CSCI 274 – Intro to Linux Final
PRACTICE EXAM

Name: _____

1. Answer these questions about your editor of choice (`vim` or `emacs`). `vim` users, assume you are in normal mode.

   (a) (2 points) What keystrokes would you use to delete from the current cursor position to the end of line?

   Solution: Vim users: `d$`
   Emacs users: `C-k`

   (b) (2 points) Assume your cursor is at the beginning of a word. What keystrokes would you use to change the word to `LINUX`?

   Solution: Vim users: `cwLINUX` (optionally, press `Esc` to return to normal mode)
   Emacs users: `M-d` `LINUX`

   (c) (2 points) What keystrokes would you use to replace all instances of `WINDOWS` with `LINUX` in the current buffer?

   Solution: Vim users: `:%s/WINDOWS/LINUX/g`
   Emacs users: `M-x` `replace-string` `↩` `WINDOWS` `↩` `LINUX` `↩`

   (d) (2 points) What keystrokes do you use to close your editor **without saving**?

   Solution: Vim users: `:q!`
   Emacs users: `C-x` `C-c`, then answer `yes`
   or `M-x` `kill-emacs`

2. For each stage of the following pipelines, explain what each stage does, then explain what entire pipeline computes.

> For example:
>
> $$\underbrace{\texttt{cat file.txt}}_{1} \mid \underbrace{\texttt{tr -cd 'aeiou'}}_{2} \mid \underbrace{\texttt{wc -w}}_{3}$$
>
> You might write: *Command 1 prints the file* `file.txt` *to stdout, command 2 deletes things that are not vowels, then command 3 counts the number of words. Overall, this pipeline counts the number of words in* `file.txt` *which have vowels.*

Grading note: You will be given 1 point for each stage of the pipeline you correctly explain, and 2 points for a correct overall explanation.

(a) (5 points) $\underbrace{\texttt{cat file.txt}}_{1} \mid \underbrace{\texttt{grep '\^\$'}}_{2} \mid \underbrace{\texttt{wc -l}}_{3}$

> **Solution:** Command 1 prints the file `file.txt` to stdout, command 2 keeps just empty lines, then command 3 counts the number of lines. Overall, this pipeline counts the number of empty lines in `file.txt`.

(b) (5 points) $\underbrace{\texttt{cat wonderland.txt}}_{1} \mid \underbrace{\texttt{grep -v Alice}}_{2} \mid \underbrace{\texttt{wc -l}}_{3}$

> **Solution:** Command 1 prints the file `wonderland.txt` to stdout, command 2 deletes lines with `Alice`, then command 3 counts the number of lines. Overall, this pipeline counts the number lines without `Alice` in `wonderland.txt`.

(c) (7 points) $\underbrace{\texttt{cat file.txt}}_{1} \mid \underbrace{\texttt{sort}}_{2} \mid \underbrace{\texttt{uniq -c}}_{3} \mid \underbrace{\texttt{sort -n}}_{4} \mid \underbrace{\texttt{sed -n '\$p'}}_{5}$

> **Solution:** Command 1 prints the file `file.txt` to stdout, command 2 sorts the files lines, then command 3 deletes repeated lines and prints the number of times they occur, command 4 sorts the lines numerically, and command 5 prints just the last line. Overall, this pipeline prints the most common line in `file.txt` and the number of times it occurs.

3. Consider the following interaction at the terminal:

```
[jrosenth@toilers:/var/www/toilers]$ ls -l
drwxrwxr-x  6 crader   rmcwic   4096 Jan 16  2010 ccwic10
drwxrwxr-x  3 tcamp    toilers  4096 Oct 31  2009 Code
drwxrwxr-x  2 tcamp    toilers  4096 Apr 12  2007 Contact_us
drwxrwxr-x  2 tcamp    toilers  4096 May 14  2008 Downloads
drwxrwxr-x  2 jrosenth toilers  4096 Oct 12 11:31 Images
-rw-rw-r--  1 tcamp    toilers  3067 Oct 31  2016 index.html
drwxrwxr-x 11 acoles   csgames  4096 Feb 19  2017 jeopardy
-rw-rw-r--  3 tcamp    toilers  2060 Oct 31  2016 navi.css
drwxrwxr-x  5 tcamp    toilers  4096 Apr 17  2015 Publications
drwxr-xr-x  5 jrosenth rmcwic   4096 Aug 24 12:41 RMCWiC
drwxrwxr-x 12 crader   unplug   4096 Aug 12  2013 Unplugged
lrwxrwxrwx  1 jrosenth toilers  18   Oct 12 11:29 toilers.png -> Images/toilers.png
```

(a) (2 points) What is the user of the `Unplugged` directory?

Solution: `crader`

(b) (2 points) What is the group of the `jeopardy` directory?

Solution: `csgames`

(c) (2 points) What **ONE** octal digit represents the permissions of the group of the `navi.css` file?

Solution: `6`

(d) (3 points) How many hard links are there to the inode of `navi.css`?

Solution: 3

(e) (2 points) Is `toilers.png` a hard or soft (symbolic) link?

Solution: Soft/symbolic link

(f) (4 points) Write **one command** to create a soft link called `home.htm` that points to `index.html`.

Solution: `ln -s index.html home.htm`

4. (8 points) Write a `bash` script that behaves like the one in the terminal session below.

```
[jrosenth@isengard:~/counter]$ ls
computers.txt  counter.sh   unix.txt
[jrosenth@isengard:~/counter]$ cat computers.txt
Macintosh SE/30
Macintosh LC II
Macintosh Quadra 840av
Apple IIgs
[jrosenth@isengard:~/counter]$ cat unix.txt
Research UNIX AT&T UNIX SVR4
BSD FreeBSD NetBSD NeXTSTEP Mac OS X
Linux
[jrosenth@isengard:~/counter]$ bash counter.sh computers.txt
There are 4 lines in computers.txt.
There are 10 words in computers.txt.
The most frequent word is Macintosh.
[jrosenth@isengard:~/counter]$ bash counter.sh unix.txt
There are 3 lines in unix.txt.
There are 13 words in unix.txt.
The most frequent word is UNIX.
```

Solution:

```
echo "There are $(wc -l <$1) lines in $1."
echo "There are $(wc -w <$1) words in $1."
echo "The most frequent word is $(<$1 tr ' ' '\n' | sort
                | uniq -c | sort -n | tail -1 | cut -b9-)."
```

5. **True or False?** Write T or F for each of the following statements depending on whether the statement is true or false.
   (a) (2 points) **F** Hard links can link to directories, whereas soft links cannot.
   (b) (2 points) **T** Soft links can cross file system boundaries, whereas hard links cannot.
   (c) (2 points) **F** Hard links link to file names, whereas soft links link to inode numbers.
   (d) (2 points) **F** Soft links will continue to work after the original file moved or deleted.

6. (6 points) Consider the following output from your terminal:

```
$ ps u
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
jrosenth  8813  0.0  0.0  41716  5680 pts/1    Ss   Nov27   0:00 zsh
jrosenth 11561  0.0  0.2 259304 34500 pts/1    Sl+  Nov27   1:33 weechat
jrosenth 25383  0.0  0.0  51588  6488 pts/3    S+   Dec03   0:00 mutt
jrosenth 28609  0.0  0.0  35844  3172 pts/2    R+   07:30   0:00 ps u
```

   Write **one command** to end the `weechat` command's process.

   Solution: `kill 11561`
   Using the flags `-9`, `-15`, `-KILL`, or `-TERM` is OK. Also, `pkill weechat` is another option.

7. (2 points) As a set of octal digits, what `umask` should be used such so that new files created have octal permissions `640` and new directories created have octal permissions `750`?

   Solution: `027`