

# Linux 利用の手引き (Fortran 編) コンパイル・実行・デバック

2007 年 12 月  
学術情報センター

## 目次

1	概要.....	2
2	Fortran 言語.....	2
3	言語処理.....	2
3. 1	エディタ (プログラム作成) .....	2
3. 2	翻訳 (コンパイル) .....	3
3. 3	実行.....	3
3. 4	エラーの場合.....	3
4	リダイレクションによる実行 .....	4
5	ライブラリの連結 .....	5
5. 1	ライブラリーリンクエラー.....	5
6.	レポート作成時の注意.....	5

## 1 概要

学術情報センターの実習室に導入されているUNIX(Linux)の利用方法をFortran言語のプログラミングを用いて説明します。とくにセンターで特別に変更している点を中心に説明しますので、UNIX(Linux)やFortran言語の詳細は市販の本を参考にして下さい。なお、Linux手引き(基本編)にはログイン・コマンド・エディタ・ファイル等が記載されていますので該当する部分を参考して下さい。

## 2 Fortran 言語

数値計算用のプログラミング言語で広く利用されており、本学でも工学部の研究室には昔から利用されている。数値計算分野では機械工学・土木工学等の構造計算はライブラリがFortranから利用するケースが多く、処理時間がかかるものもある。また、計算サーバのSun(solalis)システムではコンパイラも複数用意してありますので適材適所で活用してください。また、センターで準備したツール等も紹介します。

## 3 言語処理

LinuxのFortran言語はGCCから派生して作成されたもので、オープンソースの世界で定評のあるコンパイラが利用できます。エディタもVIやEmacsの従来からのエディタに加えGEDITとよぶWindowsシステムのメモ帳ライクなエディタも準備されています。統合化された開発システムはありませんがシンプルな理解し易いコマンドがあります。ここでFortran言語のプログラムを処理する流れを説明し、機能や目的を解説します。

### 3. 1エディタ(プログラム作成)

fortran言語のプログラムをエディタを利用して入力します。

```

c234567890 sample program EX99001 chubu tarou
integer i,j,k
read(5,*)i,j
k = i + j
write(6,*)"i=",i," j=",j," k=",k
stop
end

```

エディタはViまたはEmacsでも同様です。

補足 gedit はアプリケーションメニュー→システムツールから実行できます。

### 3. 2 翻訳 (コンパイル)

エディタを利用して入力されたプログラムはソースコードとも呼ばれて、人間にとって理解し易い形式で記述されています。このソースコードを計算機 (機械) に処理できる形式 (機械語) に変換することを翻訳 (コンパイル) と呼びます。

例 sample.f をコンパイルする場合

```
a00% f77 sample.f
```

### 3. 3 実行

エラーがなかった場合はコンパイル結果として機械語のプログラムが a.out として出力されます。出力された a.out を実行するにはファイル名 a.out をそのままコマンドと同様に入力します。

```
a01% a.out
1 2
i= 1 j= 2 k=3
a01%
```

なお、実行時にエラーが発生することもあります。これは 0 で割り算をして、計算できない場合や途中で異常終了する場合などです。このエラーを実行時のエラーと呼びます。

### 3. 4 エラーの場合

Fortran 言語プログラムに何らかのエラーがある場合はコンパイル時にエラーメッセージを表示して機械語の実行ファイル ( a.out ) を出力しない場合があります。この場合にはエラーメッセージを読んで該当する Fortran 言語プログラムの部分を修正します。修正にはエディタを利用し、その後、翻訳し直します。

例 \*\* の計算式が無効である場合

```
a01% f77 sample.f
sample.f: In program `MAIN__':
sample.f:4:
      k = i % j
           ^
Invalid form for assignment statement at (^)
a01% cat -n sample.f
 1 c234567890 sample program EX99001 chubu tarou
 2 integer i, j, k
 3 read(5,*) i, j
 4 k = i % j
 5 write(6,*)"i=", i, " j=", j, " k=", k
 6 stop
 7 end
a01%
```

4 行目の計算式がおかしい。% は Fortran 言語の文法上で許されない。

### 実行時のエラー例

```

a01% f77 sample.f
a01% a.out
1 0
Floating Exception
アボート
engpc04% cat -n sample.f
   1 c234567890  sample program  EX99001 chubu tarou
   2         integer i, j, k
   3         read(5, *) i, j
   4         k = i / j
   5         write(6, *) "i=", i, " j=", j, " k=", k
   6         stop
   7         end
a01%
a01%

```

入力データが適当でなく '0' で割算を実行しようとした。

## 4 リダイレクションによる実行

UNIX ではシェルの機能を利用して入出力の切替えが容易にできます。入力するデータをファイルから読んだり、画面表示する出力をファイルに書き出すことが簡単にできる特徴を持っています。

例 input.d から入力する場合。（あらかじめ、エディタで入力データを作成してあること）

```

a01% a.out < input.d
i= 1 j= 2 k= 0
a01%

```

例 output.d に出力する場合。

```

a01% a.out > output.d
1,2
a01% cat output.d
i= 1 j= 2 k= 0
a01%

```

例 両方する場合。

```

a01% a.out < input.d > output.d
a01%

```

## 5 ライブラリの連結

Fortran 言語から PS ライブラリ (図形作成) を連結するにはつぎのように行ないます。

```
a01% f77ps sample.f
```

### 5. 1 ライブラリーリンクエラー

連結する関数名を間違えるとライブラリーエラーが発生し、実行形式が作成されない。また、リンクオプションを忘れるても同様のエラーが発生します。

例 関数名を間違えた場合

```
a01% f77 sample.f
/tmp/cc0yRhcb.o: In function `MAIN__':
sample.f:(.text+0x73): undefined reference to `plotx_'
collect2: ld はステータス 1 で終了しました
a01%
```

## 6. レポート作成時の注意

課題提出する場合、C プログラムと実行結果をまとめてファイルを作り、それに考察や感想を追加するときには以下のように行います。

実行結果のファイル ( kekka ) 作成

```
a01% a.out > kekka
a01%
```

プログラム ( program1.f ) と実行結果 ( kekka ) を 1 つのファイル ( report09 ) を作成する。

```
a01% cat program1.f kekka > report09
a01%
```

作成した ファイル ( report09 ) に Emacs エディタなどを利用して考察や感想を追加して、レポートを作成して下さい。

印刷

```
a01% flp report09
a01%
```