Short Note

Help in the SEP computer environment

Matthias Schwab [1]

/usr/local/src/our/sephelp/sephelp

INTRODUCTION

When working in the SEP computer environment you encounter certain computer tools. This document lists documentation and resources for these different tools.

UNIX

All machines at SEP are based on the UNIX operating system. It is extremely beneficial to know the basics of UNIX. All UNIX commands are documented on-line: (type `man commandname`). If you do not know the command name try `man -k keyword`. For a more overall description, I suggest Kernighan's UNIX book (1984). If you have never used a UNIX computer you probably should take Phil Farrell's fall computer class (Farrell, 1994).

EDITOR

At SEP two different editors are widely used: `vi` and `emacs`. The traditional UNIX editor `vi` exists on any UNIX machine and is a well trusted editing work horse. Any UNIX book documents `vi`. The `emacs` editor is more sophisticated: It allows you to use the mouse, comes with scroll and menu bars etc. To learn how to use `emacs` try the emacs built-in tutorial: start up `emacs` by typing `emacs` then press `Control-h t` in the `emacs` window. The on-line info file (type `xinfo`) and the emacs author Stallman (1993) give a more detailed description.

RATFOR

The AT&T ratfor preprocessor allows the author of an FORTRAN source code to use concise, C-like flow formulations. The preprocessor translates a ratfor program into a standard fortran version (`ratfor foo.r > foo.f`). A summary of the ratfor dialect is included in Claerbout's book (1994). Kernighan's book (1976) includes a in depth discussion of ratfor.

[1] **email:** matt@sep.Stanford.EDU

SAW, SAT

The `saw` and `sat` preprocessors were developed at SEP. They allow besides other things the dynamical allocation of arrays within FORTRAN. For documentation see the man pages: `man saw, man sat`.

CAKEFILE

The `Cakefile` utility is a derivative of the `Makefile`. It maintenance a set of files according to defined dependency rules.

The rule syntax and the cake capacities are best learnt by studying well written Cakefiles (such as in Claerbout's book (1994)). The on-line man pages (`man cake`) are another resource. Be aware that Cakefiles have include statements which at SEP hide commands which are common to many Cakefiles. To study what the entire Cakefile comprises, type `cake -Z`: it prints the output of the C preprocessor. The command `cake -n` prints out the actions the cake file would take if the `n` would be omitted. When cake encounters an error, typing `cake -v` results in a more verbose error message.

To learn how SEP uses the cake utility I suggest you type `CAKE` on your keyboard to read a short article.

SEPlib

SEPlib is a processing environment for seismic data processing. It comprises two basic concepts: Its concept of datafiles and its concept of processing routines.

A dataset comprises a header file which specifies the data values in terms of type, dimensions, and axes. The data values are stored in a separate file as a single array.

The processing routines are designed according the UNIX philosophy as a set of small program utility which can be combined by piping.

All SEPlib programs document themselves when you type their name without any argument (try for example: `Window`). If you do not know the name of a SEPlib program, but you know what you want to accomplish try the keyword utility: for example `keyword subsample`. By typing `SEPTOUR` you will find an on-line description on SEPlib in form of a short paper. Note, the reference list for further references. Additionally, Claerbout's book (1992) has a short summary of SEPlib as an appendix.

TEX, LATEX

TEX is a typesetting language for scientific writing. LATEX is a package of TEX macros which ease the usage of TEX. SEP uses LATEX for almost all its documents. Especially, SEP takes advantage of the `syscall` utility which allows authors to include interactive buttons in their document.

For general documentation on TEX I suggest Knuth's original TEX book (1984). The traditional documentation for LATEX is Lamport's book (1986). To learn more about SEP's adoption of the TEX typesetting language type `TEXTOUR`.

XTPANEL

xtpanel lets you build panels containing interactive objects such as buttons, sliders, and text fields. You find an introduction to xtpanel by typing `XTPANEL` on any SEP computer. More interesting is the interactive xtpanel help and tutor facility: just type `xtpanel -help` or `xtpanel-tutor`.

IDOC

Typing `IDOCORG` on your keyboard on any SEP computer, will bring up a paper that describes how to combine the different components above to an interactive document.

WHO TO ASK

If you run into problems with anything you do here at SEP, I suggest you follow a certain line of consultation. First, try find the information at your fingertip: man pages (`man commandname`), a SEPlib program's selfdoc, somebody else's example (look into Claerbout's on-line book, e.g. `~prof/tdf`), on-line documents (such as CAKE, SEPTOUR, TEXTOUR, XTPANEL, IDOCORG), or try `xwais`, if you know it. If you cannot find anything, then do not waste too much time, but ask somebody. Best you start with a colleague of yours; he probably struggled with the same problem just a few minutes ago. If he does not have a sufficient answer see (or email) a more senior student (or your teaching assistant if you are taking a class). Finally, you may want to see Jon Claerbout ...

REFERENCES

Claerbout, J. F., 1992, Earth Soundings Analysis: Processing Versus Inversion: Blackwell Scientific Publications.

Claerbout, J. F., 1994, Applications of three-dimensional filtering: Stanford Exploration Project.

Farrell, P. ., 1994, Introduction to computing in earth science: , Stanford University, Time schedule, 24.

Kernighan, B. W., 1976, Software tools: Addison-Wesley.

Kernighan, B. W., 1984, The unix programming environment: Prentice Hall.

Knuth, D. E., 1984, The TEXbook: Addison-Wesley.

Lamport, L. ., 1986, The LATEXbook: Addison-Wesley.

Stallman, R. M., 1993, Gnu emacs manual: Free Software Foundation.