Arquivos

- Um arquivo é um conjunto de dados organizados de um modo particular

- Podem ser mantidos em memória principal ou secundária

Arquivos

- open(descritor, file="nomeArquivo") → habilita o acesso ao arquivo descrito em "nomeArquivo" para uso através do descritor. Caso o arquivo ainda não exista ele será criado
- Após o nome do arquivo podem ser especificadas diversas opções como Action, que especifica o tipo de ação (leitura/escrita), Acces, que determina o tipo de acesso (sequencial/direto), entre vários outros

Arquivos

- Close(descritor) → fecha o arquivo indicado por descritor

- Este comando informa que o arquivo associado ao descritor não é mais necessário

Exemplo

```
program testeArqv
implicit none
real::a,b,c
a=10.0/3.0
b=sqrt(2.0)
c=sin(0.645)
open(10,file="data.txt")
write(10,*)a,b,c
close(10)
end program testeArqv
```

Exercício

- Faça um programa que salve em um arquivo os valores de um vetor

Exercício

- Faça um programa que salve em arquivo os elementos de uma matriz. O nome do arquivo e os valores da matriz devem ser informados pelo usuário

exercício

- Faça um programa que leia de um arquivo os valores de um vetor. O nome do arquivo deve ser informado pelo usuário

Exercício

- Faça um programa que leia uma matriz no formato descrito abaixo

1 2 3

4 5 6

7 8 9

Formatos

- Muitas vezes é necessário formatar a saída de modo somente alguns dígitos sejam considerados

- Exemplo: pi = 3.141592

- Deseja-se imprimir apenas 3.14

Exemplo

```
program precisao
real::b
b=sqrt(5.0)/1000
write(*,*)b
write(*,"(f7.5)")b
write(*,"(e10.3)")b
end program precisao
```

- "(f7.5)" → formato de ponto flutuante f com 7 dígitos totais (. e sinal são contados) sendo 5 casas decimais

- "(e10.3)" → formato de notação científica com 10 digitos totais (. , sinal e o “e” são contados) sendo 3 para as casas decimais

- "(I10.5)" → formato de inteiros i com 10 dígitos totais (sinal é contado) sendo 5 digitos no mínimo
- “(E20.7E2)” → Notação científica com total de 20 dígitos (0, .,e,sinais contam), sendo 7 para o número normalizado e 2 para o expoente

- Pi=3.141592
- Write(*, “(E12.5)”)pi
- Write(*, “(E12.3E4)”)pi
- Write(*, “(E12.7E1)”)pi

Variantes

- ES → similar ao E só que com pelo menos uma casa inteira

- EN – Similar ao E, expoentes são sempre múltiplos de 3

- “(L3)” → Descritor lógico que indica que o valor lógico deve ser escrito com 3 dígitos

- “(A3)” → Descritor de texto que indica que o texto deve ser escrito usando 3 caractéres

Múltiplos Padrões

- Mais de um padrão pode ser usado em um formato

- Exemplo: write(*, “(I5,f7.3)”) a,b

  imprime a usando I5 e b usando f7.3

- Separar os múltiplos padrões por virgulas

Repetição de padrões

- “(2I5)” → equivalente à “(I5,I5)”

- “(I5,2f7.3,2I5)” → “(I5,f7.3,f7.3,I5,I5)”

Controle horizontal

- X → indica um espaço em branco
- rX → r espaços em branco
- Exemplo

```
INTEGER :: a
REAL :: b
CHARACTER(1) :: c
READ(*,"(2X,I4,3X,F5.2,2X,A)")  a, b, c
write(*,*)a,b,c
```

Tabulação

- tc → salta para a posição c
- tlc → movimenta c casas para a esquerda
- trc → movimenta c casas para a direita

Controle vertical

- / e r/
- Entrada: A linha atual é pulada e o restante das informações não lidas ignoradas. O processo de leitura recomeça na primeira posição da próxima linha
- Saída: A linha atual é impressa e o próximo item é impresso na próxima linha
- Virgulas não são necessárias para separar múltiplos caracteres. Ex: //=/,/

Exercício

- Faça um programa que leia um vetor a partir de um arquivo e imprima um relatório de acordo com o formato abaixo

```
 Pos  valor
=== ======
  1 100.00
  2 231.00
  ...
 14 250.00
 15 379.00
Media =   1.7860001E+02
```

```
...
Average = Average / ActualSize
WRITE(*,"(A, A)")  " ", " Pos  Valor "
WRITE(*,"(A, A)")  " ", "=== ======"
DO i = 1, size
   WRITE(*,"(I4, F7.2)")  i, vet(i)
END DO
WRITE(*,*)
WRITE(*,"(A,A,ES15.7)")  " ", "Media = ", Average
...
```