

Dalhousie University
CSCI 2132 — Software Development
Winter 2017
Assignment 2

Distributed Wednesday, January 25 2017.

Due 3:00PM, Wednesday, February 1 2017.

Instructions:

1. The difficulty rating of this assignment is *bronze*. Please read the course web page for more information about assignment difficulty rating, late policy (no late assignments are accepted) and grace periods before you start.
2. Each question in this assignment requires you to create one or more regular files on bluenose. Use the exact names (case-sensitive) as specified by each question.
3. Create a directory named **a2** that contains the following files (these are the files that assignment questions ask you to create): **a2q1.txt**, **a2q2_a.txt**, **a2q2_b.txt**, **a2q2_c.txt**, **a2q2_d.txt**, **a2q3_a.txt**, **a2q3_b.txt**, **a2q3_c.txt** and **a2q3_d.txt**. Submit this directory electronically using the command **submit**. The instructions of using **submit** can be found at:

`http://web.cs.dal.ca/~mhe/csci2132/assignments.htm`
4. Do NOT submit hard copies of your work.

Questions:

1. [6 marks] Give concise answers to the two questions below. Use either **emacs** or **vi** to type your answers in one single plain ASCII file, with **a2q1.txt** as its name. Each line of this file should have at most 80 characters.
 - (i) [3 marks] Suppose that we have created a soft link to an existing target file. We then use the **rm** command to remove this target file. What will happen if we attempt to access the target file using the soft link?
 - (ii) [3 marks] The UNIX **which** command can be used to show the full pathname of the executable program of a UNIX command. For example, when you enter **which grep**, you will see **/bin/grep**. This means that the code of the **grep** program is in the executable file **/bin/grep**. When you execute the **grep** command without specifying any arguments, the command will print out the syntax of the **grep** command.

One student copied the executable file `/bin/grep` to his home folder, and named the copy `mygrep`. He then changed his current working folder to his home folder. Both of the above steps were performed correctly.

However, when he entered the command line `mygrep` using his terminal to execute the copied file, he got the following error message:

```
-bash: mygrep: command not found.
```

Explain why he got such an error message.

In your answer, clearly state the relevant shell functionality and give enough reasoning to explain why.

2. [12 marks] This question asks you to use regular expressions.

For each of the following questions, develop a **single** command line to perform the task required, and use `script` to record your most successful execution of the command line in a file.

Your command lines must work no matter what the current working directory is.

The names of the files to be generated by `script`, for the four questions below, are `a2q2_a.txt`, `a2q2_b.txt`, `a2q2_c.txt` and `a2q2_d.txt`.

You are only allowed to use the following UNIX commands: `grep`, `ls`, `cat` and `wc`. You will lose half the marks if you use a command not in this list. Your solutions need not make use of all these commands.

The first three questions below ask you to search for English words in a dictionary file. Each line of this file contains an English word. This dictionary file can be found at `/usr/share/dict/linux.words` on bluenose.

- (a) Print all the words in this dictionary file that start with `legend`. This includes the word `legend` itself.
- (b) Print all the words in this dictionary file that start with `thank` and end with `ly`.
- (c) Print a single integer that is the number of 9-character words in this dictionary file that start with `system` and does NOT end with character `u`, `v`, `w`, `x`, `y` or `z`. For example, `systemist` is such a word, but `systematy` or `systemwide` is not. **Note that you are required to print the number of such words.**
- (d) List all the files in `/usr/bin` whose filenames contain the word `tree` and do not contain any hyphens, including the file `tree`. For example, `pstree` and `pstree.x11` are such files, but `gconf-merge-tree` is not.

3. [12 marks] This question asks you to use UNIX commands to process a text file which contains student grades and personal information of a course. The pathname of this file on bluenose is `/users/faculty/prof2132/public/grades.txt`

In this file, each line corresponds to a student enrolled in a course. Each line has four fields, and they are, from the first field to the last, banner ID, first name, last name and numerical grade of a student. A single space character is used to separate two fields. The first character of each first or last name is capitalized, and the rest are lowercase characters.

For each of the following questions, develop a **single** command line to perform the task required, and use **script** to record your most successful execution of the command line in a file.

These command lines must work no matter what the current working directory is. They must also work even if we change the content of the above file, as long as we do not change its format.

The names of the files to be generated by **script**, for the four questions below, are **a2q3_a.txt**, **a2q3_b.txt**, **a2q3_c.txt** and **a2q3_d.txt**.

You are only allowed to use the following UNIX commands: **sort**, **uniq**, **cut**, **cat**, **wc**, **head** and **tail**. You will lose half the marks if you use a command not in this list. Your solutions need not make use of all these commands.

- (a) Sort the lines of this file by last name and then by first name, in alphabetical order. That is, use last name as the primary sort key, and first name as the secondary sort key, to sort the lines in lexicographically increasing order. For example, the line containing “Marie Andrew” should appear before the line containing “Tim Andrew” in the result, and “Luca King” before “Jonas Leon”. Your command line should print the sorted lines to stdout.
- (b) Sort the lines of this file in decreasing order of numerical grades. That is, highest grade first. Your command line should print the sorted lines to stdout.
- (c) Print a **single integer** that is the highest numerical grade of this course.
- (d) Print a single integer that is the number of distinct numerical grades of this course.

Hint: Some of the exercises in labs 2 and 3 might help. You can ask the instructor or TAs if you have any questions regarding these labs.