

Fortran90 プログラミング

目的

プログラムの作成から実行までの過程を理解する

準備

Linux にログイン、端末とテキストエディタ(gedit)を開く

プログラムの作成

1. ディレクトリの作成

プログラムを保存しておくため、ディレクトリを作成する。

```
mkdir directoryname
```

directoryname のところは適当な名前を付ける。mkdir はディレクトリを作るコマンドで、make directory の略である。GUI 側でも新しいディレクトリが出来ているのを確認してみよう。ディレクトリを消すには

```
rmdir directoryname
```

と入力する。なお、ディレクトリの中身が空でないと rmdir によりディレクトリを消すことは出来ない。もし、ディレクトリを消してしまったら、もう一度ディレクトリを作っておこう。

2. ディレクトリの移動

ディレクトリを移動する

```
cd directoryname
```

と入力すると移動できる。ディレクトリの名前を

```
pwd
```

と入れて確認する。(復習であるが、1つ上の階層に戻るには、

```
cd ..
```

と入れる。

```
cd
```

あるいは

```
cd ~
```

と入力するとホームディレクトリへ戻る。“~(ティルダ)”はホームディレクトリを意味する。“~nakakuki”とするとアカウント nakakuki のホームディレクトリという意味になる。戻ってしまったら、再び先ほど作ったディレクトリのなかへ入っておく。)

3. エディタによる Fortran プログラムファイルの作成

gedit &

と入力して GNOME Editor をバックグラウンドプロセスとして起動する。

問題 F1

- (1) プログラムを入れておくための適当な名前ディレクトリを作成せよ。
- (2) 別紙の別紙の FORTRAN プログラムを gedit を使って入力し、(1)で作成したディレクトリの中に“hello.f90”という名前で保存せよ。

Fortran90 プログラムのファイルの名前には、名前の後ろに拡張子“.f90”を付ける。拡張子“.f90”があるとシステムがそのファイルが自由形式の Fortran90 のプログラムであることを認識する。Fortran の固定形式を使う場合には“.f”とする。

8. プログラムのコンパイル

保存したら、端末ウィンドウの方に戻って、

f95 *prognam.f*

と入力する。“*prognam*”は先ほど保存したときに付けた名前である。この作業をコンパイルという。コンパイルは Fortran プログラムを CPU の命令の並び(機械語と言う。2進法の数字の羅列で書かれている)に直し、実行形式ファイル(アプリケーションファイル)を作成する作業である。

ちゃんとコンパイル出来たであろうか? エラーメッセージが出なければ、実行可能ファイル(executable)*、つまりアプリケーションのファイルが出来ているはずである。

ls -l

で確認する。a.out というファイルが出来ているはずである。出来ていなければ、プログラムに何らかの誤りがあつて出来ていない。この場合はエラーメッセージと元のプログラムをよく見てプログラムを直す。プログラムの誤りをバグ(bug: 虫)といい、誤りを直すことをバグ取り(debug)と言う。

* 実行形式ファイルとも言う

8. プログラムの実行

いよいよプログラムを実行してみる。

```
./a.out
```

と入力する。あらかじめインストールされているアプリケーションと違って、OS が実行可能ファイルの場所を知らないなので、カレントディレクトリ(“.”)の中に(“/”: ディレクトリの区切り記号)ある a.out と入力してやらないといけない。

```
Hello, World!
```

と出力されればプログラムは正しく記載されている。

問題 F2

プログラムをコンパイル・実行せよ。

問題 F1・F2 プログラム

```
!2345678901234567890223456789032345678904234567890523456789062345678907234567890
!  
! Hello World  
!  
program hello  
    write (6,*) 'Hello, World!'  
    stop  
end program hello
```

プログラムの構造

プログラムはだいたい次のような構造になっている。

program 文→宣言文→入力文→実行文→出力文→end 文

どうしてこうなっているのかは、実際に計算機に何をさせるのかを考えれば分かるだろう。

入力文はデータを入れるための部分であり、実行文は計算処理をさせる部分である。計算処理をさせたら、結果を出させるのが出力文である。宣言文については後日説明する。

今回のプログラムは画面に文字を出力するだけなので、宣言文～実行文の部分がない。

プログラムの説明

```
!2345678901234567890223456789032345678904234567890523456789062345678907234567890
```

今回の授業で紹介するプログラムは自由形式を使う。横の文字数は 132 文字まで書いても大丈夫だが、80 文字程度にするのが見やすいだろう。

!

で始まる行はコメント行である。処理は何も行わない。つまり、プログラムの注釈であり、内容の説明や使い方などを書いておく。

```
program hello
```

program 文：プログラムに hello という名前を付ける(実質何もしない)。省略可で、そのときは自動的に main という名前になる。

```
write (6,*) 'Hello, World!'
```

write 文：変数の値を書く。quotation mark の間の文字はそのまま出力する。

(6,*)の6は出力先の装置番号。6は画面を表す。それ以外の番号だとファイルに出力を保存する(昔のコンピュータではプリンタなどの周辺機器も番号がついていた名残)。

*はフォーマットの指定で、*の場合、デフォルト出力。

stop

stop 文。実行をここで停止。

end program hello

end 文。プログラム単位の終わりを表す。単に end だけでも良い。

Fortran の固定形式と自由形式

Fortran には固定形式と自由形式がある。Fortran77 の頃は固定形式を使用するのが多かったが、今では、自由形式が良く用いられている。固定形式と自由形式の違いには以下のようなものがある。

共通なこと

プログラム内の大文字と小文字を区別しない

スペースは無視している

異なること

拡張子

固定形式 .f

自由形式 .f90

1 行内での書き方

固定形式 7 文字目から 72 文字目 1 から 6 文字目までは特別な用途がある。

自由形式 自由(どこに書いても良い)

コメント行

固定形式 c または!で始まる。C は先頭のみを使用。

自由形式 !のみを使う

文番号

固定形式 2 文字目から 5 文字目

自由形式 実行文の前

継続行

固定形式 続ける行の 6 文字目に何か文字を書く(0 以外)。1 行目は何も書かない。

自由形式 後ろに&を書くと次の行へ続く。最後の行は何も書かない。

自由形式は、フォーマットが自由な分、規則が分かりにくく、また、読みにくいプログラムを書かないように注意が必要である。気象庁では、数値予報のプログラムに Fortran を用いている。多くの人を読みやすいプログラムとなるようプログラミングにルールをもうけているので、これを参考にすると良いだろう。

Fortran 標準コーディングルール (気象研究所)

<http://www.mri-jma.go.jp/Project/mrinpd/coderule.html>

Fortran の参考書

富田博之・齋藤泰洋「Fortran90/95 プログラミング」、培風館、2011 年。