Dalhousie University
CSCI 2132 — Software Development
Winter 2017
Lab 1, January 12/13

In this lab, you will learn how to start using UNIX and some basic UNIX commands. I will explain some of these commands and the concept of file and directories in details in the next few lectures; it will be helpful if you follow the instructions carefully to get some hands-on experience now. You will also be asked to write a simple Java program on UNIX. The purpose is not to improve your Java programming skills which you have already gained in previous courses, but to help you get ready to program in C on UNIX. You will also learn how to submit your work electronically using a UNIX command, which is required for every assignment of this course.

Be sure to get help from teaching assistants whenever you have any questions.

1. Login to server bluenose.cs.dal.ca via SSH from a CS Teaching Lab computer or from your own computer.

   You are free to choose to login to the above server from any operating system that you use, including Linux, Windows and Mac OS. The SSH terminal `PuTTy` has already been installed on Windows machines in any teaching lab, which you can also download and install on your own PC.

   When performing this task, please make efforts to avoid entering an incorrect password, because, for security reasons, **the server will disconnect all users in the lab when there are too many failed login attempts**. If you forget your password, please ask the staff at the Help Desk to reset your password.

   To use `PuTTy`, first click its icon. This will open a `PuTTY` configuration window. In the Host Name (or IP address) field of this window, enter `bluenose.cs.dal.ca`, and click on the "Open" button. Then the new dialog will ask for your username (for this server, it would be your CSID) and password.

   You need not install anything on Mac because there is already an SSH client software called `Terminal.app` shipped in the Mac OS. You can open the applications folder to find it. You might have to open this folder in `Finder` so that you can search for `Terminal`. Then you can type in the `ssh` command given in class to connect to bluenose.

2. Now, by default, you are in your home directory (or home folder). In other words, after you login, by default, the current working directory is your home directory. For a typical UNIX command, when you give the name of a file without specifying its

location, the UNIX command will try to locate the file in the current working directory. Enter the command `pwd` to show the path name of your current working directory. Memorize the output of this command displayed in your terminal.

3. Enter the following command to make a new directory in your working directory for your CSCI 2132 course work: `mkdir csci2132`.

   Enter the command `ls` to display the files (including subdirectories) in your working directory. Do you see the name of the directory you just created?

   Now, enter the command `chmod go-rx csci2132`. This command will prevent other users from entering and seeing the content of this directory. You will learn what exactly the argument of the command given here means when we learn file permissions in class.

4. Next, enter the command `cd csci2132` to change your current working directory to the directory you created in question 3. Which command, among those learned in previous questions, can you use to verify that you have performed this task successfully?

5. Create a directory with the name `lab1` in your current working directory, and change the current working directory to the directory created in this question. Verify that you have performed these tasks successfully.

6. Now we are going to write a Java application on UNIX using UNIX utilities and applications. This Java program will be the typical "Hello world!" program which prints a line saying "Hello, world!".

   The first step is to write the source code.

   We will use `emacs` to write the source code. Read pages 69-75 in the UNIX textbook to learn how to use `emacs`, and then create the Java source file. If you do not have the textbook yet, read the following longer tutorial (read enough till you feel comfortable about using `emacs` for a small program):

   `http://www2.lib.uchicago.edu/keith/tcl-course/emacs-tutorial.html`

   The Java class in your code should be named HelloWorld. Use `HelloWorld.java` as the name of the source file.

   There is another UNIX editor called `vi` that can do the same job. If you are already familiar with `vi`, but do not know how to use emacs, you can choose to learn `emacs`, or to use `vi` (or `vim`) instead.

   The official editor of this course is `emacs`. Therefore, we will provide support for any questions regarding `emacs`. If you choose to use `vi`, then whenever there is a question in a lab that asks you to perform a task using `emacs`, perform the same task using `vi` instead.

If you do not know how to use `emacs` or `vi`, but know how to use other UNIX editors such as `nano` and `pico`, do learn to use `emacs`, which is more suitable for serious programming than `pico` and `nano`.

7. Exit emacs, and enter `javac HelloWorld.java` to compile your code. If there are compile errors, edit your source code again to fix them.

   Once your code has been compiled successfully, enter `java HelloWorld` to execute your Java Application.

8. Most of the time, we use hot keys in `emacs` to perform tasks, in order to be more productive. At the same time, `emacs` does have a menu, which is handy when we perform tasks that we run infrequently, as it is easy to forget hot keys for them. We now learn how to do search and replace using the menu of `emacs`. To do this, we make a copy of the file `HelloWorld.java`, and change it to a program that prints "Hi, world!" by following the instructions below:

   (i) Enter the command `cp HelloWorld.java HiWorld.java` to create a copy of `HelloWorld.java` and name it `HiWorld.java`. The two arguments of this command specify the source and then the destination files.

   (ii) Open `HiWorld.java` using `emacs`.

   (iii) Press the `F10` key, if you are using a UNIX or PC keyboard, to bring up the `emacs` menu. If you are using a Mac keyboard, press `Fn` and `F10`, i.e. press `F10` while holding `Fn` down.

   (iv) Look at the menu items. You can see that if you press `e`, you can bring up the sub-menu for the `edit` item. Do this.

   (v) You will see another menu. Choose the `Replace` item.

   (vi) In the new sub-menu, choose `Replace String`.

   (vii) Follow the instructions and replace all occurrences of "Hello" to "Hi". This shall change both the class name and the text to be printed by this program.

   (viii) Save, compile and run this new program to verify that we have successfully completed this task.

   When you see the `Search` menu, you can see that the hot key for `Replace` is Meta followed by the percentage character. The meta key is a key on a UNIX keyboard. On PC, you can use the Escape key. Therefore, you can also do query replace by pressing the escape key followed by the percentage character.

9. Now let's learn how to submit our work. We are going to submit a folder named `lab1` which contains two files: `HelloWorld.java` and `HiWorld.java`, without submitting any other files.

   Read the instructions about the `submit` command before you start:

`http://web.cs.dal.ca/~mhe/csci2132/assignments.htm`

Follow these steps:

(i) Your current working directory is supposed to be `lab1`. Change your current working directory to `csci2132` by entering `cd ..` as the command line. In class we will learn that the two dots in this command refers to the parent directory of the current working directory.

(ii) Rename the directory `lab1` to `lab1.bk`. Use the command `mv lab1 lab1.bk` for this renaming task. This step is recommended; as you will see soon, this directory will store a backup of our work. Use the `ls` command to verify that this task has been done successfully.

(iii) Now, in the `csci2132` directory, there is no subdirectory called `lab1`. Create a directory called `lab1`.

(iv) Use the UNIX command `cp` to copy both `HelloWorld.java` and `HiWorld.java` from directory `lab1.bk` to `lab1`. Here we need use relative pathnames. For example, to copy the former file over, we can enter one of the following two commands: `cp lab1.bk/HelloWorld.java lab1`

`cp lab1.bk/HelloWorld.java lab1/HelloWorld.java`.

Make sure to use a similar command to copy the other file over.

(v) Check the content of the directory `lab1` to make sure that both files have been successfully copied. This step is **very important**.

(vi) Make sure that your current working directory is the directory `csci2132` in your home directory. Which command can you use to verify this?

(vii) Type `submit lab1` to submit the subdirectory `lab1`. The submit command will print error messages if this is not done successfully.

10. By now, you have finished the required work of this lab. Although you will learn more about `emacs` in future labs, it is a good idea to learn more about it now. Practice using emacs yourself. Make sure that you have read both the textbook and the tutorial given in step 6. Then create bigger files using emacs. You could write longer java programs and run them on UNIX, and/or even write a long diary.