Unix Family of Operating Systems

Daniel R. Cender

Grand Canyon University: CST-221

January 12, 2020

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *whoami* | *users* | *ls* | | *chmod* | *grep* | | *grep an file-dir -r -h | sort -d* | | |
| *sudo* | *find* | | *traceroute* | | | *vi* | | *nslookup* |  |

**Table of shell commands run for this assignment.**

**This segment contains the results of running basic *bash* commands and scripts.**

1. *whoami*

A screenshot of a cell phone

Description automatically generated

1. *users*

A screenshot of a cell phone

Description automatically generated

1. *ls*

*ls -ls* – Lists files in order, sorted by file size, largest to smallest.

*ls -i* – Lists files with inode identification number, normally hidden.

*ls -a* – Lists all files, including hidden files (ex *.env*).

A screenshot of a cell phone

Description automatically generated

1. *chmod*  - Everyone is set to be able to read this file through the command

A screenshot of a computer

Description automatically generated

1. Check any 3 environment variables – checking my current terminal language setting, my root terminal path, and the path of my NVM (Node Version Manager) directory path.

A screenshot of a computer

Description automatically generated

1. *grep* – Searches for a pattern in a file

A screenshot of a computer

Description automatically generated

1. *grep and sort* – To find a string in a list of files in a directory.

This command, *grep an file-dir -r -h | sort -d,* runs a search on all files (text1.txt, text2.txt, and text3.txt) in the directory *file-dir/* for the pattern “an”, which then pipes to a sort function which arranges the resulting rows in dictionary order with the *-d* flag.

A screenshot of a cell phone

Description automatically generated

1. *sudo* account – The use of the “super user” account is in situations where a user needs to execute protected commands or applications in a Unix-based OS. While logged in as the “super user” account, a user can execute any and all system applications, change file ownership, bind ports numbered under 1024, and modify the root directory (Gite, 2018).

A screenshot of a cell phone

Description automatically generated

1. Five different, useful commands.
   1. *find –* This command allows the user to search for files or directories matching a certain pattern or file type. It can take wild characters (like “\*”) as well.

A screenshot of a cell phone

Description automatically generated

* 1. *traceroute –* This command sends a packet to a designated web address and tracks what servers and and locations the packet travels through to arrive at its destination.

A screenshot of text

Description automatically generated

* 1. *vi* – This command opens an in-console editor for files.

A screenshot of a cell phone

Description automatically generated

*Command to open the vi editor.*

A screenshot of a cell phone

Description automatically generated

*The vi editor open on a file.*

* 1. *gzip* – This utility compresses files and/or directories together into minified packages using Lempel-Ziv encoding algorithms.

A screenshot of a cell phone

Description automatically generated

* 1. *nslookup –* This utility returns the IP addresses associated with a domain name, which is useful for testing whether a machine is communicating properly with DNS servers.

A screenshot of a cell phone

Description automatically generated

**Basic Shell Script**

The script I created possesses very simple commands, all chained together to accomplish a simple set of tasks. In the script is a SAMPLE variable containing lines of text. The script creates a temporary file “temp.txt” and a “/tmp/” directory, then iterates through the lines of the temporary file. It then creates a new file in the “temp/” directory, then writes the current line of text, capitalized, from the temp.txt file to it.

A screenshot of a cell phone

Description automatically generated

The source code to this script may be found on GitHub at: <https://github.com/DanielCender/CST-221/blob/master/Wk1/test.sh>. It will be printed below as well:

#!*/bin/bash*

# *A simple bash script that creates a tmp/ directory with*

# *as many files as lines in the SAMPLE variable.*

# *Each file contains one line of text, capitalized.*

SAMPLE='*lorem ipsum*

*anakin skywalker*

*obi-wan*

*windu*

*United States*

*Friendship*

*Leroy Jenkins*

*Solomon Mines*

*Wine*

*friends forever*

*college days*

*summer sweat*

*workout time*'

DEST=/temp

COUNT=1

# *if file exists*

*if* [ -r temp.txt ]

*then*

echo "*Temp file found, re-creating now!*"

rm temp.txt

*else*

echo "*No temp file found, creating now!*"

*fi*

touch temp.txt

mkdir tmp

echo "$*SAMPLE*" *>* temp.txt

cat temp.txt *|* *while* read line

*do*

FILE\_NAME="*tmp/tmp*${*COUNT*}*.txt*"

echo $FILE\_NAME

touch $FILE\_NAME

echo "$*line*" *|* tr '*[:lower:]*' '*[:upper:]*' *>* $FILE\_NAME

((*COUNT*++))

*done*

# *cleanup temporary holding file*

rm temp.txt

This script uses these explicit commands: *rm, echo, touch, mkdir, cat,* and *tr.*

**Resources:**

For this assignment, I used a few different tools and references. The references are listed below in APA format, although not all of them are explicitly cited in this document, but had influence over the script I wrote.

**Software:** Visual Studio Code, the OSX Terminal, Git, and GitHub.

References

Gite, V. (2017, May 2). HowTo: Check If a Directory Exists In a Shell Script. Retrieved January 12, 2020, from https://www.cyberciti.biz/faq/howto-check-if-a-directory-exists-in-a-bash-shellscript/.

Gite, V. (2018, December 2). Linux Login as Superuser ( root user ). Retrieved January 11, 2020, from <https://www.cyberciti.biz/faq/linux-login-as-super-user/>.

Kamara, I. (n.d.). match command-line arguments to their help text. Retrieved January 10, 2020, from https://explainshell.com/.