Input/output in Fortran

Victor Eijkhout

0th Edition, 2018

Types of I/O

- Formatted vs unformatted
- Default formatting vs specified format

I/O commands

- `Print` simple output to terminal
- `Write` output to terminal or file (‘unit’)
- `read` input from terminal or file
- `Open`, `Close` for files and streams
- `Format` format specification that can be used in multiple statements.

Implicit do loops

Parametrized printing with an *implicit do loop*:

```
print *,( i*i,i=1,n)
```

Array printing

- `print *,A` prints whole array, column-major
- Implicit do loops:

  ```
  print *,( A(i,i),i=1,n)
  ```

  Can also be nested.

Formats

- Fine control of input/output.
- Direct use in print statement:

  ```
  print '(a6,3f5.3)',"Result",x,y,z
  print '("Result:",3f5.3)',x,y,z
  ```
- Format statement:

```
  print 10,"result:",x,y,z
10 format('(a6,3f5.3)')
```

Format specifiers

- '`a`*n*' specifies a string of *n* characters. If the actual string is longer, it is truncated in the output.
- '`i`*n*' specifies an integer of up to *n* digits. If the actual number takes more digits, it is rendered with asterisks.
- '`f`*m.n*' specifies a fixed point representation of a real number, with *m* total positions (including the decimal point) and *n* digits in the fractional part.
- '`e`*m.n*' Exponent representation.
- Strings can go into the format:

  ```
  print '("Result:",3f5.3)',x,y,z
  ```
- '`x`' for a space, '`/`' for newline

Format repetitions

```
print '( 3i4 )', i1,i2,i3
print '( 3(i2,":",f7.4) )', i1,r1,i2,r2,i3,r2
```

Repeats and line breaks

- If `abc` is a format string, then `10(abc)` gives 10 repetitions. There is no line break.
- If there is more data than specified in the format, the format is reused in a new print statement. This causes line breaks.
- The / (slash) specifier causes a line break.
- There may be a 80 character limit on output lines.

Exercise 1

```
print '(10(i2,1x))',( (10*(i-1)+j-1,j=1,10), i=1,10 )
```