

Version Control Systems

Offensive and Defensive Tool Construction

Table of Contents

[Objectives 2](#_Toc476812815)

[Background Reading 2](#_Toc476812816)

[Part 1: Review Basic Linux Commands 3](#_Toc476812817)

[Part 2: Install a Text Editor on Linux 3](#_Toc476812818)

[Part 3: Install Git on Linux 4](#_Toc476812819)

[Part 4: Install Git on Windows 6](#_Toc476812820)

Offensive and Defensive Tool Construction

Version Control Systems

Objectives

This lab will focus on the following objectives:

* Review the functionality and behaviour of a command line interface (CLI).
* Install and test the Git version control system.
* Apply Git to code development.

Background Reading

* <https://en.wikipedia.org/wiki/Bash_%28Unix_shell%29>
* <http://www.tldp.org/LDP/Bash-Beginners-Guide/html/>
* <https://tiswww.case.edu/php/chet/bash/bashref.html>
* <https://www.gnu.org/software/bash/>

# Part 1: Review Basic Linux Commands

This section of the lab will demonstrate your ability to use basic Linux commands.

1. Review the following list of common commands.

* ls – list contents of a directory
* cd – change directory
* pwd – show current working directory
* mv – move file
* rm – remove file
* mkdir – make directory
* rmdir – remove directory

1. In your home directory, create a directory called **FoodMenu**.
2. In that directory, create two files named **drinks.txt** and **snacks.txt**, and enter few lines into each file.
3. Create new directory called **$HOME/newmenu**, and then copy both text files there.

**Note:** Characters’ case (lower- or uppercase) matters on Linux.

# Part 2: Install a Text Editor on Linux

Most systems have two text editors, “vi” and “nano,” preinstalled. However, a powerful text editor called “Emacs” can also be easily installed. Emacs is specifically designed for programming, and we will use it for the remainder of this lab.

A reference card can be found at: <https://www.gnu.org/software/emacs/refcards/ps/refcard.ps.gz>

Emacs uses two special keys. The CTRL key is denoted as “C-” and the ESC key, called the “meta” key, is denoted as “M-.”

There are two types of commands in Emacs: short key sequences for frequently used commands, and longer commands which are typed in. You can see a few examples in the following table:

|  |  |
| --- | --- |
| **Key Sequence** | **Command** |
| C-x C-f | Read file into Emacs buffer |
| C-x C-c | Exit emacs |
| C-x C-s | Save file in the buffer to disk |
| C-x s | Save all bufers to disk |
| C-x I | Insert contents of another file into this buffer |
| M-x replace-string | Search and replace every occurrence of a string |
| M-x mark-whole-buffer | Next command will operate on the entire buffer |
| M-x sort-lines | Alphabetically sort selected lines |

**Note:** To enter non-printing ASCII characters, use the sequence **C-q C-something** (e.g., the command for a new line is **C-q C-j**).

1. Install Emacs, by entering the following command:

sudo apt-get install emacs24

1. Use Emacs to open your file for editing using the basic command:

emacs some-file-name

1. Once the file is opened in the Emacs buffer, type in some content (e.g., a short paragraph describing why you like this course).
2. Save the file you created, display its content on the screen using the **shell cat** command, and then show your work to the instructor.

Instructor Sign-off: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Part 3: Install Git on Linux

In this class, we will be using two repositories. One is a read-only repository for the whole class, containing code examples, code written in previous classes, and so on. The other is your personal repository, where you have read/write access. Use this repository to save your work in progress, as well as backup copies and copies of your work’s development history (laptops do fail, and “my computer crashed” is not an acceptable excuse).

We will make the access to repositories as automated as possible, using public key cryptography.

1. If a Git client isn’t already installed on your computer, install it using:

$ sudo apt-get install git

1. Then, generate a pair of keys using:

ssh-keygen -t rsa

1. Press ENTER when prompted to enter a password.

This command generates two files containing two keys in the subdirectory of your HOME directory, called **.ssh**. The files are **id\_rsa** and **id\_rsa.pub**.

1. To enable the instructor to create your personal repository (or “repo”), copy the **id\_rsa.pub** file to your personalized file using the following naming scheme. If your name was Fred Flint Stone, and your SAIT email address was fred-flint.stone@edu.sait.ca, execute the command:

cp id\_rsa.pub fredflint-stone.pub

**Note:** Delete any hyphens in the email, and replace the period in the email with a dash.

1. Your instructor will tell you how to submit the file.

Instructor Sign-off: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Next, make a directory to store the repositories on your computer using:

cd

mkdir git

cd git

git clone gitoli@itss.ict.sait.ca:itsc203

git clone gitoli@itss.ict.sait.ca:fred-flintstone

Two directories are created: **/home/user/git/itsc203** and   
**/home/user/git/fred-flintstone**.

Instructor Sign-off: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Copy a file called **test.txt** from the class repo to your own repo using:

cp itsc203/drinkstemplate.c fred-flintstone/drinks.c

1. Edit it and save it using:

cd fred-flintstone

emacs drinks.c

1. Complete the program, so when it is executed, it prints on the console the line: My favorite beverage is [fill in your own].
2. Save the file using the Emacs **C-x C-s** command, and then exit Emacs using the   
   **C-x C-c** command.
3. Track this file using the **git add drinks.c** command, and then commit your changes to your local copy of the repository by entering:

git commit -a -m “[Write your own short commit message]”

1. Push the changes to the Git server using the **git push origin master** command, and then compile the program by entering:

gcc -g -o drinks drinks.c

1. Execute the program using the drinks command, and then confirm that the output appears the way you intended.

Instructor Sign-off: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Part 4: Install Git on Windows

Google “git for windows” for a general understanding of what it is.

1. Download Git for Windows using the appropriate link listed below.

* <https://git-scm.com/download/win/>
* <https://github.com/git-for-windows/git/releases/download/v2.10.1.windows.1/Git-2.10.1-32-bit.exe>
* <https://github.com/git-for-windows/git/releases/download/v2.10.1.windows.1/Git-2.10.1-64-bit.exe>

1. Install Git, and ensure that the following default components are selected:. Note that a number of Microsoft warnings appear.

* Git Bash Here
* Git GUI Here
* Associate .git\* configuration files with the default text editor
* Associate .sh files to be run with Bash

Notes:

* Uncheck the *Enable file system caching* option (it is enabled by default).
* This uses the MinGW environment.

1. When the installation is complete, launch Git Bash from the installation window or from the Start menu. A Unix-like terminal window opens where Unix-like commands can be typed.

Instructor Sign-off: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Install a graphical user interface (GUI) for Git from <https://tortoisegit.org>. Click the **Download** button and choose the appropriate TortoiseGit installer.<https://download.tortoisegit.org/tgit/2.3.0.0/TortoiseGit-2.3.0.0-32bit.msi>

1. On the *Choose SSH Client* screen, select the **OpenSSH, Git default SSH Client** option.

Instructor Sign-off: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Create a public/private key pair for this computer (or virtual machine) by launching the Git shell:

ssh-keygen -t rsa

This saves keys to **/c/Users/user/.ssh/id\_rsa**.

**Note:** Do not use a passphrase.

1. Copy the file **id\_rsa.pub** to firstname-lastname.pub, for example:

cp id\_rsa.pub fred-flintstone.pub

1. Your instructor will tell you how to submit the file electronically, either by submitting it on the web or using a USB key. For example, provided you have exported your Linux file system as drive L:, copy the key to the L: drive and then email it to your instructor..

cp fred-flintstone.pub /L

cd /c/git

git clone gitoli@itss.ict.sait.ca:itsc203

git clone gitoli@itss.ict.sait.ca:fred-flintstone

Instructor Sign-off: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Repeat the steps in Part 3: Install Git on Linux, this time using the file **snacks.c** from the class repository instead.

Instructor Sign-off: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_