

Trabalho 01
Estrutura de Linguagens
Ciência da Computação – UERJ
Isabelle Barbalho
FORTRAN

FORTRAN

(IBM Mathematical **FOR**mula **TRAN**slation System)

- Elaborado especificamente para aplicações científicas;
- Foi a primeira linguagem de programação imperativa. O primeiro compilador de FORTRAN foi desenvolvido para o IBM 704 em 1954-57 por uma equipe da IBM chefiada por John W. Backus.
- O FORTRAN foi um programa verdadeiramente revolucionário, antes dele todos os programas de computadores eram lentos, tendenciosos e originavam muitos erros.
- O Fortran permite a criação de programas que primam pela velocidade de execução. Daí reside seu uso em aplicações científicas computacionalmente intensivas como meteorologia, oceanografia, física, astronomia, geofísica, engenharia, economia etc.

Linha do Tempo (Resumida)

- 1977 - O FORTRAN 77 incluía muitas novas características, que permitiram escrever e guardar mais facilmente programas estruturados.
- 1990 - FORTRAN 90 trouxe uma novidade para o mundo das linguagens de programação: mecanismos para manipulação de arranjos.
- 2008 – Versão mais atual.

Pontos Fortes/Fracos

- A sintaxe da linguagem é considerada arcaica relacionada a linguagens modernas.
- É difícil de escrever um loop for.
- Erros na escrita de apenas um carácter podem levar a erros durante o tempo de execução em vez de erros de compilação.
- Fortran 95 já possui comandos muito breves para efetuar operações matemáticas com matrizes e disposições de tipos.
- Grande parte do código é reutilizável. Por exemplo, se alguém criou uma rotina data em Fortran e disponibilizado ao público através da Internet, que o código pode ser facilmente incorporado em seu programa Fortran.

Classificação

- Linguagem Imperativa
- Linguagem de Alto Nível
- Compilada
- Estática
- Destinada para Computação Científica



Fortran 2003 (ISO/IEC 1539-1:2004)[12] : Desde 1997, o FORTRAN vem passando por um novo processo de revisão. O FORTRAN 2003 irá incorporar um conjunto de novos comandos que permitirão, entre outras coisas, o controle de exceções e programação orientada a objetos.

Fonte: Wikipedia

Comparações

FORTRAN	C
Versões atuais possuem estruturas capazes de tornar o código mais fácil para entendimento.	O vasto conjunto de operadores acaba atrapalhando na legibilidade do código. Ex de símbolos que causam ambiguidades: “*”, “++”
Versões atuais comparadas com as anteriores possuem mais recursos que melhoram a redigibilidade mas a linguagem C ainda possui maior poder de legibilidade.	A linguagem C possui diversos recursos que facilitam na escrita, como por exemplo o operador + .
A expressividade está exposta principalmente nas funções intrínsecas, que tornam o código mais simples.	O operador ++ e operadores ternários ? : são bons exemplos de expressividade da linguagem C.
Aplicações Científicas	Programação de Sistemas

Códigos Representativos

1) A passagem dos parâmetros das funções e sub-rotinas é sempre por referência.

```
PROGRAM bad_call  
IMPLICIT NONE  
REAL :: x=1  
CALL bad_argument(x)  
END PROGRAM
```

```
SUBROUTINE bad_argument (i)  
IMPLICIT NONE  
INTEGER :: i = 0  
WRITE(*,*) ' I= ', i  
END SUBROUTINE
```

É uma característica boa para quem está aprendendo a programar. O aluno/pessoa não precisa se preocupar com essa parte que costuma ser difícil de entender para iniciantes.

Códigos Representativos

2) Se o parâmetro for um arranjo, então o ponteiro aponta para o 1º valor no array. Porém a sub-rotina precisa saber o local e o tamanho do array para garantir que ele fique dentro do limite do array e da execução das operações.

```
PROGRAM array

IMPLICIT NONE

INTEGER :: i
REAL, DIMENSION (5) :: a = 0
CALL sub (a, 5, 6)

DO i = 1, 6
    WRITE(*,100) i, a(i)
    100 FORMAT(1X, 'A(', I1, ') = ', F6.2)
END DO

END PROGRAM

SUBROUTINE sub (a, ndim, n)
IMPLICIT NONE
INTEGER, INTENT(IN) :: ndim
REAL, INTENT(OUT), DIMENSION(ndim) :: a
INTEGER, INTENT(IN) :: n
INTEGER :: i, j = 0

DO i = 1, n
    j = j + (2 * i)
    a(i) = j
    j = j + 3
END DO

END SUBROUTINE sub
```


Bibliografia

- Livro Fortran Estruturado (Fortran 77)

Harry Farrer, Christiano Gonçalves, Eduardo Chaves,
Frederico Ferreira Campos, Heilton Fábio, Marcos Augusto,
Miriam Lourença

- Links:

<https://pt.wikipedia.org/wiki/Fortran>

<http://www.eq.uc.pt/~yoshida3/Fortra.html>

<http://www.eq.uc.pt/~batateu3/introd.html>

<http://www.inf.ufes.br/~thomas/fortran/tutorials/helder/fortran.pdf>