Reginald Edwards
ECON 41701 – Homework I
09 April 2012

Problem 1: Unix Essentials

1. **Determining File types**
   Using the command `ls -sh` I find that the file size is 9.2 megabytes.

2. **Checksums**
   Using `md5sum -c md5.txt` and `sha1sum -c sha1.txt`, both return “OK”.

3. **Tar & Counting Files**
   The expression

   ```
   ls | sed 's/-.*//' | uniq > filetypes.txt
   ```

   gives the different file types and dumps them in a file “filetypes.txt.”

   To get the count of each file type I loop over the lines in the file types file, use the directory listing function with a wildcard to get all of a given file type within that directory, and then pipe that output to the generic word count function with the flag to count lines. I dump the file types and associated counts into the file “filetype-counts.txt.”

   ```
   for fileType in $(cat filetypes.txt)
   do
        echo $fileType >> filetype-counts.txt
        ls $fileType* | wc -l >> filetype-counts.txt
   done
   ```

4. **Solver Exit Codes**
   The model was estimated 5000 times. To find how many times the model was estimated I count the number of times “SNOPTA EXIT” occurs in all of the “BLP.char.T1.J12.pbs.o” output files:

   ```
   for outputFile in $(ls BLP.char.T1.J12.pbs.o*)
   do
     grep "SNOPTA EXIT" $outputFile >> exitcodes.txt
   done

   wc exitcodes.txt -l
   ```

   The model found an optimum 4987 times.

I found this by getting a list of unique exit codes then counting the occurrence of “SNOPTA EXIT   0” in the file of concatenated exit codes.

```
sort ../../exitcodes.txt | uniq > ../../exitcodes-unique.txt

for exitCode in "SNOPTA EXIT   0" "SNOPTA EXIT  10" "SNOPTA EXIT  40"
do
  echo $exitCode
  grep "$exitCode" exitcodes.txt | wc -l
done
```

The exit codes when the solver failed were “SNOPTA EXIT 10 -- the problem appears to be infeasible” and “SNOPTA EXIT 40 -- terminated after numerical difficulties.”

5. **Job Runtimes**

The fastest job finished in 00:00:07 (7 seconds) time. I found this by looping over each file, extracting the walltimes, and passing the walltime to a file “walltimes.txt”:

```
for outputFile in $(ls BLP.char.T1.J12.pbs.o*)
do
  sed -n '/Resources/p' $outputFile | sed -e 's/^.*walltime=//' >> ../../walltimes.txt
done

cat walltimes.txt | sort > walltimes-sorted.txt

head -1 walltimes-sorted.txt
```

The slowest job finished in 2 minutes and 53 seconds. I found it in a similar way to the fastest time:

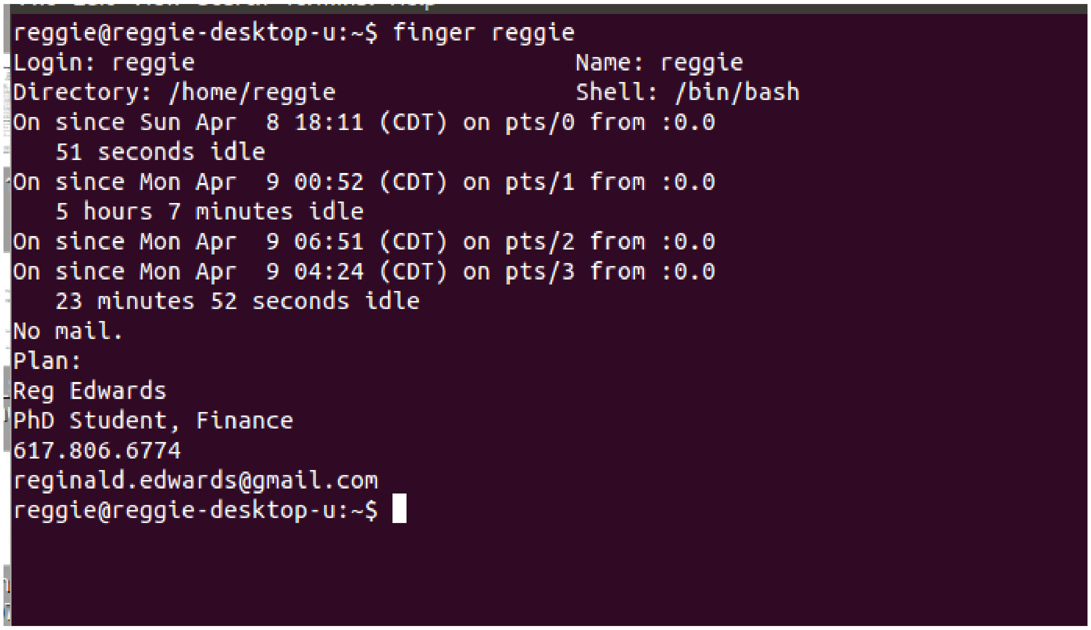
```
tail -1 walltimes-sorted.txt
```

The median execution time was 2 minutes and 8 seconds. I found it by combining `head` and `tail` to get the 51st element in the sorted list of walltimes:

```
head -51 walltimes-sorted.txt | tail -1
```

6. **Aliases & Functions & Shell Scripts**

7. A screenshot of my finger:

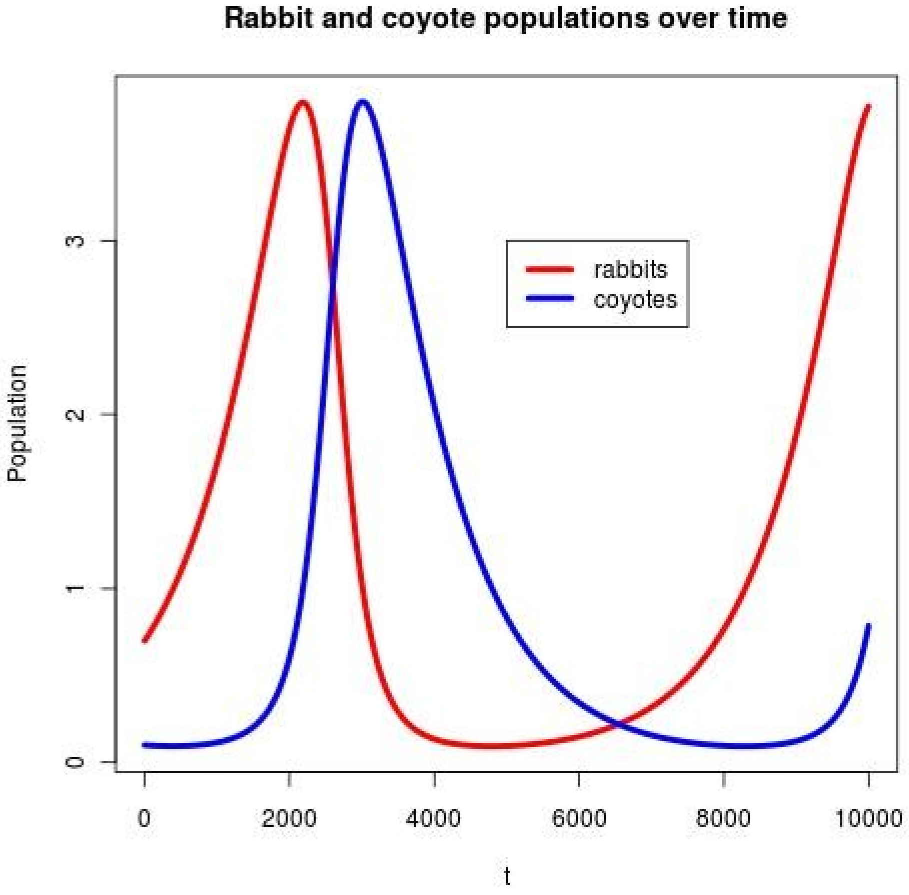


```
reggie@reggie-desktop-u:~$ finger reggie
Login: reggie                           Name: reggie
Directory: /home/reggie                 Shell: /bin/bash
On since Sun Apr  8 18:11 (CDT) on pts/0 from :0.0
   51 seconds idle
On since Mon Apr  9 00:52 (CDT) on pts/1 from :0.0
   5 hours 7 minutes idle
On since Mon Apr  9 06:51 (CDT) on pts/2 from :0.0
On since Mon Apr  9 04:24 (CDT) on pts/3 from :0.0
   23 minutes 52 seconds idle
No mail.
Plan:
Reg Edwards
PhD Student, Finance
617.806.6774
reginald.edwards@gmail.com
reggie@reggie-desktop-u:~$ 
```

Problem 2: Basic Programming: Rabbits and Coyotes

(Code attached)

The initial values are those given in the problem satement.


Rabbit and coyote populations over time
Population
t
rabbits
coyotes