

# FORTRAN (と C) によるプログラミング

## 補遺 1 emacs の使い方 (1)

すでに存在するファイル sample.f を開くときは、まず、emacs を起動します。

```
/home/snaoki> emacs &
```

最後に"&"をつけてください。つけ忘れると emacs が終了するまでターミナルがふさがってしまいます。そのあと、C-x C-f でファイル名 sample.f を指定して開きます。emacs を起動するときに、

```
/home/snaoki> emacs sample.f &
```

とすれば、はじめからファイルを開いた状態で emacs を起動できます。C-x C-c で emacs を終了します。

## 補遺 2 emacs の使い方 (2)

prog02\_1.f という名前の新しいファイルを作成することにします。まず、emacs を起動します。

```
/home/snaoki> emacs &
```

次に、C-x C-f でファイル名 prog02\_1.f を指定して開きます。存在しないファイルが指定されたときは、新ファイルとして作成されます。ファイルの作成が終わったら、C-x C-s でファイルを保存します。名前を変えて保存したいときは、C-x w とします。最後に、C-x C-c で emacs を終了します。

## 補遺 3 ディレクトリを作成

prog03\_1.f の作成が終わったら、一度 emacs を閉じて、ls コマンドでファイルの一覧を確認します。

```
/home/snaoki> ls  
a.out* prog02_1.f prog02_2.f prog03_1.f sample.f
```

ファイルが増えてきたので、ディレクトリ (フォルダ) を作って整理します。ディレクトリを作るときには、mkdir というコマンドを使います。prog03 という名前のディレクトリを作るときには、

```
/home/snaoki> mkdir prog03
```

とします。

## 補遺 4. 1 emacs の使い方 (3)

prog04\_1.fの中に似た内容の行が繰り返している箇所があります。このようなとき、**emacs**上で、まず1行目を作成し、そのあとで、その行をコピーして書き替えたような効率的な場合があります。コピーしたい行の最初に移動し、**C-k**を2回実行してください。その行が消去されます。実はこのとき、消去された内容がメモリに記憶されています。**C-y**とすると、消去した行が復活します。さらに、別の行にカーソルを移動して、再び**C-y**とすると、消去した行が挿入されます。**C-k**を4回実行すれば2行まとめて消去されますが、このようにすれば、2行まとめてコピーすることができます。

**emacs**上ではコントロールキーを使ってさまざまな処理をすることができますが、間違った操作をしたときには、**C-q**によって操作を取り消せる場合があります。

## 補遺4. 2 ターミナルの操作

ターミナル上で、同じコマンドを何度も使うことがあります。このようなとき、何度も同じコマンドをキーボードから入力するのは面倒です。ターミナル上で、**C-p**とすると、ひとつ前に実行したコマンドを表示させることができます。複数回実行すれば、その分だけ前のコマンドが現れます。自分が再度実行したいコマンドを表示された状態で**Enter**キーを押せば、そのコマンドが実行されます。**C-p**の代わりに上矢印キーでも同様の操作が可能です。

ファイル名を途中まで入力して段階で、ファイル名を同定できる場合には、**Tab**キーを押すと補完します。たとえば、「f77 sample.f」と入力したいとき、「sam」で始まるファイルがsample.fだけであれば、「f77 sam」まで入力して**Tab**キーを押せば「f77 sample.f」になります。

無限に処理を反復し終了しないプログラムを誤って作成、実行してしまうことがあります。このような場合、ターミナルにプロンプトが返ってこないのので、新たなコマンドを入力することができません。このようなときには、**C-c**で強制終了します。

## 補遺5 許可モード

作業ディレクトリに、入力ファイルdata05.txtをコピーしてきたとします。誤って入力ファイルに書きこみをしたくないように、このファイルを書きこみ禁止にしておきます。はじめに、lsコマンドに-lオプションをつけて実行し、許可モードを確認します。

```
/home/snaoki/prog05> ls -l
-rw-r--r-- 1 snaoki snaoki 209 Sep  2 2009 data05.txt
```

「rw-」は読みこみ(r)は許可、書きこみ(w)も許可、実行(x)は不許可(そもそも実行ファイルではない)を意味します。ここが、「rwx」となっていたら、読みこみも書きこみも実行もすべて許可、「r--」となっていたら、読みこみは許可、書きこみと実行は不許可を意味します。2~4文字目で自分に対する許可モード、5~7文字目で同じグループのユーザに対する許可モード、8~10文字目で他のユーザに対する許可モードを表しま

す。通常のテキストファイルであれば、自分に対しては「rw-」、同じグループのユーザや他のユーザに対しては「r--」とすることが多いです。ファイルの所有者であれば、許可モードを変更することができます。r = 4、w = 2、x = 1として、許可モードを数値で表し、自分、同じグループのユーザ、他のユーザに対する数値を並べて3ケタの数にします。たとえば、「rw-r--r--」であれば644です。自分に対しても書きこみ禁止にして「r--r--r--」にすれば444です。許可モードを変更するときには、chmod コマンドを使って、数値で指定します。たとえば、data05.txt の許可モードを「r--r--r--」(444)にする場合は、

```
/home/snaoki> chmod 444 data05.txt
```

とします。再び、許可モードを確認すると、

```
/home/snaoki/prog05> ls -l
-r--r--r-- 1 snaoki snaoki 209 Sep  2  2009 data05.txt
```

となっているはずです。

## 補遺6 バックアップ

ディレクトリの内容のバックアップを作成するときには、以下のようにします。まず、ホームディレクトリに移動します。

```
/home/snaoki> ls
prog03/ prog04/ prog05/ prog06/
```

ここで、ディレクトリ prog06 のバックアップを作成することにします。バックアップを作成するには、以下の例のように tar コマンドを使います。「cvf」は、それぞれ、tar ファイルを生成する、途中経過を表示する、ファイル名を指定する、という意味のオプションです。この通りに入力してください。オプションの後に、バックアップファイル名 (tar ファイル名)、バックアップするディレクトリやファイルの名前を書きます。バックアップするディレクトリやファイルは、複数を列挙することもできます。以下の例では、prog06 の中身を backup.tar という名前の tar ファイルにバックアップします。

```
/home/snaoki> tar cvf backup.tar prog06
/home/snaoki> ls
prog03/ prog04/ prog05/ prog06/ backup.tar
```

次に、backup.tar を圧縮します。圧縮することによってファイルのサイズを小さくすることができます。実行したら、ファイル backup.tar が backup.tar.gz に変わっていることを確かめてください。

```
/home/snaoki> gzip backup.tar
/home/snaoki> ls
```

```
prog03/ prog04/ prog05/ prog06/ backup.tar.gz
```

バックアップファイルを展開して、内容を復元するためには、まず、

```
/home/snaoki> gunzip backup.tar.gz
```

として圧縮を解きます。この段階で、backup.tar.gz は backup.tar に戻っているはずですが、次に、

```
/home/snaoki> tar xvp backup.tar
```

として tar ファイルを展開します。このとき、同じファイルが存在すると上書きされることに注意が必要です。xvf でも展開できますが、p オプションをつけると許可モードなどのファイルの属性を保存します。