Other useful tools

Eugeniy E. Mikhailov

The College of William & Mary

Lecture 29

Specialization is . . .

A human being should be able to change a diaper, plan an invasion, butcher a hog, conn a ship, design a building, write a sonnet, balance accounts, build a wall, set a bone, comfort the dying, take orders, give orders, cooperate, act alone, solve equations, analyze a new problem, pitch manure, program a computer, cook a tasty meal, fight efficiently, die gallantly.

Lazarus Long, "Time Enough For Love"
by Robert A. Heinlein

Specialization is . . .

A human being should be able to change a diaper, plan an invasion, butcher a hog, conn a ship, design a building, write a sonnet, balance accounts, build a wall, set a bone, comfort the dying, take orders, give orders, cooperate, act alone, solve equations, analyze a new problem, pitch manure, program a computer, cook a tasty meal, fight efficiently, die gallantly. Specialization is for insects.

Lazarus Long, "Time Enough For Love"
by Robert A. Heinlein

Scientist computer related toolbox

- Programming languages
  - Matlab/Octave - computational prototyping and post processing
  - C/C++/Fortran - fast number crunching
  - Other languages of choice: Python, Java, . . .
  - sed, grep, perl - data parsing
  - bash, zsh, tclsh - shells for program start up and interfacing
  - tcl/TK - gui, program interfacing, bindings to compiled functions
- remote computers access
  - ssh, putty - secure terminals, scp (and other analogs) - secure copy
  - screen - ability to deattach and reattach programs output
- Operational Systems
  - Unix, Linux - stable and easy to use
  - Macs - flashy but has Unix inside (if you know how to get there)
  - Windows - main stream, usually has drivers for scientific hardware
- data acquisition
  - LabWindows - C/C++ like language
  - LabView - easy things are easy, hard are almost impossible
  - home made interfaces - what ever you do make your data human readable (ideally text)

Notes

Notes

Notes

Notes

Scientist computer related toolbox (continued)

- make - automatic dependencies resolver
- Data synchronization between computers
  - rsync - only modifications are copied
  - unison - easy to use wrapper for above
- Data backups and archiving
  - ideal backup/archive should put important data to at least 3 location separated by more than 50 km
  - rdiff-backup - archiving wrapper for rsync
- Report preparation
  - vim (Vi iMproved), emacs - editors
  - latex and friends - publication quality output
  - txt2tags - convert simple text format to tex, html, txt, pdf and so on
  - powerpoint, beamer (latex) - presentations
- version control software
  - cvs, svn, darcs, git, bazaar, subversion

General guidelines

- what ever you use read the manual!
- if you cannot access computer remotely do not use it
  - do you really want to run home if you forget a file?
- be a human and thus lazy
  - if you did something more than twice write a script for it
- what ever you do stay away from point and click interfaces
  - they are easy to start using but hard to expand and automate
- if you can type commands so can the other program
  - here come requirements for human readable data streams and command interfaces
- do not rerun computations if you can save the intermediate results
  - sometimes there are power failures, you do not want to restart a month long computation
- split your work
  - gather data
  - process/analyze data
- analyze your data with scripts
  - if you find error in the method, you just fix the script and let computer to reanalyze the data

Free software

When ever you can use free (see The Free Software Definition) software

- it is usually free of charge as well
- developers usually more responsive to your bug reports
- worse case scenario you can fix it yourself and nobody will put you in jail for this

I personally use

- Linux - as a free operational system which has more than 29,000 software packages available (at least if you use Debian)
- Octave as an open software substitution for Matlab
  - they are not 100% interchangeable but close enough
- gnuplot for publication quality plots preparation

Version control software - git

git - Designed by Linus Torvalds and community to

- have back up of your job
- easily recover mistakes
- annotate changes
- synchronize with remote repositories
- share your work and patches with others (web, emails, etc)
- see who did what
- keep your own forks of projects
- keep decision about including other people contribution to yourself

While it was designed mainly for programmers it can be used for many other things (especially if your files are mainly in the text format)

Notes

Notes

Notes

Notes

Final remarks

- Science is not about computers, it is about their use
- Computer is just a tool, it useless without infrastructure
- Computers do not do what you want but what you asked for
- Programming is easy, debugging is hard
  - if you do not know what to do, you want be able to ask computer to do it
  - ALWAYS think about test cases
- Things to avoid
  - never call the result of computer simulation/modeling as result of experiment
  - never blindly trust the result of modeling or program output
  - don't you ever base your decision only on numerical simulation i.e. do not let computer make decisions instead of you
    - Always have an override switch
- Be lazy but in a good way

Notes

Notes

Notes

Notes