.   | .FALSE.  | .FALSE.   | .TRUE.  |

Prof. Luiz Paulo S. Fortes      Computação Aplicada à Cartografia II

22

## Introdução ao FORTRAN (1/20)

- Comandos de controle
  - O FORTRAN foi desenvolvido ANTES dos conceitos de programação estruturada
  - Importante usar apenas os comandos que seguem estes conceitos estruturados (i.e., comandos sequenciais, condicionais e de repetição)
  - **NÃO** usar o comando **GO TO !!!!**

Prof. Luiz Paulo S. Fortes      Computação Aplicada à Cartografia II

23

## Introdução ao FORTRAN (2/20)

- Comandos de controle (Cont.)
  - Comandos condicionais
    - IF – THEN – ELSE – END IF
      - Implementa o pseudo-código **Se-então-Senão-Fim Se**
      - Caso Geral

```
IF (expressão_lógica) THEN
 comando(s)
ELSE
 comando(s)
END IF
```
      - 1º caso particular: ausência do bloco ELSE

```
IF (expressão_lógica) THEN
 comando(s)
END IF
```
      - 2º caso particular (versão antiga do IF):

```
IF (expressão_lógica) comando
```

Prof. Luiz Paulo S. Fortes      Computação Aplicada à Cartografia II

24

## Introdução ao FORTRAN (3/20)

- Comandos de controle (Cont.)
  - Comandos condicionais
    - IF – THEN – ELSE – END IF
    - Exemplos

```
IF (A .GT. B) THEN
 Write (*,*) 'A maior que B'
 A = A - B
ELSE
 Write (*,*) 'A não maior que B'
END IF
```

```
IF (A .GT. B) A = A - B
```

Prof. Luiz Paulo S. Fortes      Computação Aplicada à Cartografia II

25

## Introdução ao FORTRAN (4/20)

- Comandos de controle (Cont.)
  - Comandos condicionais
    - IF – THEN – ELSE IF – ELSE – END IF
    - Extensão do caso geral do IF-THEN-ELSE-ENDIF

```
IF (expressão_lógica_1) THEN
 comando(s)
ELSE IF (expressão_lógica_2) THEN
 comando(s)
...
ELSE
 comando(s)
END IF
```

- Na estrutura acima, o bloco ELSE IF pode se repetir
- O bloco ELSE pode ou não estar presente

Prof. Luiz Paulo S. Fortes      Computação Aplicada à Cartografia II

26

## Introdução ao FORTRAN (5/20)

- Comandos de controle (Cont.)
  - Comandos condicionais
    - IF – THEN – ELSE IF – ELSE – END IF
      - Exemplo

```
IF (A .GT. B) THEN
 Write (*,*) 'A maior que B'
 A = A - B
ELSE IF (A .LT. B) THEN
 Write (*,*) 'A menor que B'
 A = B - A
ELSE
 Write (*,*) 'A igual a B'
 A = 0.0
END IF
```

Prof. Luiz Paulo S. Fortes      Computação Aplicada à Cartografia II

27

## Introdução ao FORTRAN (6/20)

- Comandos de controle (Cont.)
  - Comandos condicionais
    - IF – THEN – ELSE – END IF
      - Exemplos de ninhos de IF

```
IF (A .NE. B) THEN
 IF (A .GT. B) THEN
 Write (*,*) 'A maior que B'
 A = A - B
 ELSE
 Write (*,*) 'A menor que B'
 A = B - A
 END IF
ELSE
 Write (*,*) 'A igual a B'
 A = 0.0
END IF
```

Prof. Luiz Paulo S. Fortes      Computação Aplicada à Cartografia II

28

## Introdução ao FORTRAN (7/20)

- Comandos de controle (Cont.)
  - Comandos de repetição
    - DO – END DO
      - Implementa o pseudo-código **Para-faça-Fim Para**
      - Caso geral (extensão do FORTRAN 77)

```
DO variável = início, fim [, incremento]
 comando(s)
END DO
```

- Versão FORTRAN 77

```
DO XX variável = início, fim [, incremento]
 comando(s)
XX CONTINUE
```

## Introdução ao FORTRAN (8/20)

- Comandos de controle (Cont.)
  - Comandos de repetição
    - DO – END DO
      - Exemplos

```
DO I = 1, 3
 DO J = 1, 5
 Write (*,*) MATRIZ(I, J)
 END DO
END DO
```

```
DO 20 I = 1, 3
 DO 10 J = 1, 5
 Write (*,*) MATRIZ(I, J)
 10 CONTINUE
 20 CONTINUE
```

## Introdução ao FORTRAN (9/20)

- Comandos de controle (Cont.)
    - Comandos de repetição
      - DO WHILE – END DO
        - Implementa o pseudo-código **Enquanto-faça-Fim Enquanto**
        - Extensão do FORTRAN 77
- ```

DO WHILE (expressão_lógica)
  comando(s)
END DO
  
```
- Exemplo


```

      I = 1
      DO WHILE (I .LE. 3)
        J = 1
        DO WHILE (J .LE. 5)
          Write (*,*) MATRIZ( I, J )
          J = J + 1
        END DO
        I = I + 1
      END DO
      
```

Prof. Luiz Paulo S. Fortes Computação Aplicada à Cartografia II

31

Introdução ao FORTRAN (10/20)

- Comandos de controle (Cont.)
 - Comandos de repetição
 - WHILE – DO – END WHILE
 - Também Implementa o **Enquanto-faça-Fim Enquanto**
 - Extensão do FORTRAN 77
- ```

WHILE (expressão_lógica) DO
 comando(s)
END WHILE

```
- Exemplo
 

```

 I = 1
 WHILE (I .LE. 3) DO
 J = 1
 WHILE (J .LE. 5) DO
 Write (*,*) MATRIZ(I, J)
 J = J + 1
 END WHILE
 I = I + 1
 END WHILE

```

Prof. Luiz Paulo S. Fortes    Computação Aplicada à Cartografia II

32



### Introdução ao FORTRAN (11/20)

- Comandos de controle (Cont.)
    - Comandos de repetição
      - LOOP – UNTIL
        - Implementa o pseudo-código **Repita-Até que**
        - Extensão do FORTRAN 77
- LOOP**  
comando(s)  
**UNTIL** (expressão\_lógica)
- Exemplo
- X = 1.0  
**LOOP**  
Write (\*,\*) X, SQRT(X)  
X = X + 1  
**UNTIL** (X .GT. 10.0)
- Qual a diferença entre WHILE DO e LOOP UNTIL ?

Prof. Luiz Paulo S. Fortes

Computação Aplicada à Cartografia II

33

### Introdução ao FORTRAN (12/20)

- Comandos de entrada e saída (*Input/Output*)
  - Fornecem meios para o programa FORTRAN se comunicar com os periféricos
  - O comando READ transfere dados para dentro do ambiente de execução do programa
  - Os comandos WRITE e PRINT transferem dados para fora do ambiente de execução
  - Outros comandos realizam funções adicionais, tais como posicionar um certo registro em um arquivo, estabelecer quais arquivos serão processados pelo programa, etc.

Prof. Luiz Paulo S. Fortes

Computação Aplicada à Cartografia II

34

## Introdução ao FORTRAN (13/20)

- Comandos de entrada e saída (*Input/Output*)
  - Arquivos
    - Podem ser externos ou internos
      - Trataremos aqui apenas dos arquivos externos
      - Arquivos externos existem ou são criados em meios tais como discos rígidos, impressoras, monitores, etc.
      - Um arquivo externo (ou simplesmente arquivo) pode existir antes da execução do programa ou pode ser criado e/ou apagado durante a execução
      - Arquivos são formados por **registros**
    - Propriedades de arquivos externos
      - **Nome**: geralmente um arquivo tem um nome
      - **Tipo de acesso**: refere-se à forma com que um registro é posicionado para leitura ou gravação no arquivo
        - » **Sequencial**: registros são lidos e gravados em ordem, desde o primeiro até o último
        - » **Direto**: registros podem ser lidos e gravados em qualquer ordem, através do referenciamento ao número do registro

Prof. Luiz Paulo S. Fortes

Computação Aplicada à Cartografia II

35

## Introdução ao FORTRAN (14/20)

- Comandos de entrada e saída (*Input/Output*)
  - Registros
    - Exemplo
      - Arquivo de pontos de um navegador GPS
      - Cada registro é composto pela identificação do ponto e suas coordenadas geodésicas (latitude, longitude e altitude elipsoidal), por exemplo
    - Podem ser formatados (*formatted*) e não formatados (*unformatted*)
      - **Formatados**: formados por caracteres
      - **Não formatados**: formados por valores numéricos, lógicos e tb caracteres; basicamente estes valores têm a mesma representação que possuem na memória do computador

Prof. Luiz Paulo S. Fortes

Computação Aplicada à Cartografia II

36

### Introdução ao FORTRAN (15/20)

- Comandos de entrada e saída (*Input/Output*)
    - Registros
      - Exemplo de registros formatados (*formatted*) e não formatados (*unformatted*)
- ```

100      INTEGER NUM
          NUM = 12345
          WRITE (*, 100) NUM      ! ⇒ 5 bytes para escrever NUM
          FORMAT(1X, I5)
          WRITE (UNIT=7) NUM      ! ⇒ 4 bytes para escrever NUM

```
- Registro de final de arquivo (*endfile*)
 - Registro especial situado após todos os outros registros de um arquivo, ou seja, localizado no final do arquivo
 - É um registro que não possui dados
 - Após todos os registros de um arquivo serem lidos, se uma nova tentativa de leitura é efetuada, o registro *endfile* é “lido” e a condição *end-of-file* é atingida

Prof. Luiz Paulo S. Fortes Computação Aplicada à Cartografia II

37

Introdução ao FORTRAN (16/20)

- Comandos de entrada e saída (*Input/Output*)
 - Unidades (*Units*)
 - Comandos de entrada e saída referem-se a arquivos externos através de números inteiros denominados **unidades**
 - Unidades são números inteiros que variam de 0 a 999
- ```

 WRITE (1,*) 'Gravação na unidade 1'

```
- O comando **OPEN** associa uma unidade a um arquivo particular, o que se denomina conexão
- ```

          OPEN (UNIT=1, FILE='FILE1')
          OPEN (UNIT=2, FILE='FILE2')

```
- Para desconectar uma unidade de um arquivo, pode-se usar o comando **CLOSE** ou usar **OPEN** com o mesmo número de unidade
- ```

 CLOSE (UNIT=1) ! ou
 OPEN (UNIT=1, FILE='FILE3')

```
- Existem unidades pre-conectadas a arquivos
    - \* ou **5**: teclado (para leitura)
    - \* ou **6**: monitor de vídeo (para escrita)

Prof. Luiz Paulo S. Fortes    Computação Aplicada à Cartografia II

38

## Introdução ao FORTRAN (17/20)

### • Comandos de entrada e saída (*Input/Output*)

#### – Especificadores

- Comandos de entrada e saída contêm um ou mais especificadores, dependendo do comando
- Especificador de unidade: [UNIT =] u
- Especificador de formato: [FMT =] f
- Especificador de registro: REC = nr
- Especificador de status de input/output: IOSTAT = ios
- Especificador de erros: ERR = e
- Especificador de final de arquivo: END = s
- Especificador de status de arquivo: STATUS = sta
- Especificador de acesso de arquivo: ACCESS = acc
- Especificador de formato de registro: FORM = fm
- Especificador de comprimento de registro: RECL = rcl

Prof. Luiz Paulo S. Fortes

Computação Aplicada à Cartografia II

39

## Introdução ao FORTRAN (18/20)

### • Comandos de entrada e saída (*Input/Output*)

#### – Especificadores

- Se **UNIT** é omitida, **u** deve ser o primeiro especificador usado no comando
- Se **FMT** é omitido, **f** deve ser o segundo especificador usado no comando; **f** também pode ser \* (= formato livre)
- Se **IOSTAT**, **ERR** e **END** são especificados, o programa não é terminado caso ocorra um erro ou uma condição *end-of\_file*
- **f**, **e**, **s** são números (inteiros) de comandos do programa
- **ios** = 0 (sem erro), > 0 (erro), < 0 (final de arquivo)
- **sta**
  - No comando OPEN: = 'OLD', 'NEW', 'SCRATCH', ou 'UNKNOWN', sendo este último o *default*; para 'OLD' e 'NEW', FILE tem que ser especificado; para 'SCRATCH', FILE não pode ser especificado (corresponde a arquivo temporário, apagado quando o comando CLOSE é executado)
  - No comando CLOSE: = 'KEEP' ou 'DELETE', sendo o primeiro o *default*, exceto para arquivos 'SCRATCH', onde o segundo é o *default* (e a única opção possível neste caso)
- **acc** = 'SEQUENTIAL' ou 'DIRECT', sendo o primeiro o *default*; 'APPEND' é uma extensão Open Watcom
- **fm** = 'FORMATTED' ou 'UNFORMATTED'; sendo o primeiro *default* para arquivos sequenciais e o segundo o *default* para arquivos de acesso direto
- **rcl** = comprimento do registro (inteiro); deve ser fornecido para arquivos de acesso direto e não deve estar presente em arquivos sequenciais

Prof. Luiz Paulo S. Fortes

Computação Aplicada à Cartografia II

40

### Introdução ao FORTRAN (19/20)

- Comandos de entrada e saída (*Input/Output*)

- Descrição dos mais usados

```
OPEN ([UNIT =] u, IOSTAT = ios, ERR = e, FILE = f, STATUS =
 sta, ACCESS = acc, FORM = fm, RECL = rcl)
```

```
CLOSE ([UNIT =] u, IOSTAT = ios, ERR = e, STATUS = sta)
```

```
READ ([UNIT =] u, [FMT=] f, REC = nr, IOSTAT = ios, ERR = e,
 END = s) lista_de_variáveis
```

```
WRITE ([UNIT =] u, [FMT=] f, REC = nr, IOSTAT = ios, ERR = e)
 lista_de_variáveis
```

```
PRINT f, lista_de_variáveis
```

Prof. Luiz Paulo S. Fortes      Computação Aplicada à Cartografia II

41

### Introdução ao FORTRAN (20/20)

- Comandos de entrada e saída (*Input/Output*)

- Exemplos

```
OPEN (UNIT=5, FILE='TEST', STATUS='UNKNOWN',
 ACCESS='SEQUENTIAL', FORM='FORMATTED')
OPEN (5, FILE='TEST')
READ (5, 100) X, Y, Z ! ou
READ (UNIT=5, FMT=100) X, Y, Z ! ou
100 FORMAT (3F10.5)
READ (UNIT=5, FMT='(3F10.5)') X, Y, Z ! ou
READ(5, '(3F10.5)') X, Y, Z
READ (*,100) X, Y, Z ! Leitura do teclado
READ (*, *) X, Y, Z ! Leitura com formato livre
WRITE (6, 100) X, Y, Z
WRITE (UNIT=6, FMT=100) X, Y, Z
WRITE (UNIT=6, FMT='(3F10.5)') X, Y, Z
WRITE (6, '(3F10.5)') X, Y, Z
WRITE (*, 100) X, Y, Z ! Escrita na tela
WRITE (*, *) X, Y, Z ! Escrita na tela com formato livre
CLOSE (5)
```

Prof. Luiz Paulo S. Fortes      Computação Aplicada à Cartografia II

42

### Exercícios Propostos (1/2)

- Desenhar o DFD (1º nível, i.e., diagrama de contexto), escrever o algoritmo em pseudo-código e depois implementar os programas-fonte em FORTRAN 77 para:
  - Calcular as raízes  $x_1$  e  $x_2 \in \mathbb{R}$  de uma equação do 2º grau
  - Os coeficientes  $a$ ,  $b$  e  $c$  da equação devem ser lidos de teclado
  - A saída do programa (as raízes da equação) será fornecida na tela
  - Caso não haja raízes  $\in \mathbb{R}$ , emitir mensagem “Não há raízes reais”

Prof. Luiz Paulo S. Fortes

Computação Aplicada à Cartografia II

43

### Exercícios Propostos (2/2)

- Desenhar o DFD (1º nível, i.e., diagrama de contexto), escrever o algoritmo em pseudo-código e depois implementar os programas-fonte em FORTRAN 77 para:
  - Calcular a soma dos  $n$  elementos de uma progressão aritmética de razão  $r$ 
    - O primeiro elemento da PA,  $n$  e  $r$  devem ser lidos do teclado
    - A saída do programa (os  $n$  elementos da PA e a sua soma) deve ser na tela
    - Usar um vetor para armazenar e exibir os elementos da PA (ou seja, os elementos devem ser escritos em linha na tela)
  - Calcular o fatorial de qualquer número inteiro
    - Fornecer o número do teclado
    - A saída do programa (o número lido e o fatorial) deve ser na tela
    - Emitir mensagem de erro no caso de números negativos ou não inteiros

Prof. Luiz Paulo S. Fortes

Computação Aplicada à Cartografia II

44

## Introdução ao FORTRAN (1/16)

- Comando FORMAT
  - Fornece os meios para se especificar a forma com que os dados internos são convertidos em cadeias de carácter e vice-versa
  - Sintaxe

**label FORMAT fs** , onde

**label:**

número de comando usado pelo comando de I/O para identificar o comando FORMAT

**fs:**

cadeia de caracteres contendo a especificação de formato

- Exemplo

```
REAL X
X = 234.43
WRITE (*, 100) X
100 FORMAT (F10.2)
END
```

Prof. Luiz Paulo S. Fortes

Computação Aplicada à Cartografia II

45

## Introdução ao FORTRAN (2/16)

- Comando FORMAT (Continuação)
  - Como alternativa ao uso do comando FORMAT, pode-se utilizar a cadeia de caracteres **fs** no próprio comando de I/O
  - Exemplo

|                                                                      |                                                                                     |                                                         |
|----------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------|
| <pre>REAL X X = 234.43 WRITE (*, 100) X 100 FORMAT (F10.2) END</pre> |  | <pre>REAL X X = 234.43 WRITE (*, '(F10.2)') X END</pre> |
|----------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------|

Prof. Luiz Paulo S. Fortes

Computação Aplicada à Cartografia II

46

### Introdução ao FORTRAN (3/16)

- Comando FORMAT (Continuação)

- Cadeia de caracteres **fs**

- Sintaxe

( flist )

**flist** é uma lista de descritores de formato separados por vírgula

- Descritores de formato

**red**, onde **ed** é um editor de formato repetível e **r** é um número inteiro positivo representando o número de ocorrências de **ed**

**ned**, onde **ned** é um editor de formato não repetível

**rfs**, onde **fs** é uma outra cadeia de caracteres

- Exemplo

(//, 2x, 'X =', F11.3, 2x, 'Y =', F11.3, 2x, 'Z =', F11.3)

Prof. Luiz Paulo S. Fortes

Computação Aplicada à Cartografia II

47

### Introdução ao FORTRAN (4/16)

- Comando FORMAT (Continuação)

- Alguns descritores de formato repetíveis

**Iw**      números inteiros

**Fw.d**    números reais

**Ew.d**    números reais de precisão simples

**Dw.d**    números reais de dupla precisão

**Lw**      valores lógicos

**A**        caracteres

**Aw**      cadeias de caracteres

onde **w** é o comprimento total do campo a ser lido ou escrito e **d** é o comprimento da parte fracionária

OBS: se o número a ser exibido for maior que **w** (descritores I, F, E e D), o campo é preenchido com asteriscos

Prof. Luiz Paulo S. Fortes

Computação Aplicada à Cartografia II

48



### Introdução ao FORTRAN (5/16)

- Comando FORMAT (Continuação)

- Alguns descritores de formato não repetíveis

|          |                                                          |
|----------|----------------------------------------------------------|
| /        | muda de registro ou de linha                             |
| X        | avança uma posição no registro ou linha                  |
| nX       | avança n posições no registro ou linha                   |
| 'hh...h' | grava ou imprime a cadeia de caracteres entre apóstrofes |
| nHhh..h  | grava ou imprime os n caracteres após o descritor H      |
| \$       | mantém o cursor na posição corrente (extensão)           |
| \        | mantém o cursor na posição corrente (extensão)           |

Prof. Luiz Paulo S. Fortes

Computação Aplicada à Cartografia II

49

### Introdução ao FORTRAN (6/16)

- Comando FORMAT (Continuação)

- Exemplos

```

C Exemplo de formato de LEITURA de variáveis
C234567890123456789012345678901234567890

 real*8 x, y, z, lat_seg, lon_seg, alt
 integer lat_grau, lat_min, lon_grau, lon_min
 character lat_sinal, lon_sinal
 character*4 ponto

 read (*,1000) ponto, x, y, z,
! lat_sinal, lat_grau, lat_min, lat_seg,
! lon_sinal, lon_grau, lon_min, lon_seg,
! alt
1000 Format (A4, 5X, 3(F11.3,1X), 4X,
! 2(A1,1X, 2(I2,1X), F8.5, 1X), F8.3)

```

Prof. Luiz Paulo S. Fortes

Computação Aplicada à Cartografia II

50

## Introdução ao FORTRAN (7/16)

- Comando FORMAT (Continuação)

- Exemplos

```

C Exemplo de formato de IMPRESSÃO de variáveis
C234567890123456789012345678901234567890
 real*8 x, y, x, lat_seg, lon_seg, alt
 integer lat_grau, lat_min, lon_grau, lon_min
 character lat_sinal, lon_sinal
 character*4 ponto
 write(*,2000) ponto, x, y, z,
! lat_sinal, lat_grau, lat_min, lat_seg,
! lon_sinal, lon_grau, lon_min, lon_seg,
! alt
2000 Format (' Ponto: ', A4, 5X,
! 'X = ', F11.3, 'm', 1X,
! 'Y = ', F11.3, 'm', 1X,
! 'Z = ', F11.3, 'm', 1X, /,
! 'Lat = ', A1, 2(I2, 1X), F8.5, 1X,
! 'Lon = ', A1, 2(I2, 1X), F8.5, 1X,
! 'Alt = ', F8.3, ' m')

```

## Introdução ao FORTRAN (8/16)

- Comando FORMAT (Continuação)

- Exemplos

```

C Exemplo de formato de LEITURA de uma matriz por
C linha, com até 8 elementos por registro
C234567890123456789012345678901234567890

 real*8 M(100,100)
 integer i, j, n

 n = 100

 DO i = 1, n
 read (*, 1000) (M(i,j), j=1,n)
1000 Format (8F10.3)
 END DO

```

## Introdução ao FORTRAN (9/16)

- Comando FORMAT (Continuação)

- Exemplos

```
C Exemplo de formato de IMPRESSÃO de uma matriz por
C linha, com até 8 elementos por linha
C234567890123456789012345678901234567890
```

```
 real*8 M(100,100)
 integer i, j, n

 n = 100

 DO i = 1, n
 write(*, 2000) i, (M(i,j), j=1,n)
2000 Format (' Linha ', i3, (8F10.3))
 END DO
```

## Introdução ao FORTRAN (10/16)

- Funções e subrotinas

- Uma unidade ou módulo de programa (*program unit*) é um conjunto de comandos FORTRAN e comentários, podendo ser um programa principal ou um subprograma
  - O programa principal é o módulo de programa onde a execução começa
    - a execução completa do programa principal implica na execução completa do programa
    - Um programa executável pode conter apenas um programa principal
  - Um subprograma é um módulo de programa que possui como primeiro comando SUBROUTINE, FUNCTION ou BLOCK DATA

### Introdução ao FORTRAN (11/16)

- Funções e subrotinas
  - Podem ser
    - Funções de comando (*Statement functions*)
    - Funções intrínsecas (*Intrinsic functions*)
    - Funções externas (*External functions*)
    - Subrotinas (*Subroutines*)

Prof. Luiz Paulo S. Fortes

Computação Aplicada à Cartografia II

55

### Introdução ao FORTRAN (12/16)

- Funções e subrotinas
  - Funções de comando (*Statement functions*)

```
SUBROUTINE CALC(U, V)
 REAL POLY, X, Y, U, V, Z, CONST
 *
 * Defina a Statement Function.
 *
 POLY(X,Y) = X**2 + Y**2 + 2.0*X*Y + CONST
 *
 * Invoque a Statement Function.
 *
 CONST = 23.5
 Z = POLY(U, V)

 PRINT *, Z
 RETURN
 END
```

Prof. Luiz Paulo S. Fortes

Computação Aplicada à Cartografia II

56

## Introdução ao FORTRAN (13/16)

- Funções e subrotinas
  - Funções intrínsecas (*Intrinsic functions*)
    - São fornecidas pelo compilador FORTRAN
      - Exemplo  
DSIN (LAT)
  - Funções externas (*External functions*)
    - É um módulo de programa que possui o comando FUNCTION como primeiro comando
      - Exemplo

```

INTEGER FUNCTION VECSUM (A, N)
INTEGER A(N), I
VECSUM = 0
DO I = 1, N
 VECSUM = VECSUM + A(I)
END DO
RETURN
END

```

Prof. Luiz Paulo S. Fortes      Computação Aplicada à Cartografia II

57

## Introdução ao FORTRAN (14/16)

- Funções e subrotinas
  - Subrotinas (*Subroutines*)
    - É um módulo de programa que possui o comando SUBROUTINE como primeiro comando
    - Uma subrotina difere de uma função externa no fato de que ela não retorna um resultado e portanto não possui um tipo associado a ela. Entretanto, é possível retornar valores através de seus argumentos
    - Ao contrário de uma função externa, uma subrotina não pode aparecer numa expressão; subrotinas são referenciadas usando-se o comando CALL
    - Exemplo

```

SUBROUTINE GRA_RAD (ANG, ANG_RAD)
REAL*8 ANG, ANG_RAD, PI
PI = 3.141592653589793D0
ANG_RAD = ANG * PI / 180.D0
RETURN
END

```

Prof. Luiz Paulo S. Fortes      Computação Aplicada à Cartografia II

58

## Introdução ao FORTRAN (15/16)

- Funções e subrotinas
  - Argumentos
    - Argumentos fornecem uma forma de comunicação entre módulos de programa
    - Argumentos são passados para subprogramas através de uma lista de argumentos e são recebidos pelo subprograma através de uma lista de argumentos
    - A lista de argumentos passada para uma função ou subrotina deve concordar com a lista de argumentos definida no subprograma em quantidade, ordem e tipo dos argumentos
    - Quando um dos argumentos for um ARRAY, o tamanho do ARRAY no subprograma deve ser menor ou igual ao tamanho no módulo de programa que chama o subprograma; além disso, caso o ARRAY não seja um vetor, é preciso garantir a correspondência entre os elementos (**em FORTRAN os argumentos são passados por referência e os ARRAYS armazenados por coluna na memória!**)

Prof. Luiz Paulo S. Fortes

Computação Aplicada à Cartografia II

59

## Introdução ao FORTRAN (16/16)

- Funções e subrotinas
  - Dimensionando ARRAYS como argumentos em subprogramas
    - ARRAYS ajustáveis
 

```
DIMENSION A (10, 20)
I = 10
J = 20
CALL SQUARE (A, I, J)

SUBROUTINE SQUARE (B, M, N)
 DIMENSION B (M, N)
```
    - ARRAYS que assumem o tamanho da variável passada
 

```
DIMENSION A (10, 20)
I = 10
CALL SQUARE (A, I)

SUBROUTINE SQUARE (B, M)
 DIMENSION B (M, *) ou DIMENSION B (M, 1)
```
  - No caso de vetor
 

```
DIMENSION B (1)
```

Prof. Luiz Paulo S. Fortes

Computação Aplicada à Cartografia II

60

## Introdução ao FORTRAN

- Bibliografia
  - <http://www.ibiblio.org/pub/languages/fortran/ch1-1.html/>
  - Open Watcom FORTRAN 77 Guides