Mac OS X 10.5 (Leopard) ユーザのための Seismic Un*x (SU) 導入ガイド

八木直史*・筒井智樹*

要 旨

本稿は Seismic Un*x (SU) を Mac OS X 上で動作させる方法について解説する。SU は反射法地震探査データの処理ソフトウェア・パッケージであり、さまざまな UNIX ベースの OS にインストールできる。Mac OS X は UNIX ベースの OS であり、SU を Macintosh としての利用の延長上で使用できる。Mac OS X 上に SU を導入するために必要な UNIX 環境の整備について解説するとともに、SU のソフトウェア・パッケージの入手方法から SU のインストールの手順までを詳しく記述した。

キーワード : Seismic Un*x・Mac OS X・UNIX

1. はじめに

Seismic Un*x (SU) は Colorado School of Mines の Center for Wave Phenomena (CWP) が開発している反射法地震探査データの処理ソフトウェア・パッケージである。SU は、C コンパイラなどのインストールに必要なプログラムをいくつか準備すれば、さまざまな UNIX ベースの OS¹ にインストールできる。フリー・ソフトウェアの SU は、ターミナル² を使った軽快なコマンド操作が可能であり、手軽にデータを解析できるソフトウェア・パッケージとして知られている。解析された地震波形データは PostScript 形式のファイルで保存できる。SU を構成するすべてのプログラムはソースコードの状態で配布されており、自由に書き換えることもできる。SU は地震探査データ処理だけでなく、幅広い波動に関連した処理を行うことが

できるので、“seismic”という語が示すよりずっと一般的なパッケージといえる。SU は世界中で使われており、国内では渡辺 (1999) が概要から導入までを紹介し、佐伯 (1999) がデータ処理の実例を解説するなど、積極的に取り上げられるようになった。また、渡辺 (2003) は Windows 上で Cygwin を導入して SU を動作させる方法を紹介し、SU の利用できる環境を紹介した。

さて、読者は SU を Mac OS X 上でも利用できることをご存知だろうか。Mac OS X は UNIX ベースの OS であり、UNIX 系のプログラムである SU を Macintosh としての利用の延長上で体験できる。2006 年より、アップル社 (Apple Computer Inc.。現在は Apple Inc. に改称) は Mac OS X の対象とする CPU をそれまでの PowerPC の代わりに Intel CPU の採用を発表した。アップル社が Intel CPU を採用するように

¹ OS とはオペレーティングシステムの略称のことである。オペレーティングシステムは、入出力の制御、ハードウェアの管理、プロセスの管理といったコンピュータの基本的な管理・制御を行っている。アプリケーション・ソフトウェアは OS の提供する機能を使って動作するように設計されている。

² ターミナルとは UNIX 系 OS において、CUI (Character User Interface) を用いてファイルの操作などを行うアプリケーション・ソフトウェアのことである。なお、本稿で述べるターミナルとはアップル社の Mac OS X に標準で付属している UNIX 系端末エミュレータのことである。

なった主な理由のひとつは、Macintosh の消費電力を下げることにあったといわれている。しかし、PowerPC で動作するように書かれたすべてのプログラムを Intel CPU で動作するように修正しなくてはならないという弊害も生じた。Intel CPU の採用から 2 年以上が経過した現在においても、Intel CPU 版の Mac OS X を UNIX 系 OS として使う場合は、対応していないコンパイラや UNIX 系プログラムは少なくない。著者は初め Mac OS X への SU のインストールの失敗に悩まされていたが、C や Fortran のコンパイラ群のアップデートにより、SU のインストールに成功した。Mac OS X の UNIX 系プログラムの部分について、Intel CPU での動作がうまくいくように修正が進められている。

Mac OS X は優れた GUI の設定ツールが他の UNIX ベースの OS よりも充実している。また、Mac OS X は表現力の豊かな優れたアプリケーション群を揃えているために、Macintosh 用に開発されたアプリケーション・ソフトウェアを使って処理後の地震波形データの編集や表示を行うことができる。本稿は Mac OS X 上で SU を動作させる方法について解説する。

2. Mac OS X の概要

アップル社はバージョン 9 まで続く Mac OS の後継として、NeXT 社の OPENSTEP という OS を基にした Mac OS X を開発した。Mac OS X は Macintosh 系 OS と UNIX 系 OS の 2 つの側面を持つ (Fig. 1)。Mac OS X のコア OS である Darwin は Mach のマイクロカーネルに FreeBSD のファイルシステムやネットワーク機能を実装したもの³であり、従来の Mac OS と異なる UNIX ベースの OS である。Mac OS X の開発環境は OPENSTEP の API (Application Prog-

³ Darwin とは Mac OS X の基礎となる OS (コア OS) の名称である。NeXT 社が開発した OPENSTEP という OS は BSD と呼ばれる UNIX の系統をもとに開発されており、これに Mach のマイクロカーネルの技術を実装したものが Darwin である。Mach (Multiple Asynchronously Communication Hosts) とは Carnegie Mellon University の Richard Rashid らの手によって開発されたもので、マルチプロセッサ対応やメモリ空間の有効利用などを目的に BSD のカーネルソースコードに修正を加えている。マイクロカーネルとは階層型に設計された OS の中核となる部分のことで、システムのリソースと、ハードウェアとソフトウェアコンポーネントのやりとりを管理する。小規模な機能に限定して設計されているので、マイクロカーネルという意味で呼ばれている。

Fig. 1. Architecture of Mac OS X.

Mac OS X runs on top of Darwin core operating system, and includes Carbon, Cocoa, and Aqua. Darwin provides UNIX environment, with X Window System and Terminal, including UNIX API.

ram Interface) をベースとした Cocoa と、従来の Mac OS の API をベースとした Carbon から構成されている。Macintosh 系のネイティブ・プログラムのインターフェースは従来の Mac OS とも OPENSTEP と異なる Aqua を実装している。これらに加えて、Mac OS X は BSD 互換の API を OPENSTEP から受け継いでいるために、開発環境に UNIX API、インターフェースに Terminal と X Window System も実装している。したがって、Mac OS X は Macintosh 系のネイティブ・プログラムだけでなく、ほとんどの UNIX 系のプログラムもネイティブに利用できる構成になっている。他の UNIX 用のプログラムを Mac OS X 上で開発することも可能である。

UNIX は The Open Group の登録商標であり、その使用には同団体の標準仕様 (Single UNIX Specification) を満たす必要がある。2007 年 11 月 19 日に発売された Mac OS X 10.5 (Leopard) は The Open Group より UNIX 03 標準規格を満たす製品として正式に認定された。ただし、認定によって、Mac OS X と UNIX ベースの OS との互換性が上がったわけではない。先述のように、2006 年より Mac OS X の CPU は PowerPC の代わりに Intel CPU を採用した⁴。Mac OS X 10.5 では従来よりも UNIX 系の環境が大きく変化したために、本稿執筆時 (2008 年 10 月現在)、Mac OS X 10.5 に対応していない UNIX 系のプログラムも残されている。今後はしばらく OS レベルでの大きな変更はないと思われ、Mac OS X の

⁴ アップル社は PowerPC 版の Mac OS X の設計・販売している裏で、別チームを設けて Intel CPU 版の Mac OS X の開発を行っていた。両者は見た目こそ同一であるものの、全く別の OS であり、互換性はない。

UNIX 系の環境に対応したプログラムが徐々に増えるだろう。

3. SU の導入

3.1 SU 導入に必要な環境

3.1.1 Xcode Tools

3.2 で後述するように、SU のコンパイル時の設定ファイルである Makefile.config の内容を確認すれば、SU のメインとなるプログラムのソースコードが C や C++ で書かれており、これらのプログラムをコンパイルするためのツールが必要であることがわかる。本稿では、Mac OS X のインストール DVD に標準で付属している Xcode Tools と呼ばれる開発環境を利用する。Xcode Tools をインストールすると、C や C++ のコンパイラである gcc が `/usr/bin` のディレクトリに作成される。このインストール時に、gcc へのリンクである cc と c++ が自動的に作成されるので、cc と c++ のコマンドを使えば、C と C++ のコンパイラをそれぞれで呼び出すことができる。

なお、Mac OS X 10.5 (Leopard) のインストール DVD に付属している Xcode Tools (バージョン 3.0) の gcc は Intel CPU の環境だと動作しないバグがあり、実際に著者の環境ではこれが原因で、当初、SU を正常にインストールできなかった。アップル社の Apple Developer Connection (ADC) (<http://developer.apple.com/>) の会員に登録すれば、このバグを修正した Xcode Tools の最新バージョン (2008 年 10 月現在で 3.1.1) が無償でダウンロードできる。こちらを利用すると正常に gcc が動作するので、Xcode Tools を利用する際はアップル社の Web ページ (<http://developer.apple.com/technology/xcode.html>) で最新バージョンに更新することをすすめる。

3.1.2 Fortran コンパイラ

SU には C や C++ で書かれたメインとなるプログラムとは別に Fortran 77 で書かれたサブプログラムもある。Fortran 77 で書かれたプログラムは絶対に必要というわけではないが、これらのプログラムには便利な機能も含まれている。そこでこの Fortran 77 で書かれたプログラムを利用するために Fortran コンパイラもインストールしておくことをすすめる。この場合、上記の Xcode Tools には Fortran コンパイラが付属していないので、別途 Fortran コンパイラをインストールする必要がある。著者は f2c と fort77 を組み合わせて利用した。f2c を使って Fortran 77 で書かれたプログラムのソースコードを一旦 C のソースコードに変換した後、その変換した C のソースコードを gcc によりコンパイルする。このコード変換とコンパイルの一連の処理は、fort77 スクリプトを使って実行できる。こ

のように見かけ上、fort77 が Fortran コンパイラのように動作する。どちらもインターネットのいろいろな場所に公開されているので、ダウンロードしてから解凍を行い、適切な場所にコピーする。SU でこれらを使うには、f77 というリンクを作成する必要がある。例えば、`/usr/bin` に fort77 をコピーした場合、ターミナル上で下記のようにして fort77 のシンボリックリンクを作成する。

```
$ ln -s /usr/bin/fort77 /usr/bin/f77
```

なお、本稿ではこの例のように、ユーザがキーボードから入力を行う場合は行頭にコマンドプロンプトを表す「\$」を付けている。上記の場合だと、ユーザが入力する文字列は `ln -s /usr/bin/fort77 /usr/bin/f77` である。

これらのプログラムを組み合わせなくとも、Fortran コンパイラである g77 や gfortran を使ってもよい。ただし、筆者の環境でこれらの Fortran コンパイラを使って SU のインストールを試みると、エラーが返されて作業の途中で終了していた。お手持ちの環境によってはコンパイラのアップデートによりインストールに成功することも考えられるので、先にこちらを試してみてもいいかもしれない。

3.1.3 X Window System

X Window System とは、マサチューセッツ工科大学 (Massachusetts Institute of Technology) を中心に開発された UNIX 向けの開発環境である。SU は基本的に、ターミナルによるコマンド操作だが、X Window System を併用することで地震波形などグラフィックス表示できる。Mac OS X 10.5 では標準で“Applications/Utilities”に X11 としてインストールされている。もし、何らかの理由で X11 がインストールされていないければ、Mac OS X のインストール DVD か X quartz の Web ページ (<http://xquartz.macosforge.org/trac/wiki>) から導入する。

ターミナル上から X11 を立ち上げるには、以下のコマンドを実行する。

```
$ open -a X11
```

X11 を立ち上げると xterm というターミナルエミュレータが自動で開く。Mac OS X 付属のターミナル (Terminal.app) と xterm は好みで使い分けてもらってかまわない。xterm よりもターミナルの方が Mac OS X に最適化されているので使いやすい。もし、xterm を自動で開きたくないなら、ホームディレクトリ以下の“`~/Library/Preferences/org.x.X11.plist`”をテキストエディタで開いて、

```
app_to_run string xterm
```

の xterm の記述されている行を削除すればよい。

3.2 SU のコンパイル、インストール

3.2.1 SU のダウンロード

SU のソースコードは Web ページ(<http://www.cwp.mines.edu/cwpcodes/>) からダウンロードする。本稿執筆時 (2008 年 10 月) の最新バージョンは 41 (2008 年 5 月 16 日公開) である。ソースコードは、アーカイブファイル、`cwp_su_all_41.tgz` あるいは `cwp_su_all_41.tar.Z` として公開されている。これらのアーカイブファイルは、`tar` コマンドによって複数のファイルを 1 つにまとめた後に、`tgz` 形式は `gzip` コマンドによって、`tar.Z` 形式は `compress` コマンドによって圧縮して作成されたものである。どちらのアーカイブファイルも中に含まれているソースコードは同じだが、そのファイルサイズは `cwp_su_all_41.tgz` は 10,858,530 byte、`cwp_su_all_41.tar.Z` は 15,354,451 byte となっている。最近ではブロードバンド回線の普及でファイルのダウンロード速度などを気にしなくてよくなったが、ネットワークのトラフィックを軽減するためにも前者を選択するほうが望ましい。3.2.4 に示すように、Mac OS X ではどちらの形式であっても展開できる。

3.2.2 環境変数の設定

`Makefile.config` を見ると、環境変数 `CWPROOT` に `SU` をインストールするディレクトリ名を設定する必要があることがわかる。まず初めにこの環境変数を設定するために、ターミナルを立ち上げ、UNIX 系の標準テキストエディタ (`vi`, `Emacs` など) を使って、ホームディレクトリ上に `.bash_profile` というログインシェルを作成する。“`/usr/local/SU`”に `SU` をインストールする場合、例えば以下のように `vi` を実行し

```
$ vi .bash_profile
```

`.bash_profile` の中身に

```
export CWPROOT=/usr/local/SU
```

```
export PATH=$PATH:$CWPROOT/bin
```

と記入し、保存する。この `.bash_profile` の内容を反映するには、一度ターミナルをログアウトしてからログインし直す。もしくは `source` コマンドで

```
$ source .bash_profile
```

とすれば、すぐに内容を反映できる。環境設定がきちんと行われているか確認するには

```
$ echo $PATH
```

を実行する。“`/usr/local/SU/bin:`”とターミナルに表示されていれば `CWPROOT` と `PATH` の設定は完了である。なお、ターミナルでなく、`xterm` を利用しているのであれば、`.bash_profile` の代わりに `.bashrc` というログインシェルを同じように作成する。

3.2.3 ファイルの移動

上述の環境変数が設定できた後、以下の操作を行う。

先ほど指定した `SU` インストール用ディレクトリが存在しない場合には、以下のコマンドを用いて作成する。

```
$ mkdir $CWPROOT
```

ダウンロードしたアーカイブを `$CWPROOT` に移動させる。

```
$ mv cwp_su_all_41.tgz $CWPROOT
```

3.2.4 アーカイブファイルの展開

カレントディレクトリを `SU` インストール用ディレクトリに移動する。

```
$ cd $CWPROOT
```

圧縮されたアーカイブファイルを展開する。

```
$ tar zxvf cwp_su_all_41.tgz (もしくは cwp_su_all_41.tar.Z)
```

ソースコードは `$CWPROOT/src` 以下に展開され、バージョン 41 の場合、その全ファイル容量は 42,393,114 byte である。正常に展開できたらアーカイブファイルは必要ないので以下のコマンドで削除する。

```
$ rm cwp_su_all_41.tgz (もしくは cwp_su_all_41.tar.Z)
```

3.2.5 Makefile.config の設定

ソースコードのあるディレクトリ `src` に以下のようにして移動する。

```
$ cd src
```

このディレクトリ `src` の中には、`SU` のプログラムファイルのコンパイル時 (後述の `make` コマンド実行時) に参照される設定ファイル `Makefile.config` がある。この `Makefile.config` には、複数のプログラムファイルの依存関係などが記述されている。この設定ファイルにより、プログラムのコンパイル作業を効率的に実行することができる。ただし、通常各自の環境に合わせて `Makefile.config` を書き直しておく必要がある。Mac OS X 10.5 に `SU` をインストールするために、`Makefile.config` の中で変更すべき項目を Fig.2 に示した。なお、ここまでの作業はターミナル上での作業を前提に解説してきた。しかし、`Makefile.config` ファイルは 500 行以上にもおよぶ分量があるので、Macintosh 上で使い慣れたテキストエディタを使って設定の修正作業を行ってもいいだろう。

3.2.6 実行ファイルの作成

ディレクトリ `src` 上で `make` コマンドの後ろに以下のようなオプションをつけて、`SU` のインストールを行う。

```
$ make install
```

```
$ make xtinstall
```

```
$ make fininstall
```

```
$ make mglininstall
```

```
$ make xmininstall
```

途中でエラーメッセージが返されなかったらインストールは完了である。コンパイルされた `SU` のコマンド


```

100 # XDR is the external data representation. This flag, required for
101 # installations on all 64 bit machines, is to provide data portability
102 # among all platforms. (We recommend that everybody use XDR.)
103 #
104 # Default is to not set the SUXDR option
105 #XDRFLAG =                                コメントアウト
    中略
110 # for XDR under Linux and Mac OS X
111 XDRFLAG = -DSUXDR -D_BSD_SOURCE           こちらを選択
    中略
114 #####
115 # Large file support, see Portability/README_LARGE_FILES for more info
116 # Default, do not assume LARGE_FILES
117 #LARGE_FILE_FLAG =                        コメントアウト
    中略
123 # uncomment if you want to install with large file support
124 # Mac OS X
125 LARGE_FILE_FLAG = -D_FILE_OFFSET_BITS=64 -D_LARGEFILE_SOURCE -DGNU_SOURCE
    . C                                    こちらを選択
    中略
335 #####
336 # Default C preprocessor (cpp) definition
337 # this is the one you will use on most systems
338 # AIX, IRIX if using the native cc compiler, Linux
339 #CPP = /lib/cpp                            コメントアウト
    中略
359 # use this on Mac OS X and Free BSD
360 CPP = /usr/bin/cpp                        こちらを選択
    中略
363 #####
364 # Fortran materials.
365 # The majority of the SU package does not depend on fortran, so you
366 # are free to ignore this if you are not planning on doing the 'finsinta
    . ll'
367 # portion of the installation.
368 #
369 # Please note, there may be additional compiler flags necessary to get
370 # the Fortran codes to work properly. You may need to experiment.
371 #
372 # Default for IBM RS6000, SGI, also for the Intel compiler ifort
373 OPTF = -O                                こちらを選択
374
375 # Default for Linux and other GCC installs (most likely FreeBSD and Mac
    . OS X)
376 # with g77
377 #OPTF = -O -static -mcpu=i686 -fno-globals -fno-automatic -fno-second-un
    . derscore                            コメントアウト
    中略
388 #####
389 # Fortran compiler
390 # default (use g77 on Linux systems, f77 there is really f2c)
391 FC = f77                                  こちらを選択
392
393 # GNU fortran compiler (Linux systems)
394 #FC = g77                                コメントアウト
    中略

```

Fig. 2. The setting of makefile.config of SU Release 41.

Only the changed item is excerpted. A left number shows line number, and · shows turn of line.

Fig. 3. Demo for traditional finite difference datuming.

は\$CWPROOT/bin 以下に収められている。バージョン 41 の場合、上記 5 つの make コマンドの実行により作成される SU コマンドの合計は 399 で、その全容量は 23.6MB となっている。

なお、アップデータの反映など、再インストールを行う場合は以下のコマンドを実行する。

```
$ make remake
$ make xtremake
$ make fremake
$ make mglremake
$ make xmremake
```

著者の環境では Xcode Tools バージョン 3.0 に含まれていた gcc のバグにより、最初 SU のインストール作業に失敗した。しかし、Xcode Tools バージョン 3.1.1 にバージョンアップしたところ SU のインストールに成功した。このように、Makefile.config の設定ファイルの問題でなく、コンパイラやスクリプトなどのプログラムに問題がある可能性もある。どうしても SU インストール作業に失敗する場合には、コンパイラの再インストールやバージョンアップを行うか、あるいは同様の機能を持つ別プログラムを用意した後、SU のインストールを再度試みることをすすめる。

4. 動作の確認

SU のインストールが完了したら、標準で用意され

ているデモ用のスクリプトを使って SU 環境の動作を確認できる。これらのデモスクリプトは \$CWPROOT/src/demos のディレクトリ内にある。参考として音響波動方程式の (2 次) 有限差分法モデリングのデモスクリプト (XDemo3 と XDemo4) の実行例を以下に示す。

4.1 ディレクトリの移動

カレントディレクトリを移動する。

```
$ cd $CWPROOT/src/demos/Synthetic/Finite_Difference/Sufdmod2
```

4.2 デモスクリプトの実行

X11 が起動していることを確認し、ターミナル上で、

```
$ ./XDemo3
$ ./XDemo4
```

と入力すると、モニター画面上に Fig. 3 のような波形が表示される。

波形表示の画像ファイルを作成するデモスクリプトとして PSDemo が用意されている。PSDemo は PostScript 形式のファイルを作成するスクリプトの例である。

```
$ ./PSDemo
```

4.3 ファイルの削除

上記のデモスクリプトの実行により作成されたファ

イルは、Clean.sh というスクリプトで一度に削除できる。

```
$ ./Clean.sh
```

SU にはアイコンやウィンドウのような、直感的で視覚的な操作感はない。しかし、デモを動作させれば、素早くファイルを操作でき、複雑な操作を簡単に行うことが可能だと気付く。観測記録を用いた詳細な処理については渡辺 (1999) や佐伯 (1999) を参照していただきたい。

5. おわりに

本稿は Mac OS X ユーザを対象に SU をインストールする方法を解説した。SU は反射法地震探査に興味がある人にとって非常に魅力的なツールである。商用のソフトウェアにあるすべてのデータ処理をまかなうことはできないが、基本的な解析なら十分にこなすことができる。また、いままで Mac OS X を UNIX として利用したことのないユーザは本稿に書かれているインストール作業を一通り理解できれば、Mac OS X の世界がさらに広がることだろう。また、表現力豊かな Mac OS X のアプリケーション群と SU の処理がスムーズにつながることは、生産性の向上に大きく寄与することであろう。本稿が各位の SU の利用の参考になれば幸いである。

謝 辞

最後に、本稿の執筆をすすめていただいた名古屋大学渡辺俊樹先生、さまざまな形でサポートしてくれた秋田大学の学生の皆様に感謝いたします。

参 考 文 献

- Apple, “Apple Developer Connection”, <http://developer.apple.com/>
- アスキー書籍編集部 (2008): Mac OS X 10.5 Leopard UNIX 的システム構築, アスキー, 東京.
- Center for Wave Phenomena, “CWP/SU: Seismic Un*x”, <http://www.cwp.mines.edu/cwpcodes/index.html>
- Cohe, J. K. and Stockwell, Jr. J. W. (2008): CWP/SU: Seismic Unix Release 41: a free package for seismic research and processing, Center for Wave Phenomena, Colorado School of Mines.
- 大津真 (2008): Mac OS X 10.5 Leopard as UNIX, ローカス, 東京.
- 佐伯龍男 (1999): Seismic Un*x (SU) 入門(2)—データ処理の実例(1), データ入力とトレースヘッダの設定—, 物理探査, **52**, 465–476.
- 渡辺俊樹 (1999): Seismic Un*x (SU) 入門(1)—概要・入手から導入まで—, 物理探査, **52**, 274–286.
- 渡辺俊樹 (2003): Windows ユーザのための CWP/SU 導入ガイド, 物理探査, **56**, 91–96.

An introduction guide to Seismic Un*x (SU) for Mac OS X 10. 5 (Leopard) users

Naofumi Yagi* and Tomoki Tsutsui*

ABSTRACT

This tutorial shows an installation guide of Seismic Un*x (SU) on Mac OS X. SU is one of popular processing software packages of seismic reflection data, and runs on UNIX system. Mac OS X is an operation system (OS) developed based on UNIX system, and therefore you can use SU on Mac OS X. Practical procedures for installing SU on Mac OS X system are described in this article.

Keywords : Seismic Un*x, Mac OS X, UNIX