

Applied Statistics

Quick Reference (Cheat Sheet) to Linux, Emacs, C++, and ROOT

Linux:

The advantage of the operating system **Linux** is (apart from being free and frequently updated!), that it allows fast computations, the use of scripts (command macros) and quick access to a large number of files spanning many computers on a network. For this and countless other reasons, it has become the choice of scientific computing. The directories "home", current, and one level below are shortened "~", ".", and "..", and help on commands is obtained by "man [command]".

Command's Purpose	Command	Linux Example
Go to directory	cd	cd thisdirectory
Copies files	cp	cp thisfile.txt ~/thisdirectory/.
Moves files	mv	mv thisfile.txt ~/thisdirectory/.
Lists files	ls	ls -l thisdirectory
Deletes files	rm	rm thisfile.txt, rm -r thisdirectory
Creates a directory	mkdir	mkdir thisdirectory
Finds a string in a file	grep	grep thisword thisfile.txt
Screens through a file	more/less	more thisfile.txt, less thisfile.txt
Access other computer	ssh	ssh top.nbi.dk, ssh -X thisuser@top.nbi.dk
Copy from other computer	scp	scp petersen@top.nbi.dk:thisfile.txt .

Emacs:

The strength of the editor **Emacs** is its great speed, many possibilities, good color coding of recognized file types, and use of short cuts. The latter is also the difficulty of Emacs, but once it is learned, it makes it a *very* fast way of editing text. Generally, **Emacs** short cuts works by the use of the [Ctrl] key followed by something (with [Ctrl] or not).

Command's Purpose	Command	Example/Comment
Starting emacs	emacs [file]	emacs thisfile.txt
Changing buffer	[Ctrl] x + b	Toggle between files
Quitting emacs	[Ctrl] x + [Ctrl] c	
Aborting command in progress	[Ctrl] g	Stop e.g. [Ctrl] x
Undo last command	[Ctrl] x + u	Also works with [Ctrl] x + -
To the beginning of the line	[Ctrl] a	
To the end of the line	[Ctrl] e	
Search	[Ctrl] s	
Search and Replace	[Esc] %	
Spell check	[Esc] \$	Checks word at cursor
Set mark at cursor	[Ctrl] SPC	For marking regions
Kill (and copy) region	[Ctrl] w	
Kill (and copy) line	[Ctrl] k	
Insert last thing killed	[Ctrl] y	Paste
Open (new) file	[Ctrl] x + [Ctrl] f	Either existing or new file
Save file under current name	[Ctrl] x + [Ctrl] s	Frequently used
Save file under new name	[Ctrl] x + [Ctrl] w	
Start keyboard macro	[Ctrl] x + (Records keys pushed
End keyboard macro	[Ctrl] x +)	Registers recording
Execute keyboard macro	[Ctrl] x + e	Executes recording
Multiple executions	[Ctrl] [number]	Extremely useful with macro

C++:

The programming language **C++** has become a standard in many fields, due to its programming liberty, performance, object orientation, and many other things. **C++** is case sensitive, and commands/lines are separated by a ";". Regions (called scopes) are defined between { and }, and variables only exist within the scope of declaration. Lines are commented out by "//" and sections by "/*" [section] "*/". An additional advantage is, that **ROOT** is written in **C++** and can be easily included in **C++** programs.

Command's Purpose	Command	C++ Example
Loop over commands	<code>for</code>	<code>for (int i=0; i<N; i++) { ...; }</code>
Ask questions	<code>if/else</code>	<code>if (a < b) { ...; } else { ...; }</code>
Do while	<code>do/while</code>	<code>do { ...; } while (a < b)</code>
Arithmetic operators	Add +, Subtract -, Multiply *, Divide /, Modulus %	
Relational operators	Less than (or equal) < (=<), Greater than (or equal) > (>=), Equal ==, Not equal !=	
Logical operators	And &&, Or , Not !	<code>if (a < b) && (c == 1) { ...; }</code>
Data types	bool (0-1), int (± 32767), long int (± 2147483647) float (6-7 digits), double (14-15 digits), char (letter), string, FILE	
Arrays of types	<code>int a[2] = {0,1}; float b[3]; char[] = "statistics";</code>	
Structures of types	<code>struct</code>	<code>struct box{int l; int w; int h;}; int a = box.h;</code>
Objects (types + oper)	<code>class</code>	<code>class plane{int l; int w; GetArea();};</code>
Print to screen	<code>printf</code>	<code>printf("int a and float b: %4d %5.2f \n", a, b);</code>
Print to file	<code>fprintf</code>	<code>fprintf("int a and float b: %4d %5.2f \n", a, b);</code>
Open and close file	<code>open/close</code>	<code>FILE *f = fopen("test.txt","w/r"); fclose(f);</code>
Read from file	<code>fscanf</code>	<code>fscanf(f,"%d %s %lf \n", &int1, &char1, &float1);</code>

ROOT:

For analysing, fitting, and displaying data **ROOT** is CERN's C++ based software. It can read both text files and Ntuples (.root) efficiently, and has a very extensive library of functions/procedures. Also, more advanced tools such as TMVA (neural network) and RooFit (multi dimensional fitting) are based on **ROOT**. All **ROOT** classes start with a "T", and generally have a long list of associated member functions (i.e. operations), which controls the class.

Object	Class	ROOT Example and most used classes
Canvas	TCanvas	<code>TCanvas* c1 = new TCanvas("c1", "Title", X, Y, sizeX, sizeY);</code>
1D histogram	TH1F	<code>TH1F* h1 = new TH1F("h1", "Title", nBins, Min, Max);</code> <code>Draw(); Sumw2(); Fit(); Add(); Integral(); KolmogorovTest();</code>
2D histogram	TH2F	<code>TH2F* h2 = new TH1F("h2", "h2", nX, lX, hX, nY, lY, hY);</code>
1D function	TF1	<code>TF1* g1 = new TF1("g1", "gaus", Min, Max);</code> <code>TF1* p1 = new TF1("p1", "[0]*x + [1]", Min, Max);</code> <code>Draw(); Eval(); FixParameter(); GetProb();</code>
1D Graph	TGraph	<code>TGraph* g1 = new TGraph(N, x, y); float x[N]; float y[N];</code>
with errors	TGraphErrors	<code>TGraphErrors* ge1 = new TGraphErrors(N, x, ex, y, ey);</code>
Profile plot	TProfile	<code>TProfile* prof1 = h2->ProfileX("prof1", Min, Max);</code>

The best introductions are found on the web, and generally searching Google gives many results. For **C++** and **ROOT** the pages www.cppreference.com and root.cern.ch are very good.

Good Luck entering the real om scientific computing,
Troels C. Petersen (NBI, September 2009)