CS216: Introduction to Software Engineering Techniques (Summer, 2020)
Lab Assignment 2
(20 points)

Today's Date: Friday, June 12

Due Date: Tuesday, June 16

The purpose of this lab assignment is:

- Getting familiar with Unix/Linux environment
- Practicing more Unix commands such as: **`sort`** and **`grep`**
- Learning the basic Regular Expression

Requirement:

1. Connect to your VM and change your working directory to CS216.

In the terminal window (either from "terminal" application in the GUI-based Linux or from connecting your Virtual Machine through SSH client), make the `CS216` directory (which you created in `Lab1`) your current working directory.

2. Create the directory

Create a directory underneath the `CS216` directory named `Lab2` and make the `Lab2` directory your current working directory.

3. Download the file.

Without opening a browser, use command **`curl`** to download a text file named **`imdb.txt`** from the link (http://www.cs.uky.edu/~yipike/CS216/imdb.txt) and save the file into your current working directory **`~/CS216/Lab2`** by typing the following command at the prompt **`$`**:

```
$curl -O http://www.cs.uky.edu/~yipike/CS216/imdb.txt
```

Please note that the option at the above command is upper case letter O instead of number zero. You will use this text file to search for some movies so that you can generate the output for this lab assignment. You can type the following command at the prompt **`$`** to check the content of the file named `imdb.txt`:

```
$ cat imdb.txt
```

It displays one movie title (with the year made) at each line, in the decreasing order of the number of votes (note that the number of votes for each movie is not included in the file though).

4. Read the manual.

You need to use commands **`sort`** and **`grep`** to finish this Lab assignment. Spend some time to read the manual pages of commands **`sort`** and **`grep`** respectively by typing the following commands at the prompt **`$`**:

```
$ man sort
$ man grep
```

You can also read the reference part at the end of this document for **`sort`** command and lecture notes for **`grep`** command.

5. Solve problems.

The problem you need to solve for this lab assignment is:

> (1) to sort all the movies from `imdb.txt` you downloaded from step 3, in Lexicographical order (dictionary order) by title.
>
> (2) to search from the sorted file in step(1) and find all the movie titles, which start with "**`Star Wars`**" and which were made in between **year 2002 and year 2009** (including 2002 and 2009), and print them to an output file, named **`Lab2output.txt`**.
>
> (3) to search from the sorted file in step(1) and find all the movies which contain "**`land`**" as part of the movie title (Ignore case distinctions) and which were made in **either year 1979 or year 1999**, and print (append) them at the end of the same output file, named **`Lab2output.txt`**.

The following is the sample output of Lab2output.txt:

```
Star Wars: A Musical Journey (2005) (V)
Star Wars: Episode II - Attack of the Clones (2002)
Star Wars: Episode III - Revenge of the Sith (2005)
Star Wars Episode V 1/2: The Han Solo Affair (2002)
Star Wars: Revelations (2005) (V)
Star Wars: The Clone Wars (2008)
Abendland (1999)
A Dog of Flanders (1999)
Backstreet Boys Homecoming: Live in Orlando (1999) (V)
Bear Island (1979)
Clubland (1999)
Drowning on Dry Land (1999)
Heartland (1979)
Hinterland (1999)
In a Savage Land (1999)
Jackson Pollock: Love and Death on Long Island (1999) (V)
```

```
Landspeed: CKY (1999) (V)
Sayehaye bolande bad (1979)
Schlaraffenland (1999)
The Adventures of Elmo in Grouchland (1999)
The Clandestine Marriage (1999)
The Lost Treasure of Sawtooth Island (1999)
The Wacky Adventures of Ronald McDonald: The Legend of
Grimace Island (1999) (V)
Treasure Island (1999/I)
Treasure Island (1999/II)
Wonderland (1999)
```

Note the first **SIX** lines of the sample output is the result from matching the pattern (2): movies starting with "**`Star Wars`**" and made in between **year 2002 and year 2009** (including 2002 and 2009); and the rest part is the result from matching the pattern (3): movies made in **year 1979 or year 1999** and each containing "**`land`**" (ignore case distinctions) in the title.

6.type the following command at the prompt $, then click "Enter" key.

```
$ history | tail -10 >> Lab2output.txt
```

The above command sends the output from the command "**`history`**" to the command **`tail`** as input through the **`piping operator(|)`**, then append the output to the existing file named **`Lab2output.txt`**. It actually appends the latest ten commands you typed from the command line to the end of the file **`Lab2output.txt`**, make sure that it contains the commands you used to generate the output file at step 5. Check the content of your output file **`Lab2output.txt`**, if it does not contain the commands you used at step 5, then adjust the number 10 in the above command, so that the commands you used at step 5 are inside.

Extra (No credit)

If you still have time, please think about based on the matched pattern of step 5(3), how can you separate them into two groups: (i) all the movie titles containing "land" that form whole words; (ii) all the movie titles containing "land" as a sub-string but not the whole word.

Your matched pattern for this Extra part (i) group should be:

```
Drowning on Dry Land (1999)
In a Savage Land (1999)
```

Your matched pattern for this Extra part (ii) group should be:

```
Abendland (1999)
A Dog of Flanders (1999)
Backstreet Boys Homecoming: Live in Orlando (1999) (V)
Bear Island (1979)
Clubland (1999)
Heartland (1979)
```

```
Hinterland (1999)
Jackson Pollock: Love and Death on Long Island (1999) (V)
Landspeed: CKY (1999) (V)
Sayehaye bolande bad (1979)
Schlaraffenland (1999)
The Adventures of Elmo in Grouchland (1999)
The Clandestine Marriage (1999)
The Lost Treasure of Sawtooth Island (1999)
The Wacky Adventures of Ronald McDonald: The Legend of
Grimace Island (1999) (V)
Treasure Island (1999/I)
Treasure Island (1999/II)
Wonderland (1999)
```

Submission

Open the link to Canvas LMS (https://uk.instructure.com/), and log in to your account using your linkblue user id and password. Please submit your file (`Lab2output.txt`) through the submission link for "**Lab2**". You can find your file `Lab2output.txt` from "`Place`", then the directory where you saved your file (if you are under the GUI-based Linux) or transferred from your VM to your local computer via Filezilla, or BitVise, or NoMachine, or scp command☺

Grading (20 points + Bonus 3 points)

1. Attend the lab session or have a documented excused absence. (5 points)
2. Finish step 5 and generate the file named Lab2output.txt correctly. (8 points)
   (if you just copy the output, you get 0 point)
   - generate the matched text lines for pattern 5(2). (4 points)
   - generate the matched text lines for pattern 5(3). (4 points)
3. At the end of the file Lab2output.txt, it contains the command you used to generate the sub-directory named Lab2 under CS216 directory. (1 point)
3. At the end of the file Lab2output.txt, it contains the commands you used to generate the list of matched text lines for patterns 5(2) and 5(3). (3*2=6 points)

Bonus: Demonstrate your program to your TA and answer TA's questions. (3 points)

(Late assignment will be reduced 10% for each day that is late. The assignment will not be graded (you will receive zero) if it is more than 3 days late. Note that a weekend counts just as regular days. For example, if an assignment is due Friday and is turned in Monday, it is 3 days late.)

Reference

Unix sort command

NAME

```
       sort - sort lines of text files
```

SYNOPSIS

```
       sort [OPTION]... [FILE]...
```

OPTIONS

```
   -d, --dictionary-order
       consider only blanks and alphanumeric characters

   -f, --ignore-case
       fold lower case to upper case characters

   -g, --general-numeric-sort
       compare according to general numerical value

    -r, --reverse
        reverse the result of comparisons

    -k, --key=KEY
        sort via a key, for example -k 2 will sort
according to the second column of the text

    -r, --reverse
        reverse the result of comparisons

    -o, --output=FILE
        write result to FILE instead of standard output

    -t, --field-separator=SEP
        fields are defined as anything separated by
whitespace by default. This option will use SEP instead of
whilespace.
```

EXAMPLE

If you have an input file data.txt with the following data:

```
     apple       55
     orange      22
     grape       11
     pineapple   20
     peach       39
```

…and you sort lines by dictionary order:

```
$ sort  -d data.txt
```

you will receive the following output:

```
        apple           55
        grape           11
        orange          22
        peach           39
        pineapple       20
```

However, if you want to sort based on the second field in reverse numeric order, you can use the following command:

```
$ sort –rg –k2 data.txt
```

It will produce the following g output:

```
        apple           55
        peach           39
        orange          22
        pineapple       20
        grape           11
```

Redirection and Piping

Redirection is a mechanism for sending data to and from files.

> redirects std-out to a file

>> appends std-out to a file

< redirects std-in from a file

Piping (|) is a mechanism for sending data from one command (or program) to another. The **piping operator|** feeds the output from the command (or program) on the left as input to the command (or program) on the right.

Example:

```
  $ history | tail -10 >> Lab2output.txt
```