Exam 1: Rework

Tyler Berezowsky

September 28, 2014

1 Problem 1

For the database provided in class, count the number of directories (sessions) that have at least three vowels (defined as "a", "e", "i", "o" and "u" to keep things simple) in the last name and at least 5 letters in the first name of their directory name.

1.1 Solution

```
find -type d | cut -d "/" -f 4 | grep -iE \(.*[aeiou]\)\{3,\}.*_[a-z]\{5,\} | wc -l
```

The name of the last directories is tuh_eeg database is a patient's name. Taking advantage of this, the solution begins by listing every directory in the database via `find`. This essentially creates a list of the patients in the database, but each patient's name is preceded by the directories filepath. To reduce the amount of filtering need, the `cut` command is used to filter out the names from the filepaths by using `/` as the field deliminator. The list of names is then grep'd (new word!) without relevance to case, option `-i`, and extended expressions, option `-E` with the following expression

```
\(.*[aeiou]\)\{3,\}.*_[a-z]\{5,\}
```

The expression is best discussed individually and then assembled. The `\( \)` meta-characters allow one to treat a series of characters as a single element, allowing `.*[aeiou]` to be manipulated by other meta-characters. The `\{3,\}` repeats the preceding character at least 3 times. For this solution `.*[aeiou]` would be repeated. The `.` meta-character stands for any character, and the `*` symbol allows tells grep to look for the preceding character zero to infinite number of times. The final `.*` tells grep the string of `.` with atleast three intermingling vowels can end with any characters until `_` which denotes the beginning of the patients last name. The `[a-z]\{5,\}` simply looks for five or more alphabetical characters which of course would equate to a last name of 5 or more letters. The number of lines left by `grep` is then counted by `wc -l`.

1.2 Results

As can be seen from listing 1 below, the number of individuals with at least 3 vowels in their last name and 5 characters in their first is 40,054.

Listing 1: Terminal view exercising the soultion to problem 1

```
tyler@tyler-Latitude-E6410:~/bin$ tuda
tyler@tyler-Latitude-E6410:~/ece_3822/data/tuh_eeg$ find -type d |
cut -d "/" -f 4 | grep -iE \(.*[aeiou]\)\{3,\}.*_[a-z]\{5,\} |
wc -l
40054
tyler@tyler-Latitude-E6410:~/ece_3822/data/tuh_eeg$
```

2 Problem 2

Write a shellscript that loops over all *_eg_00.txt and accumulates the size of the file in bytes. Compare this to the output of the du command and show that they produce the same result. Use only the book_01 data.

2.1 Solution

2.1.1 Part 1

The solution to part 1 consists of a shellscript illustrated below containing the command `wc $(find -type f) -c` The command `wc` with option `-c` counts the number of bytes of each file placed in its standard input which in this case is `$(find . -type f)`. The `$` character tells bash to execute the enclosed command. This allows wc to calculate the total number of bytes for each file in book_01 when the shellscript e1_p2.sh is run in book_01.

Listing 2: The contents of shellscript e1_p2 for the soluton of problem 1.

```
#!/bin/bash

wc $(find . -type f) -c
```

2.1.2 Part 2

The solution to part 2 is fundamentally the same, except it utilizes the `du` command as suggested. The solution is as follows: `du $(find . -type f) -bc`. Just as part 1, `$(file . -type f)` is used to deliver all the files in book_01 to `du`. `du` by default calculates the size of files, but the options `-bc` had to be set to change the block size to 1 and to tell `du` to sum the results for a total.

2.2 Results

The results of the shell script and the du equivalent can be seen in listings 3 and 4. Note that both total to 15,712,872 bytes for book_01.

2.2.1 Part 1

Listing 3: Output of shell script e1_p2.sh.

```
tyler@tyler-Latitude-E6410:~/ece_3822/data/tuh_eeg/book_01$ e1_p2.sh
...
1266 ./00007018_20130301/Bruender_Shena/eg_00.txt
1365 ./00007018_20130301/Bruender_Shena/eg_01.txt
1007 ./00007018_20130301/Toomey_Lu/eg_00.txt
1365 ./00007018_20130301/Toomey_Lu/eg_01.txt
1805 ./00007018_20130301/Pastorius_Verlie/eg_00.txt
1365 ./00007018_20130301/Pastorius_Verlie/eg_01.txt
15712872 total
```

2.2.2 Part 2

Listing 4: Results of command line eqviulent via du.

```
tyler@tyler-Latitude-E6410:~/ece_3822/data/tuh_eeg/book_01$ du $(find . -type f)
-bc
...
1365     ./00007018_20130301/Madonna_Draft/eg_01.txt
```

```
1266      ./00007018_20130301/Abrom_Ammie/eg_00.txt
1365      ./00007018_20130301/Abrom_Ammie/eg_01.txt
1266      ./00007018_20130301/Bruender_Shena/eg_00.txt
1365      ./00007018_20130301/Bruender_Shena/eg_01.txt
1007      ./00007018_20130301/Toomey_Lu/eg_00.txt
1365      ./00007018_20130301/Toomey_Lu/eg_01.txt
1805      ./00007018_20130301/Pastorius_Verlie/eg_00.txt
1365      ./00007018_20130301/Pastorius_Verlie/eg_01.txt
15712872      total
```

3 Problem 3

ssh into electrodata and after the login prompt open a new bash shell. Generate a list of all processes under your name and explain how they are related through process IDs. Kill the top level process so that all children are killed. Also demonstrate that this logs you out and leaves no processes running. (do the latter by logging in again and see what is running under your name)

3.1 Solution

`ps fx` was used to list all of the processes running under the user with a graphical representation of the process IDs (PID)s and parent process IDs (PPID)s. The PID consists of simply an identification number for the job, and the PPID contains the PID of process which spawned that job. The `x` option instructs `ps` to list only processes running its own effective user ID (EUID), essentially listing only the processes run by the user running `ps`. The `f` option places all the processes in a cute graphical representation which shows the top process and all its children. The graphical representation can be seen in the results under listing 5.

Logged into electrodata, a new bash shell was generated and `ps fx` was run. The top process was found and stopped via `kill` forcing a log-out. electrodata was logged into again and `ps fx` run to insure the opened bash shell was killed.

3.2 Results

As visible in listing 5 the top process is sshd and under that the default bash shell, the opened bash shell(20746), and last the `ps` command. The sshd process (42999) was killed and the local bash shell states that the connection to the server was closed by the remote host. After logging in again and running `ps`, it can be seen the opened bash shell is dead.

Listing 5: The terminal view of the solution for problem 3

```
tud12685@electrodata:~$ # logged in <--
tud12685@electrodata:~$ bash # new bash shell
tud12685@electrodata:~$ ps fx # look at processes with tree because I like visual
  PID TTY      STAT   TIME COMMAND
42999 ?        S      0:00 sshd: tud12685@pts/0
43064 pts/0    Ss     0:00  \_ -bash
20746 pts/0    S      0:00      \_ bash
35216 pts/0    R+     0:00          \_ ps -fx
tud12685@electrodata:~$ # okay, i see the opening remote bash
 shell and sometype of login process (43046,42999). lets
 kill the login process.

tud12685@electrodata:~$ kill 42999
Connection to electrodata.eng.temple.edu closed by
remote host.
Connection to electrodata.eng.temple.edu closed.
tyler@tyler-Latitude-E6410:~$
```

```
tyler@tyler-Latitude-E6410:~$ ssh tud12685@electrodata.eng.temple.edu
tud12685@electrodata.eng.temple.edu's password:
Welcome to Ubuntu 12.04.5 LTS (GNU/Linux
3.13.0-36-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

  System information as of Sat Sep 27 01:51:01 EDT 2014

  System load:  1.71               Processes:           267
  Usage of /:   16.8% of 910.59GB  Users logged in:     0
  Memory usage: 76%                IP address for eth0: 129.32.60.13
  Swap usage:   0%

  Graph this data and manage this system at:
    https://landscape.canonical.com/

5 packages can be updated.
1 update is a security update. <---AHHHHHHHHHH I tried to no avail.

New release '14.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Your Hardware Enablement Stack (HWE) is supported until
April 2017.

Last login: Sat Sep 27 01:47:05 2014 from
pool-72-94-192-215.phlapa.fios.verizon.net
tud12685@electrodata:~$ ps fx # looking for bodies
  PID TTY      STAT   TIME COMMAND
40076 ?        S      0:00 sshd: tud12685@pts/0
40129 pts/0    Ss     0:00  \_ -bash
49746 pts/0    R+     0:00      \_ ps -fx
tud12685@electrodata:~$ # the opened shell is dead...Muhaha.
```