**CSC 505 Unix / Linux OS Assignment #2**

**I.** **Unix text file related manipulation using advanced Unix commands**

1. Get the 100 most used words in the text of Moby Dick.

<https://www.dropbox.com/s/tvhrps2g58vo6il/moby.txt?dl=0>

Ans :

1. Get the 100 most words in the text of Moby Dick that are not preposi-

tions (use the given list).

<https://www.dropbox.com/s/09pgq4s74brlmj9/prepositions.txt?dl=0>

1. Parse out the text (from HTML) of the provided Glenn Beck Show transcript

<https://www.dropbox.com/s/hmadjz285sj9lgy/Glenn%20Beck.html?dl=0>

* + Get the 100 most used words in the text.
  + Get the 25 words that appear before and after any mention of the following words in text: “democrats”, “obama”, “pelosi”.

1. **Advanced Unix**

**1.** Following dropbox link has a file called tartest.tar. Download

this file into your Linux environment. If you are using an ssh client

from a Windows computer, you may need to use `wget` or winscp (Windows

program) to get the file into Linux. Display the table of content of the

file. Extract the contents of the tartest.tar file (use the verbose

option to tar). Create a new tar file with the same directory tree as

what was extracted from tartest.tar. Use the gzip program to compress

the tar file you just created. Show your session. **Dropbox Link:** <https://www.dropbox.com/s/qenuubc5yva8q09/tartest.tar?dl=0>

**2.** What are the inode numbers of the following files and directories on your

system: / (root of the filesystem), your home directory, /etc/passwd,

/usr/include, and /usr/share/man? Hint: use the `-d` option to `ls` to

display directory entries rather than the contents of the directories.

Show your session.

**3.** Create a symbolic link from your home directory to /etc/passwd and to

/usr/bin. Create a hard link to another file in your home directory.

Use the ``ls -il`` command to display the symbolic and hard link files.

Show your session.

4. First, enter the `who` command to see the output displayed. If a user is

logged in more than once, there may be multiple entries for that person.

Now devise a pipeline sequence of commands to display a sorted list of

people currently logged in with no duplication in names. Note that you

will need to first filter the data vertically to show only the first

column. See the study guide notes on `awk` for how to filter vertically.

Show the commands used.

**5.** Use the `find` command to find any regular files in your home directory

that have the execute bit set for others. Note from the study guide that

the `-perm` option to the `find` command takes a mode argument and that

the mode (either 001 or o+x) is expressed one of: ``-perm o+x``,

``-perm -o+x`` or ``-perm /o+x``

You may want to seed the activity by creating a file with execute bit for

others as follows:

touch testfile

chmod o+x testfile

Show the command used.

**6.** Use the ``crontab -e`` command to schedule the cron facility to run the

`who` command each day at 2pm for the remainder of October. Show the

output from entering the ``crontab -l`` command. Please refer to notes on

the EDITOR environment variable. That will be needed if

you do not feel comfortable using `vi`.

1. **Unix Shell Scripting:**

**1.** Write a Bash shell script which uses the following utilities to display

information (date, whoami, pwd, echo). Before each piece of information

is displayed, print a heading similar to below.

Todays date is:

My login name is:

My home directory is:

My current directory is:

**2.** Write a Bash shell script program that takes a login name as an argument

and outputs the home directory of the user.

**3.** Write a Bash shell script program that takes the name of a file or

directory as an argument and reports if the file is a directory, regular file

or other and also reports if the user has read, write and execute permission on

the file or directory. Use a sequence of if statements on the file name to

determine the information.

**4.** Write a Bash shell script program that takes two integer arguments. The

second argument is assumed to be greater than the first. The output of the

program is a counting of numbers beginning at the first number and ending with

the second number. (Note, this simple script is surprising useful for a number

of activities when working with a Unix system.)

**5.** Write a Bash shell script program that will read a list of numbers from

standard input and the output is the sum of the numbers.

**6.** Write a Bash shell script program that removes all zero length ordinary

files in the directory passed as an optional argument. If you do not specify

the directory argument, the script uses the present working directory as the

default argument. Do appropriate exception handling in your script. Show a few

sample runs of your script.

**Hint** This may take a little more effort (not much more) than other problems we have done. You may want to use a function to help with this problem. There are actually multiple ways to determine if a file is of zero length.

1. see [**find -size**](http://faculty.salina.k-state.edu/tim/unix_sg/advanced/find.html#cmdoption-find-size)
2. see [**set --**](http://faculty.salina.k-state.edu/tim/unix_sg/bash/set.html#cmdoption-set--)
3. [awk](http://faculty.salina.k-state.edu/tim/unix_sg/advanced/awk.html#awk)

! [ -s $file ]

**7.** This one is a little bit trickier than it first appears.

Write a Bash shell script that will rename all the files in a directory. It

will mostly be useful to rename digital picture files. The first argument is a

base name, second argument is a file extension. If it is run as:

$ renumber 1stBirtday jpg

then the resulting files should have names like:

1stBirthday001.jpg, 1stBirthday002.jpg, 1stBirthday003.jpg, etc.

**Hint:** consider the case where some files already exist in the directory that match the filenaming convention. Before renaming a file, make sure that a file with the new name does not already exist. To test for this, run your program once and then add some other files to the directory and run your program again.

**8.** Following problems requires file/folders from following dropbox link:

<https://www.dropbox.com/s/72bfjy3fs0fa23g/hw4_data.tar?dl=0>

1. Create a tar.gz compressed tarball of all files in the folder p1. Call it, p1.tar.gz
2. Use find OR write a bash script to rename all the files whose names begin with “wahle” to begin with “whale” (i.e. we want to fix a typo in the names of multiple files). You can test your work on the contents of the directory p2.
3. Write a bash script called keyword bundle.sh that takes two argu- ments: the first is a keyword to search for, and the second is a name of a tarball to create. Your script will look for all files that contain the given keyword, and create a tarball that is compressed with bzip2 and whose name is the second argument followed by .tar.bz2

**Hint:** Example usage:  
$> ./keyword\_bundle.sh whale whale\_files

will create a file called whale\_files.tar.bz2 that contains all files that have the word “whale” in them.

– grep has a nice flag that can make things very simple for you.

* 1. –  If the script is called with less or more than two arguments, you should print the following message and exit.  
     Correct Usage: keyword bundle.sh <keyword> <bundle name>
  2. –  I will only test cases where there is an actual match (i.e. do not worry about dealing with keywords that are not in any file).

Remember your two best friends are the man tool and your favorite search engine.

**9.** Follow the instructions as stated below and submit your outputs in detail:

1. In your home directory create a directory called hw1 and navigate to that directory

* + 1. Use wget to download the file

<https://www.dropbox.com/s/i9nbobfn1pehi41/hw1download.sh?dl=0>

* + 1. Rename hw1download.sh to hw1run.sh
    2. Make hw1run.sh executable
    3. Create a file hw1file

1. Write

Ww+hexekf=+jwoxawlj+yfi+heai+fyyq[<<ooo?bk?bwuxiaa?ijr<bwrukik<bk(!&(<(!)!hl<yfiixjglzi+lxj+hwaawo+eyk+exkyurbyewxk?

to hw1file.

1. Remove write permissions from hw1file
2. hw1run.sh is a bash script. Once you have changed the permissions as asked above, run it by typing ./hw1run.sh
3. If you have completed the above steps correctly, instructions will be printed on the screen. Follow the printed instructions to complete the assignment question 9.

Remember your two best friends are the man tool and your favorite search engine.