Sam Dinkelman

01/20/2019

CIS 360

Lab Report 1

Bash Scripting

Introduction:

The purpose of this assignment was to introduce us to bash scripting. It required us to write a small script that iterates through some log files pulling out useful information and incorporate a function as well.

Process:

I began by carefully reading the instructions for this assignment so I could be sure that I had fulfilled all of the requirements. The instructions state that this script can be run against a specific log file so I chose the syslog log files. It must iterate through the log file looking for useful information, I chose to iterate through multiple log files. The useful information I chose to look for was information that would be highlighted red in these log files. Which always contained the keywords: Failed, Error or Warning in some way. The script must also use two command line arguments, I chose to incorporate these into my script so the user only has to run the script. We also had to include one function so I make my function copy all the log files to a specific home folder we create thus backing them up.

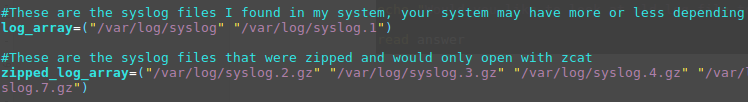
Now I will walk through specific sections of my script.

Now I will walk through specific sections of my script.

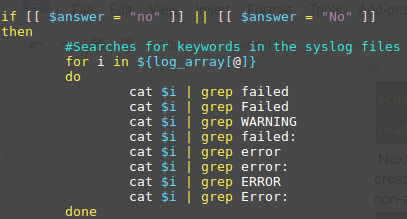
I begin by asking the user if they wish to view older log files with a Yes/No prompt and I store their answer with the read command.



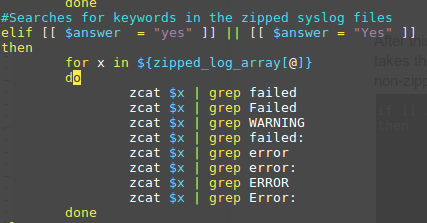
Next I create two arrays with the paths to all of the log files I want to search through. I created two different arrays because I found I could only use the cat command on non-zipped archives and I could only use zcat on zipped archives.



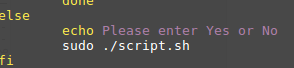
After this I create a series of if statements with for loops within them. This If statement takes the answer the user provided and if it is no then it runs through the array of non-zipped syslog files looking for the keywords and printing those entries to the terminal.



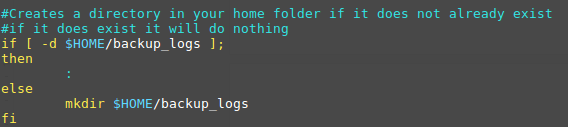
Moving along we have another similar statement that says else if the answer earlier was yes then print the zipped syslog files to the terminal with the zcat command.



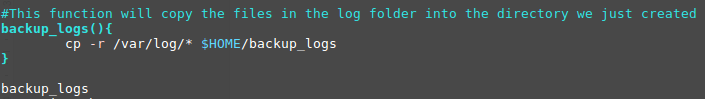
After this I made an else statement so that if the user enters anything except Yes or No it will print this helpful command and run the script again.



Finally I have an if statement that creates a directory called backup\_logs in the /home/user/ directory if it doesn’t already exist. If it does exist then it will do nothing.



After this we create a function called backup\_logs which within will copy the log files stored in /var/log/ into the directory we just created in the /home/user/ directory. Then we call it below to run it and that ends our script so we have our logs all backed up.



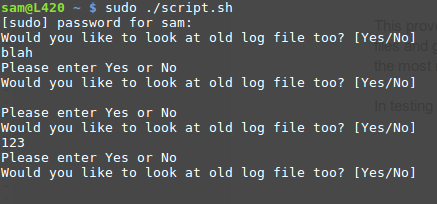
Testing:

In testing this script I looked through my log files so that I could make sure that the script would successfully find some Errors and Failures to print to the console. And it did show the correct red highlighted lines that I observed in the syslog files.

However I originally had some difficulty with cat vs zcat. I have previously only known about the cat command and was trying to use it to access the zipped up syslog logs. This didn’t work and I had to find the zcat command to get it to work. This also meant that I had to have two different for loops one for the un-zipped log files, the more recent log files, and one for the zipped log files.

This proved to be a good thing though because no one wants to look through the old log files and get pages and pages of errors and warnings. So now this script can just show the most recent errors, warnings and failures.

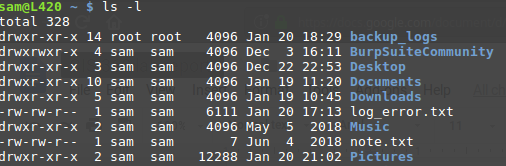
In testing I tried entering values other than yes or no for the first prompt:



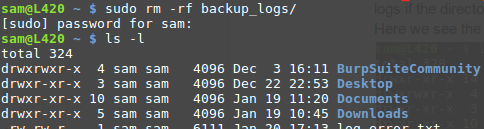
It worked great with it prompting the user to enter Yes or No and then running the script from the start again. The only thing I don’t like about this is it doesn’t have a exit option so you have to run the script at least once.

Next I made sure to test whether or not it actually creates the directory for the backup logs if the directory doesn’t exist.

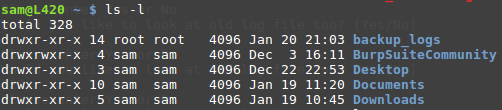
Here we see the backup\_logs directory with all the logs in it:

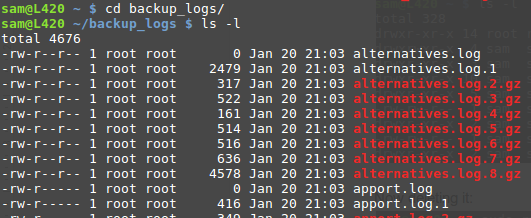


Now deleting it:



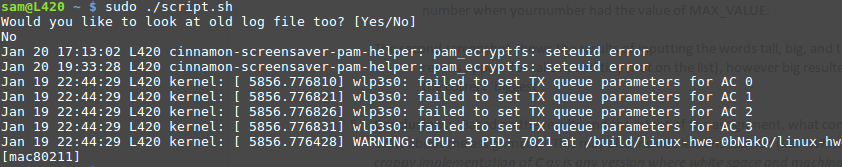
Then running the script again:



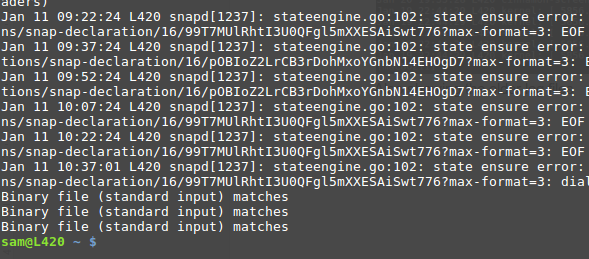


Results:

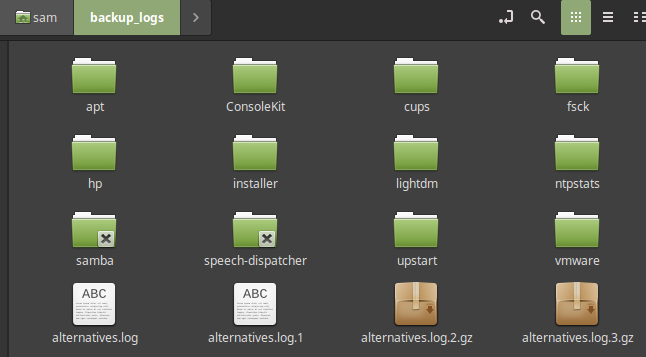
As shown below the results of my script are very robust and match the objectives of the assignment. Here we can see the items that are most recent in the log files with the matching keywords.



Also showing the older log files:



We can also see that the backup function works as well:



Conclusion:

In conclusion this assignment was a very good introduction to scripting and was really fun overall. I found myself constantly adding to my script to be able to include more functionality and features. I really enjoyed the loose constraints of the assignment too which meant that we could look at any log files we wanted and extract any information from them that we wanted. The biggest thing I took away from this is that the skills we developed using C are very easy to translate into bash scripting.

References and Acknowledgements:

I mostly used man pages for commands and then used the sites provided by the instructor such as ryanstutorials.

Chadwick, Ryan. “Bash Scripting Tutorial.” *A Collection of Technology Tutorials*, ryanstutorials.net/bash-scripting-tutorial/.