Start Recording Your Commands

Run **script**to record the commands that are typed in to the prompt. Run the following

% script

5.1 Combining Commands

Let's combine a bunch of commands using pipes, i.e. '|'.  In your unixstuff directory, type

% cat /etc/passwd

You will see a list of info about users for your system. The **passwd** contains lines like

ubuntu:x:1000:1000::/home/ubuntu:/bin/bash

This line gives information about the user "ubuntu" (that's you for the online IDE). The information is separated by colons, i.e., ':'.  Read about the passwd file here: [https://en.wikipedia.org/wiki/Passwd#Password\_file (Links to an external site.)](https://en.wikipedia.org/wiki/Passwd#Password_file) and note that based on the line above, the user is "ubuntu", the home directory for the user is "/home/ubuntu" and the default shell for this user is "/bin/bash".

Type

% cat /etc/passwd | head -n 1

This will give you info about the first user at the top of the passwd file. We are piping the output of the cat command to head. Don't forget the check the man pages for head to recall what head does.

Type

% cat /etc/passwd | head -n 1 | cut -d ':' -f1

Here we are piping the output of head to cut. Look at the man pages for cut to determine what the -d option and -f options are doing.

Note that the command above will give you user name for the user at the top of passwd.

Now try

% echo "User name: $(cat /etc/passwd | head -n 1 | cut -d ':' -f1)."

This will give you the logged in user name in a sentence. Notice that $( ... ) works like ` ... ` (back ticks). The command in parenthesis runs and the output is subbed in to the string that is then echoed to stdout.

Now recall what is in the /etc/passwd file by typing

% cat /etc/passwd

Exercise 5.1a

By piping "cat /etc/passwd" to grep write a command to output the line containing the user name "ubuntu". (Hint: **cat /etc/passwd | grep "root"** will output the lines in /etc/passwd that contain "root".)

Exercise 5.1b

By extending the command in 5.1a, write a command to output only the default shell program for the user named "ubuntu".  (Hint: use cut with delimiter ':' and field 7.)

5.2 Generating Reports

The "pipe and filter" nature of Linux commands make it easy to generate reports about files. To see this, let's combine a bunch of commands using '|' to generate reports about books in different languages. We'll be using a lot of different commands here, so be sure to check out the man pages for more information about each command.

We'll first figure out how to generate reports about the "jokes.txt" file.

In your unixstuff directory, type

% cat jokes.txt | tr -c [:alnum:] ' ' | tr -s ' ' | tr ' ' '\n'

Figure out what each command in the command above does. The man pages will help here.

awk is a powerful command for formatting output. Piping output to  awk '{ print length }' will print the length (in characters) of each line in the output. For example, the following command will print the number of characters for each joke in jokes.txt.

% cat jokes.txt | awk '{ print length }'

Now type

% cat jokes.txt | tr -c [:alnum:] ' ' | tr -s ' ' | tr ' ' '\n' | awk '{ print length }'

Figure out what the output represents.

Now pipe the output of the above command to sort -n | uniq -c

Figure out what the output represents.

Now pipe the output of the above command to tr -s ' ' | cut -d' ' -f2,3 | tr ' ' ','

Figure out what the output represents.

Putting it all together, we have the following command

cat jokes.txt | tr -c [:alnum:] ' ' | tr -s ' ' | tr ' ' '\n' | awk '{ print length }' | sort -n | uniq -c | tr -s ' ' | cut -d' ' -f2,3 | tr ' ' ','

Using     > data.csv   we can redirect the data from the command to a "csv" file. .csv is a common extension for comma delimited data and stands for Comma Separated Values.

If we graph the data in data.csv with a spreadsheet program to generate a graph, we get the following.

Exercise 5.2

Let's use the command above to analyze three books each in a different language. Use the following command to download a file called books.zip.

wget https://github.com/uwrf-csis/csis248/raw/main/lab5/books.zip

Note that the books.zip is a compressed zip file. So, we need to unzip the file using the following command.

unzip books.zip

This will create a directory called "books" that contains three books as .txt files. Use less to see what is in each txt file.

Use the command created above (the big command with all of the pipes) to make .csv files and then 3 graphs like the graph for the jokes.txt file above. If you are not familiar with Excel (or the open source LibreOffice Calc available on Linux), you can use [https://www.csvplot.com/ (Links to an external site.)](https://www.csvplot.com/). You will have to download the CSV files that you create for each of the books in order to make the graphs.

All three graphs should look similar. This is because shorter words appear more often regardless of the language being used. Information like this is at the heart of how search engines work.

Share your graphs on Discord!

Submit Your Work

Stop the script command by typing

% exit

Then type