

In this homework, you will need to work on the eniac-l.seas.upenn.edu server. If you need help logging in to this machine see the following webpage: <http://www.seas.upenn.edu/cets/answers/remote.html> (note: you need to swap eniac-l.seas.upenn.edu for eniac.seas.upenn.edu).

In some questions below, you will be asked to enter some command with some set of arguments. Depending on the command, there may or may not be output generated. For each command, you should use your mouse to highlight the command (including the prompt) and the output (if there is any) and copy it to a separate text or word document. Obviously, when you make mistakes, you should not copy the text. You should also copy the question to the text or word document as well.

When you are finished with the assignment, attach the text or word document to an email and send to mcorniss@cis.upenn.edu. The subject of the email should be "cse399 - hw3" (without the quotes). The homework is due by the beginning of Monday's class. Total points: 50.

1. Preliminary.

- (a) Change your location to the 'cse399' directory you created in homework 1 (or create it again if you deleted it). Copy the directory /home1/m/mcorniss/teaching/cse399/hw3 and all its subcontents to your current location. Change your location to this new directory (hw3).

Answer:

```
bash$ cd cse399
bash$ cp -r /home1/m/mcorniss/teaching/cse399/hw3 .
bash$ cd hw3
```

2. [12 Points] Filesystem and access rights.

- (a) Create a symbolic link to directory foo called goo.

Answer:

```
bash$ ln -s foo goo
```

- (b) Create a hard link to file foo-1.txt in directory foo called zoo.

Answer:

```
bash$ ln foo/foo-1.txt zoo
```

- (c) Change the file permissions for directory foo and all its subcontents. Add write permission for owner, group, and others.

Answer:

```
bash$ chmod -R ugo+w foo/
```

- (d) Again, change the file permissions for directory foo and its subcontents adding execute permission for all directories (not files) for owner, group, and others.

Answer:

```
bash$ chmod -R ugo+X foo/
```

- (e) Change the file permissions for file 'cse399-1.txt' to 'rwxr-xr-x' using numeric mode (hint: use man to read about using numeric mode to change file permissions).

Answer:

```
bash$ chmod 755 cse399-1.txt
```

- (f) Change the group for directory foo and its subcontents to 'penn'.

Answer:

```
bash$ chgrp -R penn foo/
```

3. [16 Points] *locate* and *find*.

- (a) Use 'locate' to find all files whose name contains the text 'inetd.conf'. Specify a flag that only lists files that exist at the time of execution rather than at the time the database was written.

Answer:

```
bash$ locate -e inetd.conf
```

- (b) Again use 'locate' to find the files, but this time pipe the output to a command called 'xargs' (man xargs for more information). Use xargs to list each file using the long format.

Answer:

```
bash$ locate inetd.conf | xargs ls -l
```

- (c) Without changing your location, use 'find' to find all directories within directory foo.

Answer:

```
bash$ find foo/ -type d
```

- (d) Use 'find' to find all files (not directories or links) that start with 'cse399' or 'unix' within the current directory or its subcontents.

Answer:

```
bash$ find . -type f -a \( -name cse399\* -o -name unix\* \)
```

- (e) Repeat the find from the previous exercise, but in addition, use '-exec' to list all files using the long format.

Answer:

```
bash$ find . -type f -a \( -name cse399\* -o -name unix\* \) -exec ls -l {} \;
```

- (f) Repeat the find from the previous exercise, but instead of using '-exec', pipe the output to xargs and list all the files using the long format.

Answer:

```
bash$ find . -type f -a \( -name cse399\* -o -name unix\* \) | xargs ls -l
```

- (g) Use 'find' to find all files, but use '-prune' to ignore the directory tmp and all its subcontents.

Answer:

```
bash$ find . -name tmp -prune -o -print
```

- (h) Use 'find' to find all files (not directories or links) that start with 'cse399' or 'unix' and are at least 1 byte in size.

Answer:

```
bash$ find . -type f -a \( -name cse399\* -o -name unix\* \) -a \( -size 1c -o -size +1c \)
```

4. [12 Points] *grep* and *sed*.

- (a) Repeat the find command from the previous exercise, but pipe the output to 'grep' and select only files that end in .txt.

Answer:

```
bash$ find . -type f -a \( -name cse399\* -o -name unix\* \) -a -size 1c | grep "\.txt$"
```

- (b) The file 'sim-file.txt' is the output generated from a simulator (use 'less' to view the file). Use 'grep' along with some other commands we've already learned to extract the 3rd occurrence of the statistic 'sim_num_insn' (and not 'sim_num_insn.sim'). Note that 'sim_num_insn' appears in multiple places, but you should only extract the lines of the form:

```
sim_num_insn                100000000 # instructions simulated ...
```

Answer:

```
bash$ grep "^sim_num_insn " sim-file.txt |head -n 3|tail -n1
```

- (c) Pipe the output from the previous command to 'sed' and use 'sed' (only one sed, don't pipe to multiple sed) to remove all text except the value (in the case above, 100000000).

Answer:

```
bash$ grep "^sim_num_insn " sim-file.txt |head -n 3|tail -n1|sed -e 's/[^0-9]*\([0-9]*\).*\1/'
```

- (d) The file 'data.txt' contains a number of columns of data separated by colons. Use sed to extract only the third column.

Answer:

```
bash$ sed -e 's/[^:]*:[^:]*:\([^:]*\):.*\1/' data.txt
```

- (e) In the file 'sim-file.txt' there are a number of lines that start with "sim: ** starting" and end with "instructions **". Use grep (and no other commands) to filter out only these lines and redirect your output to a newfile, 'sim-file-2.txt'. 'sim-file-2.txt' should contain everything in 'sim-file.txt' except the filtered lines.

Answer:

```
bash$ grep -v "sim: \*\* starting" sim-file.txt > sim-file-2.txt
```

- (f) Use grep (and no other commands) to recursively search for all files that do not contain the text "Linux". Your command should return filenames only.

Answer:

```
bash$ grep -r -L "Linux" *
```

5. [10 Points] New commands: *tar*, *gzip*, *bzip2*.

- (a) Use 'gzip' to compress the file 'large-file.txt'. Find a similar command to uncompress the file.

Answer:

```
bash$ gzip large-file.txt
```

```
bash$ gunzip large-file.txt.gz
```

- (b) Repeat the previous exercise using 'bzip2' rather than 'gzip'.

Answer:

```
bash$ bzip2 large-file.txt  
bash$ bunzip2 large-file.txt.bz
```

- (c) 'tar' is a command for creating archives and extracting files from archives. Use 'tar' (and tar only) to extract all files from 'tarfile.tar.gz'. Note that this file is compressed (using gzip) and needs to be uncompressed. A tar flag will allow you to uncompress the file while simultaneously extracting the files.

Answer:

```
bash$ tar xvfz tarfile.tar.gz
```

- (d) Use 'tar' (and tar only) to create an archive called foo.tar.bz2 of the directory foo and its subcontents. You will need to specify a flag in 'tar' that will compress the resulting file using bzip2.

Answer:

```
bash$ tar cvfj foo.tar.bz2 foo/
```

- (e) Use 'tar' to list the contents of the archive tarfile.tar.gz' without actually extracting the files.

Answer:

```
bash$ tar tvfz tarfile.tar.gz
```

6. About this assignment.

- (a) Approximately, how long did it take you to complete this homework?
(b) Would you classify this assignment as easy, straight-forward, or difficult?