Hands-on Projects

These projects should be completed in the order given. The hands-on projects presented in this chapter should take a total of three hours to complete. The requirements for this lab include:

* A computer with Fedora Linux installed according to Hands-on Project 2-1

# Project 7-1

In this hands-on project, you use the shell to redirect the Standard Output and Standard Error to a file and take Standard Input from a file.

1. Boot your Fedora Linux virtual machine. Login to your chosen desktop environment as **user1** using password **LNXrocks!** and open up a terminal window.
2. At the terminal, become **root** by typing **su -** and press enter and provide **LNXrocks!** as the password.
3. At the command prompt, type **du -hs /\*** to view the human readable size of all the top level directories. You will notice some permission denied which is stderr.
4. At the command prompt, type **du -hs /\* > output.txt** to redirect stdout to the output.txt file. You will see that standard error is still displayed to the terminal.
5. View the contents of the **output.txt** file we just created with stdout.
6. At the command prompt, type **du -hs /\* 2>/dev/null output.txt** to redirect stderr to /dev/null.
7. At the command prompt, type **cat file** and press Enter. Standard error was redirected to this file in the previous step.
8. At the command prompt, type **du -hs /\* > output.txt 2>output.err** and press Enter. Standard out and standard error have been redirected to output.txt and output.err.
9. View the contents of **output.txt** and **output.err**.
10. **Provide screenshot(s) of steps 3 through 9.**
11. At the command prompt, type **du -hs /\* > output.txt 2>&1** and press Enter. We have now redirected both standard out and standard error to the same file.
12. View the contents of **output.txt**.
13. At the command prompt, type **du -hs /\* &> output.txt** and press Enter. Notice we have accomplished the same as the previous step.
14. View the contents of **output.txt**.
15. At the command prompt, type **date >> output.txt** and press Enter.
16. View the contents of **output.txt**.
17. At the command prompt, type **tr o O /etc/hosts** and press Enter. Notice the error message. We need to redirect the contents of /etc/hosts to the tr command.
18. At the command prompt, type **tr o O </etc/hosts** to redirect the hosts file as stdin.
19. **Provide screenshot(s) of steps 11 through 18.**

# Project 7-2

In this hands-on project, you redirect Standard Output and Standard Input using pipe metacharacters.

1. Boot your Fedora Linux virtual machine. Login to your chosen desktop environment as **user1** using password **LNXrocks!** and open up a terminal window.
2. At the terminal, become **root** by typing **su -** and press enter and provide **LNXrocks!** as the password.
3. At the command prompt, type **wget -O apache\_access.log https://help.sumologic.com/@api/deki/files/2837/apache\_access.csv?revision=1** and press Enter to download the apache\_access.log file so we can manipulate it to see who has accessed our site and get an access count.
4. Use either the head or tail command to view some of the logs contained within the **apache\_access.log** file.
5. At the command prompt, type **awk '{print $1}' apache\_access.log** and review the output. As you can see, $1 outputs the IP that accessed our website. Also notice the double quote in the output.
6. At the command prompt, type **awk '{print $1}' apache\_access.log | sed 's/”//'** to remove the double quote.
7. At the command prompt, type **awk '{print $1}' apache\_access.log | sed 's/”//' | sort** to sort the output for the uniq command. If you were to read the uniq man page, it states that it filters adjacent matching lines.
8. At the command prompt, type **awk '{print $1}' apache\_access.log | sed 's/”//' | sort | uniq -c** to get a unique count of the IP addresses that have accessed our website.
9. At the command prompt, type **awk '{print $1}' apache\_access.log | sed 's/”//' | sort | uniq -c | sort -nr** to sort our output numerically and in reverse order.
10. At the command prompt, type **awk '{print $1}' apache\_access.log | sed 's/”//' | sort | uniq -c | sort -nr | tee unique\_ip.txt** to write stdout to the terminal and to the unique\_ip.txt file.
11. View the contents of the unique\_ip.txt file using any command you like that will display all the contents.
12. **Provide screenshot(s) of steps 3 through 11.**

# Project 7-3

In this hands-on project, you create and use an alias, as well as view and change existing shell variables. In addition to this, you export user-defined variables and load variables automatically upon shell startup.

1. Boot your Fedora Linux virtual machine. Login to your chosen desktop environment as **user1** using password **LNXrocks!** and open a terminal window.
2. At the terminal, become **root** by typing **su -** and press enter and provide **LNXrocks!** as the password.
3. At the command prompt, type **set | less** and press Enter to view the BASH shell environment variables currently loaded into memory.
4. At the command prompt, type **env | less** and press Enter to view the exported BASH shell environment variables currently loaded into memory.
5. At the command prompt echo the contents of the PS1 variable.
6. At the command prompt, type **PS1='** **[\u@\h \w]\$ '** and press Enter.
7. Echo the contents of the PS1 variable to verify the new value of the PS1 variable.
8. Change directory to **/etc/sysconfig/network-scripts** to see the absolute path in your PS1 prompt.
9. Modify your hidden environment file **.bash\_profile** and add the following to the end:

**export PS1='\[\e[1;32m\][\[\e[1;31m\]\u\[\e[1;32m\]@\[\e[1;34m\]\h \[\e[1;37m\]\W\[\e[1;32m\]]\[\e[1;33m\]\$\[\e[m\] '**

1. Notice that when you add the new PS1 to your .bash\_profile, it does not change your current prompt. We have two ways we can load this new variable. Type **source .bash\_profile** or **. .bash\_profile** (. is the same as using source). You will now see the new PS1 loaded.
2. At the command prompt, type **CLASSVAR = "ITMO 456"** and press Enter. Recall that we cannot have any spaces between the variable name and its contents. Now type **CLASSVAR="ITMO 456"** to create a variable called CLASSVAR. Verify its creation by using echo at the command prompt (don’t forget the $).
3. At the command prompt, type **set | grep CLASSVAR** and press Enter.
4. At the command prompt, type **env | grep CLASSVAR** and press Enter.
5. At the command prompt, type **export CLASSVAR** and press Enter. Next, type **env | grep CLASSVAR** at the command prompt and press Enter.
6. Open a new terminal/tab and at the command prompt, type **echo $CLASSVAR** and press Enter to view the contents of the CLASSVAR variable. Note that nothing returns as exporting the variable only effects sub-shells of the current shell, not new ones. The next step will show us how to have variables persist to new shells.
7. At the command prompt, type **vi .bash\_profile** and press Enter. At the bottom of the file, add the following line. When finished, save and quit the vi editor.

**export CLASSVAR="ITMO 456"**

1. Source the .bash\_profile like we did in step 10.
2. At the command prompt, type **echo $CLASSVAR** and press Enter to list the contents of the MYVAR variable.
3. At the command prompt, type **alias** and press Enter.
4. At the command prompt, type **alias** **asample="cd /etc; cat hosts; cd ~; ls -F"** and press Enter.
5. At the command prompt, type **asample** and press Enter.
6. **Provide screenshot(s) of steps 3 through 21.**

# Project 7-4

In this hands-on project, you create a basic shell script and execute it on the system.

1. Boot your Fedora Linux virtual machine. Login to your chosen desktop environment as **user1** using password **LNXrocks!** and open up a terminal window.
2. At the terminal, become **root** by typing **su -** and press enter and provide **LNXrocks!** as the password.
3. At the command prompt, type **vi myscript** and press Enter to open a new file for editing called myscript in your home directory.
4. Enter the following text into the myscript file. When finished, save and quit the vi editor.

**#!/bin/bash**

**echo -e "This is a sample shell script. \t It displays mounted**

**filesystems \a"**

**mount**

1. At the command prompt, type **ll myscript** and press Enter.
2. Next, type **bash myscript** at the command prompt and press Enter.
3. Next, type **./myscript** at the command prompt and press Enter. Note the error message you received.
4. At the command prompt, type **chmod u+x myscript** and press Enter. Next, type **./myscript** at the command prompt and press Enter. The script now executes.
5. **Provide screenshot(s) of steps 3 through 8.**

# Project 7-5

In this hands-on project, you create a shell script that uses decision and loop constructs to analyze user input.

1. Boot your Fedora Linux virtual machine. Login to your chosen desktop environment as **user1** using password **LNXrocks!** and open up a terminal window.
2. At the terminal, become **root** by typing **su -** and press enter and provide **LNXrocks!** as the password.
3. Enter the following text into the **myscript2** file. When finished, save and quit the vi editor.

**#!/bin/bash**

**echo -e "This program adds entries to a family database file.\n"**

**echo -e "Please enter the name of the family member -->\c"**

**read NAME**

**echo -e "Please enter the family member’s relation to you (i.e. mother) -->\c"**

**read RELATION**

**echo -e "Please enter the family member’s telephone number -->\c"**

**read PHONE**

**echo -e "$NAME\t$RELATION\t$PHONE" >> database**

1. At the command prompt, type **chmod u+x myscript2** and press Enter. Next, type **./myscript2** at the command prompt and press Enter. Answer the questions with information regarding one of your family members.
2. At the command prompt, type **cat database** and press Enter.
3. Perform Step 4 several times to populate the database file with entries.
4. **Provide screenshot(s) of steps 3 through 6.**
5. At the command prompt, type **vi myscript2** and press Enter. Edit the text inside the myscript2 shell script such that it reads:

**#!/bin/bash**

**echo -e "Would you like to add an entry to the family database file?\n"**

**read ANSWER1**

**if [ $ANSWER1 = "y" -o $ANSWER1 = "Y" ]**

**then**

**echo -e "Please enter the name of the family member -->\c"**

**read NAME**

**echo -e "Please enter the family member’s relation to you (i.e. mother)-->\c"**

**read RELATION**

**echo -e "Please enter the family member’s telephone number -->\c"**

**read PHONE**

**echo -e "$NAME\t$RELATION\t$PHONE" >> database**

**fi**

**echo -e "Would you like to search an entry in the family database file?\n"**

**read ANSWER2**

**if [ $ANSWER2 = "y" -o $ANSWER2 = "Y" ]**

**then**

**echo -e "What word would you like to look for? -->\c"**

**read WORD**

**grep "$WORD" database**

**fi**

1. At the command prompt, type **./myscript2** and press Enter. When prompted to enter an entry into the database, choose y and press Enter. Answer the questions with information regarding one of your family members. Next, when prompted to search the database, answer y and press Enter. Search for the name that you just entered a few seconds ago.
2. At the command prompt, type **./myscript2** and press Enter. When prompted to enter an entry into the database, choose **n** and press Enter. Next, when prompted to search the database, answer **y** and press Enter. Search for a name that you entered earlier in Step 6.
3. **Provide screenshot(s) of steps 8 through 10.**
4. At the command prompt, type **vi myscript2** and press Enter. Edit the text inside the myscript2 shell script such that it reads:

**#!/bin/bash**

**echo -e "What would you like to do?**

**Add an entry (a)**

**Search an entry (s)**

**Enter your choice (a/s)-->\c"**

**read ANSWER**

**case $ANSWER in**

**a|A ) echo -e "Please enter the name of the family member -->\c"**

**read NAME**

**echo -e "Please enter the family member’s relation to you (i.e. mother)-->\c"**

**read RELATION**

**echo -e "Please enter the family member’s telephone number -->\c"**

**read PHONE**

**echo -e "$NAME\t$RELATION\t$PHONE" >> database**

**;;**

**s|S ) echo -e "What word would you like to look for? -->\c"**

**read WORD**

**grep "$WORD" database**

**;;**

**\*) echo "You must enter either the letter a or s."**

**;;**

**esac**

1. At the command prompt, type **./myscript2** and press Enter. Choose y and press Enter.
2. At the command prompt, type **./myscript2** and press Enter. Choose a and press Enter. Enter information about another family member.
3. At the command prompt, type **./myscript2** and press Enter. Choose s and press Enter. Search for the family member entered in Step 12.
4. **Provide screenshot(s) of steps 12 through 15.**
5. At the command prompt, type vi myscript2 and press Enter. Edit the text inside the myscript2 shell script such that it reads:

**#!/bin/bash**

**while true**

**do**

**clear**

**echo -e "What would you like to do?**

**Add an entry (a)**

**Search an entry (s)**

**Quit (q)**

**Enter your choice (a/s/q)-->\c"**

**read ANSWER**

**case $ANSWER in**

**a|A ) echo -e "Please enter the name of the family member -->\c"**

**read NAME**

**echo -e "Please enter the family member’s relation to you (i.e. mother)-->\c"**

**read RELATION**

**echo -e "Please enter the family member’s telephone number -->\c"**

**read PHONE**

**echo -e "$NAME\t$RELATION\t$PHONE" >> database**

**;;**

**s|S ) echo -e "What word would you like to look for? -->\c"**

**read WORD**

**grep "$WORD" database**

**sleep 4**

**;;**

**q|Q ) exit**

**;;**

**\*) echo -e "You must enter either the letter a or s."**

**sleep 4**

**;;**

**esac**

**done**

1. At the command prompt, type **./myscript2** and press Enter. Choose **a** and press Enter. Enter information about another family member. Choose s and press Enter. Search for the family member that you just entered. Choose q to quit the shell script.
2. At the command prompt, type **vi myscript3** and press Enter to edit a new file called myscript3 in your home directory.
3. **Provide screenshot(s) of steps 17 through 19.**
4. Enter the following text into the myscript3 file. When finished, save and quit the vi editor

**#!/bin/bash**

**echo -e "This program copies a file to the /stuff directory.\n"**

**echo -e "Which file would you like to copy? -->\c"**

**read FILENAME**

**mkdir /stuff || echo "The /stuff directory could not be created."**

**cp -f $FILENAME /stuff && echo "$FILENAME was successfully copied to stuff/"**

1. At the command prompt, type **chmod u+x myscript3** and press Enter. Next, type **./myscript3** at the command prompt and press Enter. When prompted for a filename, type **/etc/hosts** and press Enter. The directory should be created that contains /etc/hosts.
2. Type **./myscript3** at the command prompt and press Enter. When prompted for a filename, type **/etc/inittab** and press Enter.
3. At the command prompt, type **vi myscript4** and press Enter to edit a new file called myscript4 in your home directory.
4. **Provide screenshot(s) of steps 21 through 24.**
5. Enter the following text into the myscript4 file. When finished, save and quit the vi editor.

**#!/bin/bash**

**echo "These are the scripts that you have created previously:"**

**ls -l myscript myscript2 myscript3**

**sleep 2**

**echo "This script will now change the permissions on each script such that the root user has exclusive rights only."**

**sleep 3**

**for FILE in myscript myscript2 myscript3**

**do chmod 700 $FILE**

**done**

**echo "The new permissions are listed below:"**

**ls -l myscript myscript2 myscript3**

1. At the command prompt, type **chmod u+x myscript4** and press Enter. Next, type ./myscript4 at the command prompt and press Enter. Were the permissions changed to rwx- - - - - - for myscript, myscript2, and myscript3?
2. **Provide screenshot(s) of steps 26 through 27.**