

TNM093 — Practical Data Visualization and Virtual Reality Laboratory Platform

August 25, 2017

1 Introduction

The laboratory exercises in this course are to be conducted in an environment that might not be familiar to many of you. It is based on open source software.

We use an open source development environment of many reasons. For example, the tools are free for the students that which to use them from home, as well as for the university. Also, these tools are used by advanced programmers and developing teams all over the world, making them typically well documented and filled with advanced features, and there are blogs and tutorials freely available on the web.

1.1 Purpose

The purpose of this exercise is to allow you to familiarize with the environment in which the other exercises are conducted. You will test some features that are useful for the exercises and other that may speed up the development work in the project.

[Note: There is no examination of this exercise.]

2 Debian/Ubuntu

The computers in the linux laboratory run a version of Ubuntu Linux called Kubuntu which belongs to the family of Debian distributions.

Software on Debian-based systems are installed through *deb* packages. These packages contain binary executables, libraries and resources for the software, and a control file specifying dependencies. The packages typically reside on a repository so that the software installation system, called *Advanced Packaging Tools* or *APT*, can automatically install dependencies when necessary, as well as update software when a new version is available.

3 Bash

Bash is the most common command-line environment for Linux systems, and a powerful environment for file browsing, searching, conversion and filtering, compilation, etc. It is not by itself a program with a user interface, but is the shell used in a terminal program, such as Konsole in KDE. To begin using Bash with Konsole, start Konsole in K / Programs / System / Konsole.

Bash provides you with a prompt where you can execute commands. A list of common commands is provided below. Try out some of the commands, for example `ls` and `pwd`. Most commands take flags that allow you to control or modify the executed behaviour. Try, for example, `ls -l` and `ls -F --color`.

3.1 Manuals

To find out what a program or command does or what flags you can use, you can either try it and try to decipher the result, or you can use the manual. Most command-line programs and commands come with a manual, called a man page, which is opened by using the command `man` with a program name or command as argument. Try for example `man man` or `man ls`.

When reading a manual, your arrow keys navigate up and down. Use slash (/) so initialize a search, 'n' to search for the next occurrence and use 'q' to quit.

Since Bash is an advanced environment, reading its manual can be daunting. Take a look at `man bash` anyway, just to get a glimpse of what you can do in the environment.

3.2 Paths and Files

Changing directory is done using the `cd` command. If you want to change directory to a subfolder, do `cd <subfolder>`. Try it and do `ls` to see what contents that folder has. The current directory has the dot as shortcut, that is `cd .` will go to the current directory. This is not as useless as it may seem. The parent folder has shortcut dot-dot, that is `cd ..` will change directory to the parent. Of course, you do not have to go to a directory to list its contents, simply try `ls <subfolder>` instead.

The Linux and UNIX operating systems have one directory tree, as opposed to some systems that expose one tree for each file system or drive. The root begins at "/" so if you want to change directory to the root of the directory tree you will write `cd /`. Other file systems or drives will be *mounted* into directories on the main directory tree, thereby creating subtrees. These subtrees will not look or behave any different from the main tree. For example, your home folder, which is on a network file system, will reside at the path `/home/<liuid>`, so to go there you may write `cd /home/<liuid>`. The home directory is not always located under `/home/` so for convenience all user's home directory has a shortcut "`~<liuid>/`". When the liuid is omitted, your own is implicit, so try for example `cd ~/`.

Task 1 — Decipher long paths: (optional)

Try to decipher the path `~/../../usr/local`. What would it mean if your current working directory is your home? What if your current working directory is the root?

Command	Description
<code>ls</code>	List all files in a directory
<code>mv</code>	Rename and/or move files to a target directory
<code>cp</code>	Copy files to a target directory
<code>rm</code>	Remove files or directories
<code>man</code>	Displays the manual page for a command
<code>pwd</code>	Prints the path of the working directory
<code>du</code>	Disk usage information
<code>grep</code>	Search for a string in files
<code>find</code>	Search for files
<code>sort</code>	Sort input data or data in a file
<code>cut</code>	Pick out parts of input data
<code>ssh</code>	Connect to another computer through a secure shell
<code>scp</code>	Copy file(s) from or to another computer through an encrypted connection

For security reasons, not all files can be executed, only files that have the correct *flags*. If you write `ls -l` you will see the flags to the left in the list. These are written as `rw-rw-rw-` meaning readable/writable/executable for the user, for the user's group and for others, respectively.

4 The Build Environment

The UNIX world, including Linux, has one of the most advanced, complete and free build environment for C and C++. It is mostly developed within the GNU project and consists of GNU Make, GNU C Compiler, GNU C++ Compiler, GNU Debugger, etc.

4.1 Compiling Code

The GNU C/C++ compiler command is `gcc`, or `g++` if all input files are known to be C++. For very simple examples, the compiler can be executed manually, for example using the following command-line, however it is more common to use a build system to handle the call to the compiler.

```
g++ file.c -o file
```

The compiler can be controlled through a long list of flags to provide different levels of optimization, warnings and debug options, among other things. You will experiment with some of these in this course.

4.2 Build Systems

Make is a build system that automatically calls the compiler and linker when necessary and with the correct flags based on the current settings. These settings are read from a text file which per default is named “Makefile”. For example, to compile a simple program called “test” write `make test`. Make will then search for a source file which starts with “test” and compile it with the appropriate compiler. Make will use dependency rules to find out which files to compile, in which order or concurrently, and will use old object files if they are newer than the source from which they came.

For smaller projects the Makefile is typically written by hand. For more advanced projects some automatic tool is typically used for configuration, especially if the software is expected to be compiled on different platforms with different hardware and third-party libraries. There are several such automatic tools, some of which are:

autoconf creates a complicated Makefile for a project. Very advanced and very complicated to set up as a developer,

SCons is a build system like Make, but Python-based and extendible,

CMake is an increasingly popular tool to create build settings for different platforms and compilers.

5 Editors

There are many editors available and we have installed a small set of the most popular ones for development: Kate, Emacs and Eclipse. Observe that all professional grade editors for source code can be configured to follow code formatting standards, which is necessary for programmers on different platforms and systems to be able to collaborate.

5.1 Kate

Kate, *K Advanced Text Editor*, is an advanced text editor with syntax highlighting for many programming languages and text formats. It is straightforward to use and supports the standard shortcuts.

5.2 Emacs

Emacs is one of the most powerful editors in existence, with syntax highlighting and automatic indentation for the most common text-based file formats, such as C, C++, XML, SVG and \LaTeX . The editing features range from basic to extremely advanced, however since the user interface was designed long before the definition of today's defacto standards for copy, paste and save, most novice users experience a very steep learning curve. Learning the basics can be very rewarding since the continuous learning of commands and tricks will make code writing and editing both fast and fun.

Emacs automatically identifies the format based primarily on the file suffix, so to get correct highlighting and indentation, a unique and correct suffix must be used. Header files for C++ files are sometimes suffixed '.h'. This is an ambiguous suffix, so the alternative suffixes are recommended such as .hh or .hpp.

5.2.1 Emacs Commands

Here is a list of some of the most important commands used in Emacs. Other software sometimes accept the some of these commands, for example Bash. The table below uses a common notation for key combinations where C means Ctrl key and M means Meta key, which typically is the Alt key but can sometimes be the Win key or the Mac key. These are then combined with other keys so C-c C-w means first hold down Ctrl and press c then hold down Ctrl and press w. In Emacs nomenclature a *file* is on disk and when you open it you can edit the data in a *buffer*, which can in turn be written back to the *file*. Also, what you copy or cut from your buffer are pushed to a stack called *kill ring*, which means that you can copy many things in succession and later paste all of those things in any order, or even mark an area to undo edits only in that area.

Command	Description
C-g	Cancel the current command or shortcut
C-_	Undo last edit (in the selected region if selection is active)
C-a	Go to beginning of line
C-e	Go to end of line
M-a	Go to beginning of sentence (or equiv. in code)
M-a	Go to end of code line (or equiv. in code)
C-M-f	Go forwards to matching parenthesis, bracket, curly bracket
C-M-b	Go backwards to matching parenthesis, bracket, curly bracket
C-x C-c	Close Emacs
C-x C-f	Open file to buffer
C-x C-s	Save buffer to file
C-x C-w	Write to file (Save buffer as)
C-s	Search forward
C-r	Search backward
C-%	Search and replace
C-<space>	Set selection start
M-w	Copy selection
C-w	Cut selection
C-k	Cut to end of line (kill)
C-y	Paste last copy or cut (yank)
M-y	Replace pasted data with next entry in the kill ring
M-z <char>	Cut to first occurrence of <char> (zap)
M-x	Open Emacs command prompt: for anything you don't have a shortcut for

A longer, but still incomplete, list of commands can be found online, for example on the following address:

<http://home.uchicago.edu/gan/file/emacs.pdf> Also, it is possible to remove, replace or add shortcuts to customize Emacs for your own purposes. It is, however, not recommended to change the default shortcuts but try instead to add functions to unused key combinations.

5.3 Eclipse

Eclipse is not just an editor but an IDE, an *Integrated Development Environment*, available for many different platforms and supporting many different programming languages. The system has straightforward tutorials to help new users to get started and a large set of powerful development tools, such as integrated debugger and real-time static code analysis.

6 Other Software

The laboratory computers are filled also with other free software, available for your convenience. You may, for example, need to draw images for CG textures, or figures for your reports and other documents, or write reports, etc. The first four applications in the list use standard formats that facilitates the move between platforms and software systems.

The GIMP One of the most powerful photo editors available, with a large tool set and lots of plug-ins available online.

Inkscape An easy to use, yet versatile vector image editor based on the SVG format.

LibreOffice One of the most complete Office suites. This suite started as a fork of OpenOffice after their owner failed to fully open up the development process for the community.

LaTeX An automatic typesetting system that takes simple text files with formatting tags to generate production grade documents.

Kdenlive A non-linear video editing software.

Scribus Layout and publishing software.