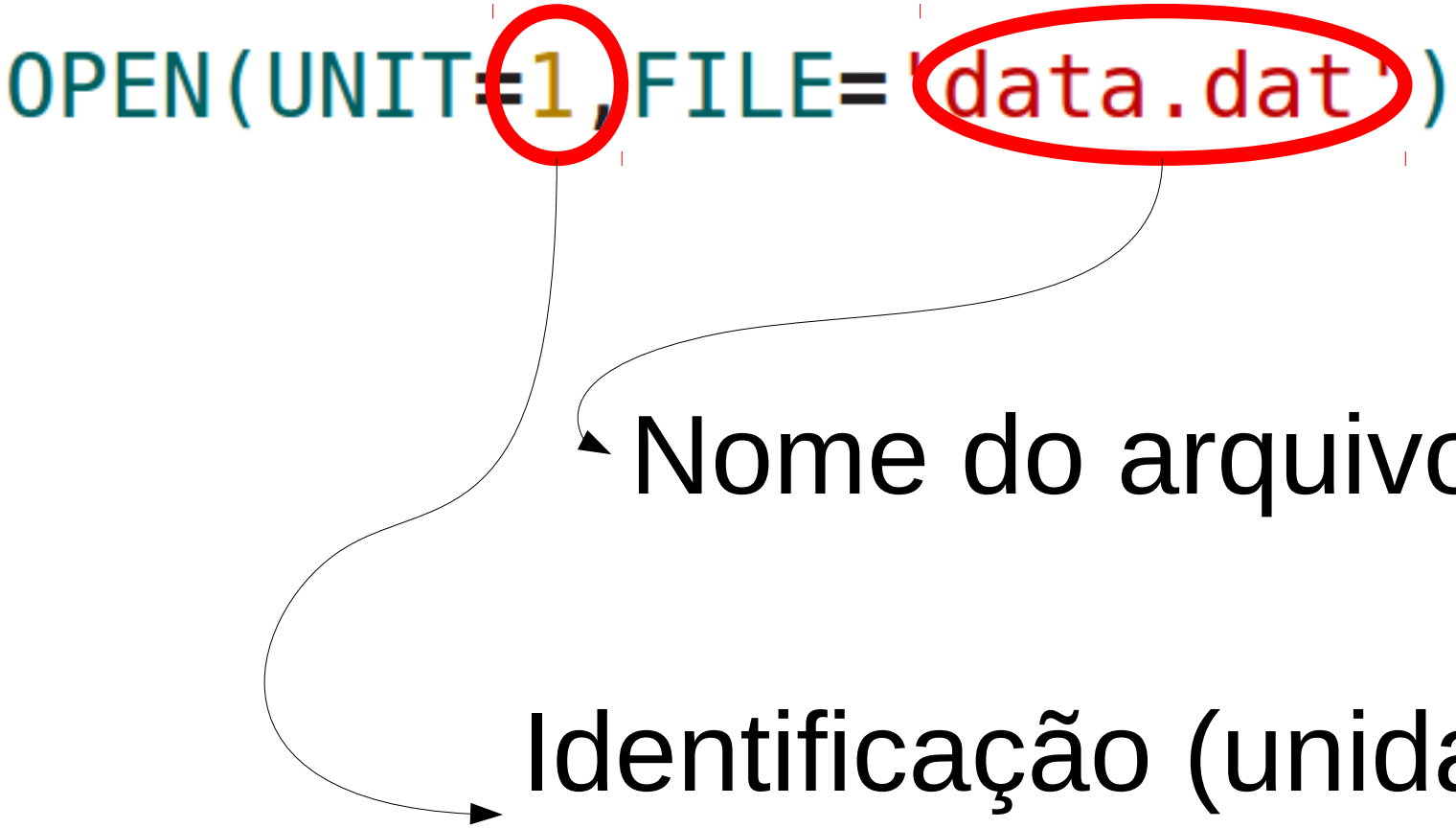
Métodos Computacionais em Física

Aula 06

Input/output: trabalhando com arquivos e formatando dados

Como abrir um arquivo em FORTRAN

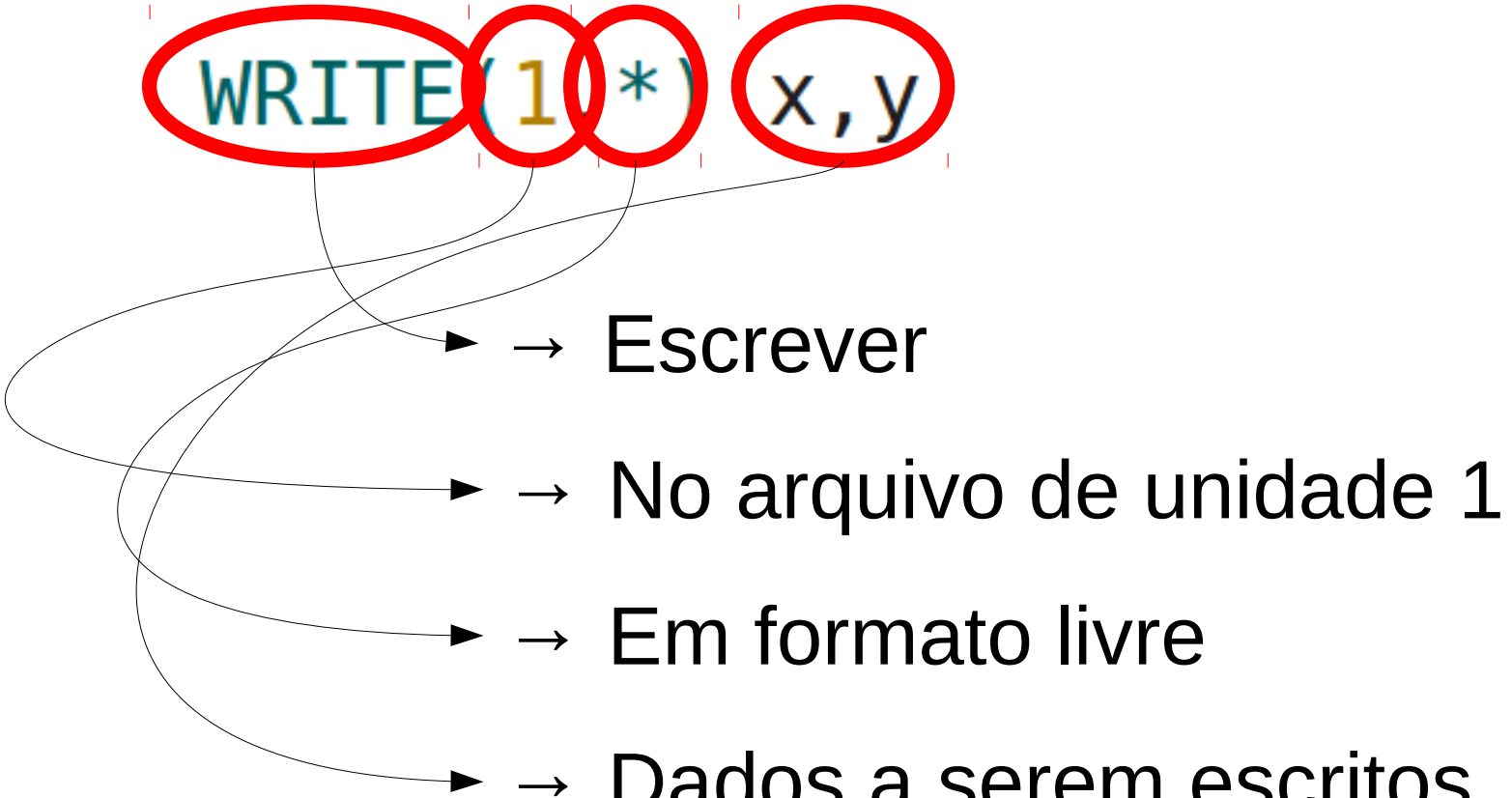


Por enquanto ...

Evite utilizar as unidades 5 e 6!!!!

Escrevendo em um arquivo

```
OPEN(UNIT=1,FILE='data.dat')
WRITE(1,*) x,y
```



Lendo a partir de um arquivo

```
OPEN(UNIT=1,FILE='data.dat')
READ(1,*) x,y
```

→ Ler

→ No arquivo de unidade 1

→ Em formato livre

→ Dados a serem lidos

Fechando um arquivo

```
CLOSE(UNIT=1)
```

```
PROGRAM print_function

IMPLICIT none
REAL:: x,y
INTEGER::i

OPEN(UNIT=1,FILE='data.dat')
DO i=1,5
   READ(1,*) x,y
   PRINT*,'f(',x,')=',y
END DO
CLOSE(UNIT=1)

END PROGRAM print_function
```

```
PROGRAM write_function

IMPLICIT none
REAL:: x,y
INTEGER::i

OPEN(UNIT=1,FILE='data.dat')
DO i=1,5
  x=2.0*i
  y=x*x
  WRITE(1,*) x,y
END DO
CLOSE(UNIT=1)

END PROGRAM write_function
```

Especificadores

- UNIT
- FILE
- STATUS
- FORM
- ACTION
- POSITION
- IOSTAT

UNIT

- Determina a unidade do arquivo

FILE

- Determina o nome do arquivo

STATUS

Opção que determina o *status* do arquivo e e igualado a uma das seguintes expressões de caracteres (a serem colocadas entre aspas ou apóstrofos):

- OLD
- NEW
- REPLACE
- SCRATCH
- UNKNOWN

STATUS='OLD'

O arquivo deve ser pré-existente para que seja aberto com sucesso

STATUS='NEW'

O arquivo deve ser inexistente para que ele seja aberto com sucesso. Após aberto, seu *status* passa a ser OLD

STATUS='REPLACE'

Se o arquivo já existir, ele será deletado antes de ser aberto um novo arquivo (que após a abertura passará a ser um OLD). Caso o arquivo não exista, o REPLACE terá o mesmo efeito de NEW.

STATUS='SCRATCH'

O arquivo a ser utilizado será temporário e será apagado quando o mesmo for fechado.

STATUS='UNKNOWN'

É o mesmo que não especificar o status (valor default). O efeito será OLD se arquivo for pre-existente, ou NEW se o arquivo for inexistente.

FORM

Opção que determina se o arquivo é formatado ou não. No caso de arquivo não formatado, os dados serão escritos em código binário. As opções são:

- FORMATTED
- UNFORMATTED

ACTION

Opção que determina as ações que podem ser feitas no arquivo:

- READ → o arquivo será tratado como somente leitura
- WRITE → não será possível ler a partir desse arquivo
- READWRITE → será possível ler e escrever no arquivo

POSITION

Opção que determina a posição a partir da qual iremos ler ou escrever no arquivo. Se arquivo for novo, esse opção será ignorada:

- REWIND → o ponto de leitura/escrita será posicionado no início do arquivo.
- APPEND → o ponto de leitura/escrita será posicionado após o último registro do arquivo.

IOSTAT

Esta opção não deve ser igualada a uma expressão caractere, mas a uma variável inteira. Se o arquivo for aberto com sucesso, o valor atribuído à variável inteira será 0 (zero) ou um número diferente de zero no caso contrário.

Reposicionando o ponto de leitura e escrita

BACKSPACE(UNIT=1)

- Volta uma linha no arquivo de unidade 1

REWIND(UNIT=1)

- Volta ao início do arquivo de unidade 1

AVISO

- Escrever dados em um ponto de um arquivo apaga todos os dados possivelmente existentes no arquivo após este ponto!!!

Unidade 5

- Utilizada para arquivo de entrada especificado na linha de execução.

```
PROGRAM print_function

IMPLICIT none
REAL:: x,y
INTEGER::i

DO i=1,5
  READ(5,*) x,y
  PRINT*,'f(',x,')=',y
END DO

END PROGRAM print_function
```

```
edu@pegaso:~$ ./print_function.x < data.dat
```

Unidade 6

- Utilizada para arquivo de saida especificado na linha de execução.

```
PROGRAM write_function

IMPLICIT none
REAL:: x,y
INTEGER::i

DO i=1,5
  x=2.0*i
  y=x*x
  WRITE(6,*) x,y
END DO

END PROGRAM write_function
```

```
edu@pegaso:~$ ./write_function.x > data.dat
```

```
PROGRAM write_function

IMPLICIT none
REAL:: x,y
INTEGER::i

DO i=1,5
  READ(5,*) x
  y=x*x
  WRITE(6,*) x,y
END DO

END PROGRAM write_function
```

```
edu@pegaso:~$ ./write_function.x < data1.dat > data1.dat
```

Formatando dados

Troque o * em

```
WRITE(1,*) x,y
```

Por '(…)' onde o … é um especificador de formato

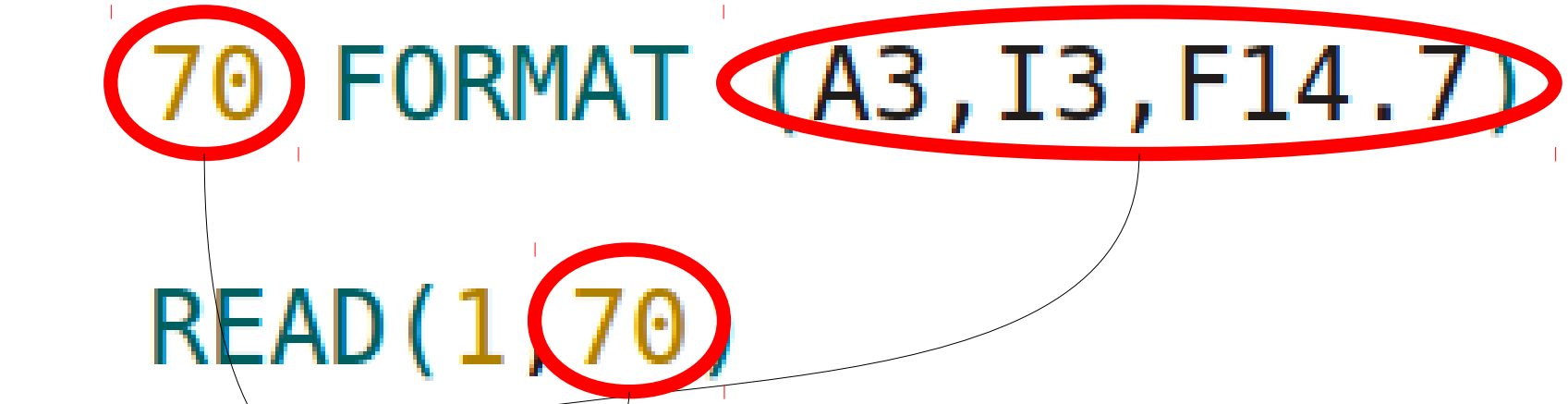
Especificadores de Formato

Descriptor	*Meaning*
`I`*w*	Output an integer in the next *w* character positions
`F`*w*`.`*d*	Output a real number in the next *w* character positions with *d* decimal places
`E`*w*`.`*d*	Output a real number in the next *w* character positions using an exponent format with *d* decimal places in the mantissa and four characters for the exponent
`A`*w*	Output a character string in the next *w* character positions
`A`	Output a character string, starting at the next character position, with no leading or trailing blanks
`L`*w*	Output *w* − 1 blanks, followed by `T` or `F` to represent a logical value
n`X`	Ignore the next *n* character positions

Exemplos

```
WRITE(1,'(I3,F14.7)') i,x

WRITE(1,'(I3,I3)') i,j

WRITE(1,'(2I3)') i,j

WRITE(1,'(A,I3,F14.7)') 'fisica',i,x

WRITE(1,'(A3,I3,F14.7)') 'fisica',i,x
```

Comando FORMAT



→ Label (rótulo)

→ Especificação do formato

→ utilizando o formato

Hands on

- 9.2
- 9.5
- 9.7
- 9.8