Other useful tools

Eugeniy E. Mikhailov

The College of William & Mary

Lecture 11

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

Scientists' computer related toolbox

- Programming languages
  - Matlab/Octave - computational prototyping and post processing
  - C/C++/Fortran - fast number crunching
  - sed, grep, perl, awk - data parsing
  - bash, zsh, tclsh - shells for program start up and interfacing
  - tcl/TK - gui, program interfacing, bindings to compiled functions
  - Other languages of choice: Python, Java, ...
- Remote computers access
  - ssh, putty - secure terminals, scp (and other analogs) - secure copy
  - screen - ability to deattach and reattach programs output

Scientist computer related toolbox (continued)

- Operational Systems
  - Unix, Linux - stable and easy to use
  - Macs - flashy, but has Unix inside (if you know how to get there)
  - Windows - usually has drivers for the scientific hardware
- Data acquisition
  - Matlab
  - LabWindows - C/C++ like language
  - LabView - easy things are easy, hard are almost impossible
    - I personally avoid it at all cost
  - home made interfaces - whatever you do, make your data human readable (ideally, it should be in plain text)

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. Yes, they are not the same.
  - ideal backup/archive should put important data to at least 3 locations 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
  - pandoc, txt2tags - convert simple text format to tex, html, txt, pdf and so on
  - beamer (latex) - presentations
  - some people use PowerPoint, but only because it did not yet fail them during an important presentation.
- Version control software
  - cvs, svn, git, darcs, bazaar, subversion

General guidelines

- Whatever 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.
- Whatever 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 comes the requirement for the human readable data streams and command interfaces.
- Do not rerun all computations, save the intermediate results.
  - You do not want to restart a month long computation because of a power glitch.
- Split your work:
  - gather data,
  - process/analyze it.
- Analyze your data with scripts.
  - If you find an error in the method, you just fix the script and let a computer reanalyze the data.

Free software

Avoid use of software which is not used by its developers.
Whenever you can, use free (see The Free Software Definition) software

- It is usually free of charge as well.
- Developers are usually more responsive to your bug reports.
- In the worst case scenario, you can fix it yourself and nobody will put you in jail for doing it.

I personally use

- Linux as a free operational system which has more than 43,000 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 an archive of your work
- 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 by programmers for programmers, it can be used for many other things (especially, if your files are in the simple text format).

Final remarks

- The most important part of the computer is the one placed between a screen and a chair.
- Computers and programs are just tools. They are useless by themselves.
- Computers do not do what you want but what you asked for.
- Programming is easy, but debugging is hard.
  - If you do not know what to do, you won't be able to ask computer to do it.
  - ALWAYS think about test cases.
- Things to avoid
  - Never call the result of computer simulation/modeling a result of an experiment. Experimental physicists will be very upset.
  - Never blindly trust the result of modeling or a program output.
  - Don't ever base your decision solely on a numerical simulation, i.e., do not let the computer make your decisions for you.
    - Always have an override switch.
- Be lazy, but in a good way.