

# UNIXワークステーションにおけるFORTRANの実行方法の紹介

— 平成10年度リプレースシステム —

総合情報処理センター 松本 豊司

平成10年度のリプレースで新たに導入されたワークステーションでFORTRANを使用できるのはcs1,cs2,ws01~ws48の各ワークステーションです。なお、教育実習ユーザが使用できるのは教育用ワークステーションws01~ws48のみです。

表1 FOTRAN 実行可能な新規導入ワークステーション一覧

| ホスト名      | IP アドレス                       | 役割             | その他                                 |
|-----------|-------------------------------|----------------|-------------------------------------|
| Cs1       | 133.28.250.102                | フロントエンド<br>サーバ | ソースファイルの作成、デバッグ、<br>コンパイル、実行、バッチの依頼 |
| cs2       | 133.28.250.103                | 計算サーバ          | バッチの実行専用                            |
| ws01~ws48 | 133.28.21.11<br>~133.28.21.58 | 教育用 UNIX 端末    | ソースファイルの作成、デバッグ、<br>コンパイル、実行        |

これらのワークステーションにインストールされているFortranはFORTRAN 90です。

## 1. FORTRAN の実行のための環境ファイルの設定

.chsrc あるいは.login ファイルに以下のような記述があるか確認ください(C シェルを使用している場合)。本年度新たに登録されたほとんどのユーザは予め設定されていますので必要はなく、継続利用のユーザの方は下記を参考に是非お確かめください。

```
setenv LD_LIBRARY_PATH /opt/FSUNf90/lib
```

```
setenv MANPATH /opt/FSUNf90/man
```

また、パスに/opt/FSUNf90/bin が含まれているか確認してください。.chsrc ファイルのパスの設定部分に以下のように/opt/FSUNf90/bin が含まれていれば問題ありません。

```
set path=( /opt/FSUNf90/bin $path ..... )
```

本年度新規登録された方ならほとんどの方がこれらの設定が登録時にされています(4月当初に登録された方は、この設定がない可能性があります)。

何らかの事故で.chsrc ファイルや.login ファイルが壊れたり、なくなった方あるいは初期状態に戻したい

いは以下のコマンドを fs1,cs1,cs2,ws01～ws48 にログインし、実行してください。新たに導入されたワークステーションにおいて必要と思われる環境変数を設定した.cshrc,.login ファイルが上書きコピーされます。  
《注意1》このコマンドを実行しますと各自のホームディレクトリーに既にある環境ファイルに上書きをしますので、今使っている環境ファイルを保存したい場合はこのコマンドを実行しないで環境変数の設定を行ってください。

OpenWind で使用する場合

```
{ws1} 1% /usr/local/bin/envset
```

```
{ws1} 2% source .cshrc
```

Xウィンドで使用する場合

```
{ws1} 1% /usr/local/bin/envset.X
```

```
{ws1} 2% source .cshrc
```

今使っている環境ファイルに環境変数を追加する場合は以下の2行の環境変数を.cshrc あるいは.login ファイルにエディターで追加し、FOTRAN の実行ファイルへのパスを追加ください(C シェルの場合)。この操作は envset コマンドを実行したユーザには必要がありません。

```
setenv LD_LIBRARY_PATH /opt/FSUNf90/lib
```

```
setenv MANPATH /opt/FSUNf90/man
```

.cshrc ファイルの path のところに/opt/FSUNf90/bin を追加します。

```
set path=( /opt/FSUNf90/bin $path ..... )
```

環境ファイルを変更したらその変更を反映させるために以下のように source コマンドを実行しましょう。

```
{ws1} 3% source .cshrc
```

```
{ws1} 4% source .login
```

## 2. コンパイルの方法

ソースファイルを使い慣れたエディターで作成します。ここではソースファイルの名前を test.f とします (FORTRAN90 のソースファイルの場合は拡張子.f90 を付けて test.f90 としますが動作はどちらも同じ)。

### 2.1 実行可能ファイルの名前を指定しないコンパイルと実行方法

FOTRAN のコマンドは frt あるいは f90 を使います(動作は同じ)。実行可能ファイルの名前を指定しない場合実行可能ファイル名が暗黙値の a.out となります。

```
[ws1] 5% frt test.f
```

```
[ws1] 6% a.out
```

## 2.2 実行可能ファイルの名前を指定したコンパイルと実行方法

以下の例では実行可能ファイルの名前を test と指定しています。

```
[ws1] 7% frt -o test test.f
```

```
[ws1] 8% test
```

## 2.3 科学技術演算用サブルーチンSSLを使う時

SSLを使う時は以下のようにSSL2のライブラリーをリンクしてお使いください。

```
[ws1] 9% frt test.f -lfssl2
```

## 3. 大きなプログラムの実行方法

大きなプログラムは計算サーバ cs2 でバッチジョブとして実行します。ジョブの依頼(キューイング)はフロントエンドサーバ cs1 から NQS(Network Quing System)を用いて行います。

cs1 と cs2 における NQS の構成は以下の表のように設定されています。

表2 NQS キュー設定

| 設定項目     | フロントエンドサーバ側              | 計算サーバ側                   |
|----------|--------------------------|--------------------------|
| ホスト名     | cs1.ipc.kanazawa-u.ac.jp | cs2.ipc.kanazawa-u.ac.jp |
| マシン id   | 10                       | 11                       |
| g-bach-l | 0                        | 5                        |
| g-pipe-l | 5                        | 0                        |
| g-exte-l | 0                        | 50                       |

ここで

g-bach-l: global-batch-limit 同時走行可能なバッチリクエストの最大数

g-pipe-l: global-pipe-limit 同時に走行が可能なパイプキューサーバ/クライアントの数

g-exte-l: global-extern-limit リモートホストから、自ホストにキューイング可能なバッチリクエストの総数を意味しています。

キューのタイプとしては表 3 に示す4種類が設定されています。フロントエンドサーバ cs1 においては計算サーバ cs2 に送るための pipe キューが作成され、cs2 に送られ batch キューに入ります。

表3 NQS のキューのタイプ

| キューtype         | キューのタイプ: 全て pipe                                  | キューのタイプ: 全て batch  |
|-----------------|---|--|
| キュー<br>(type a) | キュー名 : to_sun_a<br>転送先キュー: sun_a@cs2<br>走行制限数 : 1 | キュー名 : sun_a<br>使用メモリサイズ : 128MB<br>CPU 時間制限 : 30 分<br>経過時間制限 : 制限なし<br>走行制限数 : 2  |
| キュー<br>(type b) | キュー名 : to_sun_b<br>転送先キュー: sun_b@cs2<br>走行制限数 : 1 | キュー名 : sun_b<br>使用メモリサイズ : 256MB<br>CPU 時間制限 : 60 分<br>経過時間制限 : 制限なし<br>走行制限数 : 1  |
| キュー<br>(type c) | キュー名 : to_sun_c<br>転送先キュー: sun_c@cs2<br>走行制限数 : 1 | キュー名 : sun_c<br>使用メモリサイズ : 512MB<br>CPU 時間制限 : 120 分<br>経過時間制限 : 制限なし<br>走行制限数 : 1 |
| キュー<br>(type d) | キュー名 : to_sun_d<br>転送先キュー: sun_d@cs2<br>走行制限数 : 1 | キュー名 : sun_d<br>使用メモリサイズ : 512MB<br>CPU 時間制限 : 240 分<br>経過時間制限 : 制限なし<br>走行制限数 : 1 |

コンパイルからバッチの出力の表示までの一連の操作を紹介致します。

### 3.1 cs1 に telnet コマンドで接続

最初に cs1 に telnet コマンドで接続します。

```
[ws1] 10% telnet cs1
Trying 133.28.250.102...
Connected to cs1.ipc.kanazawa-u.ac.jp.
Escape character is '^]'.
UNIX(r) System V Release 4.0 (cs1)
login: matumoto
Password:
Last login: Tue Jul 21 15:36:47 from icews2
Sun Microsystems Inc. SunOS 5.5.1 Generic May 1996
```

### 3.2 ジョブを実行するためのスクリプトを `run.sh` というファイル名で `vi` や `mule` 等のエディターで作成します。以下の例では `vi` エディターを用いています。

```
cs1[matumoto]1% vi run.sh
```

内容は通常のシェルスクリプトの形式で以下のようにしてください。

```
#!/bin/csh
setenv LD_LIBRARY_PATH /opt/FSUNf90/lib
cd /home/kougaku1/center/matumoto
./a.out
```

ここで1行目はCシェルの指定でそのまま書いて下さい。

2行目は先に延べた環境変数の設定がされていたら不要ですが念のため書いておきましょう。

3行目は各自のホームディレクトリ(実行ファイル `a.out` のある場所)

4行目は実行可能ファイルの名前を書き、実行させます。

ちなみに `vi` を実行した時、以下のように端末タイプがないとしかられた時(`kterm` を使用している例)は一旦 `vi` を終了します。

`kterm`: この端末タイプはありません。

使用中の端末の種類がわかりません。有効な値は 'kterm'。

[オープンモード]

"run.sh" [新規ファイル]

以下のように端末を `vt100` と指定し再度 `vi` を起動してください。

```
cs1[matumoto]2% setenv TERM vt100
```

### 3.3 ソースファイルの作成

ソースファイルを使い慣れたエディタで作成します。本マニュアルにおいては以下のファイルを用いています(警告が出るように作ったソースファイルです)。

```
cs1[matumoto]3% cat test.f
do 10 x=1.0,10.0,0.5
    write(6,*) x
10  continue
    stop
end
```

### 3.4 コンパイル

以下のように実行可能ファイル名を指定しないでコンパイルします。

```
cs1[matumoto]4% frt test.f
cs1[matumoto]5% ls
test.f a.out
```

### 3.5 キューへの投入

キューの投入は qsub コマンドを使用していきます。

```
cs1[matumoto]6% qsub -q to_sun_a run.sh
Request run.sh(20.cs1) submitted to queue: to_sun_a.
```

ここで、-q to\_sun\_a は“-q”オプションで NQS のキュー名(表3 NQS のキュータイプ 参照)を指定しており、これは絶対しないといけません。最後の run.sh は用意したジョブスクリプトのファイル名を指定します。2行目に表されたメッセージに 20.cs1 とあるのは REQUEST ID というもので、投入されたジョブの ID です。

### 3.6 ジョブの実行確認

ジョブの実行の確認は qstat コマンドで行います。

長いプログラムなら以下のように STATE を見ると実行中は RUNNIG と表示され、実行されていることが確認できる(今回のサンプルではこの状態は恐らく見れないでしょう)。

```
cs1[matumoto]7% qstat @cs2
sun_a@cs2; type=BATCH; [ENABLED, INACTIVE]; pri=31
  0 exit;  1 run;  0 queued;  0 wait;  0 hold;  0 arrive;
      REQUEST NAME  REQUEST ID  USER      PRI STATE  PGRP
1:      run.sh      20.cs1     matumoto   31 RUNNIG 482
```

```
sun_b@cs2; type=BATCH; [ENABLED, INACTIVE]; pri=31
  0 exit;  0 run;  0 queued;  0 wait;  0 hold;  0 arrive;
```

```
sun_c@cs2; type=BATCH; [ENABLED, INACTIVE]; pri=31
  0 exit;  0 run;  0 queued;  0 wait;  0 hold;  0 arrive;
```

```
sun_d@cs2; type=BATCH; [ENABLED, INACTIVE]; pri=31
  0 exit;  0 run;  0 queued;  0 wait;  0 hold;  0 arrive;
```

実行が終了した場合は以下のように表示されます。

```

cs1{matumoto}8% qstat @cs2
sun_a@cs2; type=BATCH; [ENABLED, INACTIVE]; pri=31
  0 exit;  0 run;  0 queued;  0 wait;  0 hold;  0 arrive;

sun_b@cs2; type=BATCH; [ENABLED, INACTIVE]; pri=31
  0 exit;  0 run;  0 queued;  0 wait;  0 hold;  0 arrive;

sun_c@cs2; type=BATCH; [ENABLED, INACTIVE]; pri=31
  0 exit;  0 run;  0 queued;  0 wait;  0 hold;  0 arrive;

sun_d@cs2; type=BATCH; [ENABLED, INACTIVE]; pri=31
  0 exit;  0 run;  0 queued;  0 wait;  0 hold;  0 arrive;

```

### 3.7 実行結果の確認

実行結果は標準出力に出され、エラーがあれば標準エラー出力に出され、以下のファイル名の形式になっています。

標準出力:ジョブ実行スクリプト名.o"REQUEST ID の数値"

標準エラー出力:ジョブ実行スクリプト名.e"REQUEST ID の数値"

```

cs1{matumoto}9% ls
a.out                test.f
run.sh.e20           run.sh.o20

```

エラーファイルの内容は(機器番号5が正しくないと警告されている)

```

cs1{matumoto}10% more run.sh.e20
stty.: そのようなデバイスもアドレスもありません。
tty`: 指定が曖昧です。

```

演算結果の出力は

```

cs1{matumoto}11% more run.sh.o20
Warning: no access to tty; thus no job control in this shell...
Sun Microsystems Inc.  SunOS 5.5.1   Generic May 1996
1.00000000
1.50000000
2.00000000
2.50000000
3.00000000
3.50000000

```

4.00000000  
4.50000000  
5.00000000  
5.50000000  
6.00000000  
6.50000000  
7.00000000  
7.50000000  
8.00000000  
8.50000000  
9.00000000  
9.50000000  
10.00000000

ログアウト

### 3.8 ジョブの削除方法

ジョブの削除は `qdel` コマンドで行い、“-k”オプションでジョブの REQUEST ID を指定します。

```
cs1{matumoto}12% qdel -k 20.cs1
```

### 3.9 プリンター出力方法

センターでご使用頂いている場合は最寄りのプリンターへ

```
cs1{matumoto}12% lpr run.sh.o20
```

センター以外の実習室で、その実習室のプリンターへ出力する場合は“-P”オプションでプリンター名を指定して出力します。センター第4実習室のネットワークプリンターNPR215 あるいは NPR216 に出力する場合は

```
cs1{matumoto}12% lpr -PNPR215 run.sh.o20
```

### 3.10 オンラインマニュアルの表示方法

今回紹介した `frt`, `qsub`, `qstat`, `qdel` のコマンドに関するオンラインマニュアルは `man` コマンドで読むことができます(NQS 関連のマニュアルは `cs1`, `cs2` でのみ読むことが可能です)。

`frt` コマンドのオンラインマニュアルを読むには

```
cs1{matumoto}13% man frt
```

`man` コマンドではページャー `less` が使われ、1行スクロールアップするには“改行キー”を押し、1ページスク



ロールアップするには“スペースキー”を押します。途中で表示を止める場合は“q”キーを押してください。

### 3.11 その他の情報の入手方法

新規導入 UNIX システムに関する情報は総合情報処理センターのサーバ及び以下の URL でご覧ください。

<http://wwwedu.ipc.kanazawa-u.ac.jp/>

また、新規導入 UNIX システムに対する質問、要望は以下のメールアドレスまで。松本及びセンターバイト生が可能な限りお答えします。

[request@network.kanazawa-u.ac.jp](mailto:request@network.kanazawa-u.ac.jp)