Dalhousie University
CSCI 2132 — Software Development
Winter 2017
Lab 2, January 19/20

In this lab, you will first learn autocompletion, a feature of the Bash shell. You will also learn how to use the command `head`. You will then learn more about using `emacs`. In particular, we will learn some hot keys for basic common editing features. Finally, you will learn three UNIX filters: `uniq`, `sort` and `cut`.

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

1. First, perform the following steps to get started:

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

   (ii) Change your current working directory to the `csci2132` directory created in Lab 1.

   (iii) Create a subdirectory named lab2.

   (iv) Change your current working directory to this new directory.

2. Autocompletion is a feature of the Bash shell, which is the default shell program on bluenose that interprets our command lines. This is a very handy feature: it can complete a filename and command name (and more) that you have begun typing if you have typed enough to uniquely identify it.

   To try this, follow the instructions below to copy `~\csci2132\lab1\HelloWorld.java` to the `lab2` directory:

   (i) In the terminal, type `cp ../lab1/Hel`

   (ii) Press the tab key. Then, if `HelloWorld.java` is the only file in the `lab1` directory whose name begins with `Hel` (it should be the case if you followed the instructions of lab1 precisely), the shell completes the filename on the command line so that you will see `cp ../lab1/HelloWorld.java` in your command line.

   (iii) If there is another file whose name starts with `Hel`, say, `HelloWorld.class`, in directory `lab1`, then the shell will only complete the filename till the character before the position at which these two filenames differ. That is, you will see `cp ../lab1/HelloWorld.` in your command line. In this case, type `j` and press tab again.

(iv) Complete the command line `cp ../lab1/HelloWorld.java .` by typing the remaining characters. Press enter. This will copy the file `HelloWorld.java` from directory `lab1` to `lab2`.

After you try this, you can see that this feature is quite useful when you type commands that process files with long names. This can also be used to autocomplete directory names.

3. In this question, we will learn how to display the first three lines of `HelloWorld.java` using a command named `head`. You are asked to use the UNIX `man` command to get the UNIX manual page for `head`, and find out how to perform the above task. Check the description of this command, and read what the option -n does. This manual page may however be confusing for beginners. To help you understand better, read the following web page as well, and compare it with the manual page to understand the format of the manual page: `http://en.wikipedia.org/wiki/Head_%28Unix%29`

   Note that there is another way of using `head` to perform the same task. Read page 30 of the UNIX textbook. Even though it works on bluenose, it is obsolete and might not be supported in the future.

4. Now we learn `emacs` hot keys that move cursors. The two most commons ones are those that move your cursor to the beginning / end of a line.

   Make sure that you are in the `lab2` directory. Use `emacs HelloWorld.java` to open the java source file. Hint: Use autocompletion again.

   Practice the following two hot keys till you memorize them:

   (i) Control-a: Moving to the beginning of a line;

   (ii) Control-e: Moving to the end of a line.

5. We next learn how to do copy and paste in emacs.

   Follow these instructions:

   (i) Go to the second line of your program. Move your cursor to the beginning of this line.

   (ii) Press Control-@ to mark the first character of the block of text that you wish to copy.

   (iii) Move your cursor to the end of the third line.

   (iv) Press Esc-w to copy the above block of text (on a UNIX keyboard, this should be Meta-w...should you use a physical UNIX terminal or UNIX workstation/server in the future) to memory.

   (v) Move your cursor to the line below the last line of your program.

(vi) Press Control-y to paste the above block of text.

Practice this a few more times. Use a different block of text each time you try.

6. The process of cutting and pasting is very similar, and the only difference is that in step (iv) of question 4, we use Control-w instead of Esc-w. Practice this a few times by cutting and pasting a few text blocks of your choice.

7. The `emacs` hot key for undo is Control-x u. That is, type Control-x first, release both keys, and then type u (u means undo). Try this a few times.

8. The last `emacs` hot key that we learn in this lab is Control-k, which deletes the characters in one line starting from the cursor till the end of the line. Move your cursor to the middle of an arbitrary line of code, and try this.

9. A filter is a program that gets most of its data from `stdin` and writes its main results to `stdout`.

   In this lab, we will learn some filters. The first filer that we will learn here is the command `uniq`. It can be used to display a file with all of its identical **adjacent** lines replaced by a single occurrence of the repeated line.

   To try this out, use `emacs` to create a file named `halloffame` in directory `lab2`. This is a text file containing exactly 5 lines as shown below:

```
Alan Turing
Isaac Newton
Isaac Newton
Albert Newton
Albert Newton
```

   Use the `wc` command to verify that this file has exactly 5 lines.

   The syntax of `uniq` is `uniq [inputfile [outputfile]]`. The square brackets mean that parameters are optional. When they are not present, `uniq` reads from `stdin` and writes to `stdout`. The nesting of brackets means that if you only supply one file name, it will be used as the input file.

   Perform the following tasks:

   (i) Type `uniq halloffame`. The output should be 3 lines.

   (ii) Type `uniq -c halloffame`. This -c option causes each line of output to be preceded by the number of occurrences that were found.

   (iii) Type `uniq -f 1 halloffame`. The output should be two lines only. Use the UNIX `man` command to find out why. This is equivalent to `uniq -1 halloffame`. Read page 88 of the textbook to see what it means.

Make sure that you understand what this command does before moving to the next question.

10. We have seen the command `sort` briefly in class before. Let's learn more about it.

    This command sorts the lines of a file based on one or more sort fields (keys).

    The basic usage is `sort [filename]`. This uses the entire line as the sort key. Try this out by typing `sort halloffame`. By default, this sorts all lines in lexicographic order (i.e., the order in which words appear in dictionaries).

    This command can also be used to sort a file by a field in each line. In the example above, each line of the file has a set of words (fields) separated by white spaces. There can be multiple space characters between fields. The fields of each line are numbered 1, 2, ..., from left to right. The -k option can be used to sort by a particular field.

    (i) To sort the names in `halloffame` by first names, type `sort -k 1 halloffame`. To sort by last names, try `sort -k 2 halloffame`.

    (ii) To sort by last name, and then by first name, type `sort -k 2 -k 1 halloffame`.

    (iii) `sort -k n,m halloffame` can be used to sort using fields between (and including) fields n and m as sort keys. Try `sort -k 1,2 halloffame`.

    (iv) In (iii), when `,m` is absent, sort will use fields `n`, `n+1`, ..., until the end of the line as sort key. Thus, for file `halloffame`, `sort -k 1 halloffame` is equivalent to `sort -k 1,2 halloffame` since this file has two fields. If it has more fields, and you would like to perform the task of step (ii) (to sort by last name, and then by first name), you are expected to use `sort -k 2,2 -k 1,1 halloffame`. Try these commands and understand what they do.

    Note that the textbook uses a different syntax for sorting with a particular filed. It is however obsolete, even though it still works on bluenose.

    The -r option can be used to sort in reverse order. Try `sort -r -k 2 halloffame`.

11. Fields in a file can be separated by other characters other than spaces. For example, in a comma-separated values (CSV) file the data items are separated by a comma. Create a file named halloffame.csv, which contains the following 5 lines:

```
Alan,Turing
Isaac,Newton
Isaac,Newton
Albert,Newton
Albert,Newton
```

    Verify that this file has 5 lines.

To make the `sort` command work with the above file, the option -t can be used to specify the deliminator, i.e. the character that separates the fields in each line. The deliminator should be provided immediately after -t in the command line.

Try the command `sort -t, -k 2 -k 1 halloffame.csv`. What does this command do?

12. We are not done with the command `sort` yet. Recall that this command sorts keys in lexicographic order. However, sometimes we wish to sort a key in numerical order, or even sort month names (say, Jan is before Dec). The -n and -M options are used for the above purposes.

    Create a file named `holidays` whose content is (make sure that there is only one space character between any two consecutive fields in the same line; this is required to use the `cut` command in a later question):

    ```
    Dec 26 Boxing Day
    Dec 25 Christmas Day
    Jan 1 New Year
    ```

    Verify that this file has 3 lines.

    Now let's sort the lines by date. This can be done using the following command line:

    `sort -k 1,1M -k 2,2n holidays`

    Try this command and fully understand the options.

13. The `cut` command can be used to extract fields from each line of the input.

    The syntax is `cut -d deliminator -f fields [filename]`.

    Here deliminator is a single character (this can be put inside double quotes, especially if it is a space character). The fields part can be a list of field numbers separated by commas (this is different from `sort`, as commas here are not used to specify a range of field numbers).

    Enter the command `cut -d " " -f 2,4 holidays`. What does this command do?

14. By now, you have finished the required work of this lab. Assignment 1 is due at 3:00PM next Wednesday. Work on it if you have not finished it yet.