# Activity 6

Kaya Nelson

Grand Canyon University: CST-221

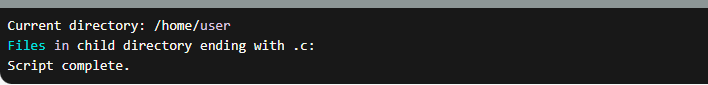
06/10/2024

# Bash Shell Scripting – Simple File System Navigation –

A screenshot of a computer program

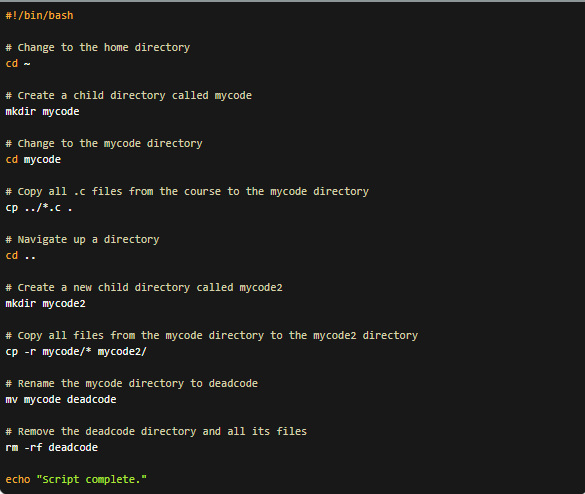
Description automatically generated

The output should look similar to:

  
The key steps in this script are:

1. Echo the current directory using pwd
2. Change to the home directory using cd ~
3. Create a child directory within the home directory using mkdir -p ~/child\_dir and navigate to it with cd ~/child\_dir
4. List all files in the child directory that end with .c using ls \*.c
5. Navigate back to the home directory using cd ~
6. Echo a completion message

# Bash Shell Scripting – Simple File Manipulation –



The key steps in this script are:

1. Change to the home directory using cd ~
2. Create a child directory called mycode using mkdir mycode
3. Change to the mycode directory using cd mycode
4. Copy all .c files from the parent directory to the mycode directory using cp ../\*.c .
5. Navigate up a directory using cd ..
6. Create a new child directory called mycode2 using mkdir mycode2
7. Copy all files from the mycode directory to the mycode2 directory using cp -r mycode/\* mycode2/
8. Rename the mycode directory to deadcode using mv mycode deadcode
9. Remove the deadcode directory and all its files using rm -rf deadcode
10. Echo a completion message

# Working with the Terminal – Simple Text File Viewing –

a. Running ls, ls -a, and ls -l on the home directory:

A screenshot of a computer

Description automatically generated

b. Navigating to the directory with C source code files:



c. Exploring the less command with options:

A black screen with white text

Description automatically generated

d. Exploring the more command with options:

A black rectangular object with a gray border

Description automatically generated

e. Running the file command:

A black screen with white text

Description automatically generated

f. Colorizing the terminal window:



g. Running ls, ls -a, and ls -l on the home directory again:

A screenshot of a computer

Description automatically generated

# Bash Shell Scripting – Misc. Commands –

a. Creating the ~/test3.sh script:

A screenshot of a computer

Description automatically generated

b. Take a screenshot of the console output:

A black background with white letters

Description automatically generated

c. Creating the ~/test4.sh script:

A screen shot of a computer

Description automatically generated

d. Take a screenshot of the console output:

A screen shot of a computer code

Description automatically generated

e. Running the scripts with chmod 777:

A black screen with white text

Description automatically generated

f. Running which gcc and man gcc:

A black screen with white text

Description automatically generated

# Bash Shell Scripting – Permissions –

a. Creating the ~./test5.sh script:



b. Saving the script.

c. Running the ls -l command:



i. The owner of this file is the user "kaya".

ii. No, I cannot execute this script because the execute permission is not set for the owner.

d. Changing the file permissions with chmod:



e. Running the ls -l command again:



i. The 'x' permission bit has been set for the owner, allowing me to execute the script.

f. Screenshot of the console output:

A black screen with white text

Description automatically generated

# Working with the Terminal – Redirection –

a. Running the ls -l > myfiles.txt command:



This command will create a file named myfiles.txt and redirect the output of the ls -l command into it.

b. Inspecting the myfiles.txt file with a text editor:

The contents of the myfiles.txt file will contain the long-format listing of the files in the current directory.

c. Sorting text in the names.txt file:

i. Creating the names.txt file in a text editor:

A black background with white text

Description automatically generated

ii. No further action needed here.

iii. Running the sort < names.txt command:

A black rectangle with white dots

Description automatically generated

vi. Screenshot of the console output:

A black rectangular object with white text

Description automatically generated

# Bash Shell Scripting – If, Loop, and Reading Files –

a. Script to determine if a number is less than, equal to, or greater than 50:

A screenshot of a computer program

Description automatically generated

b. Script to display numbers 1 through 10 using a for loop:

A screenshot of a computer

Description automatically generated

c. Script to print all files from the ls list command using a for loop:

A screenshot of a computer

Description automatically generated

d. Script to display numbers 1 through 10 using a while loop:

A screenshot of a computer program

Description automatically generated

e. Script to print sorted names from a text file and search for a desired name using an in condition:

A computer screen shot of a black background

Description automatically generated

Screenshot of the console output:

A black rectangle with white text

Description automatically generated

# Working with the Terminal – Head and Tail Files –

a. Navigating to the Linux logs directory at /var/log:

b. Using the head and tail commands to view the syslog file:

A screenshot of a computer

Description automatically generated

# Working with the Disk Utilities –

a. Running the sudo fdisk -l command:

A computer screen shot of a black screen

Description automatically generated

i. The types of devices in the Linux system are:

* /dev/sda1: EFI System
* /dev/sda2: Linux filesystem
* /dev/sda3: Linux filesystem

ii. The sudo command is used to run the fdisk command with elevated privileges (root access) because the fdisk command requires administrative access to list the details of the storage devices connected to the system.

b. Running the df command:

A screenshot of a computer program

Description automatically generated

i. The various mounts in the Linux system are:

* /dev: Device files
* /run: Runtime directory
* /: Root file system
* /dev/shm: Shared memory
* /run/lock: Lock file directory
* /sys/fs/cgroup: Cgroup file system
* /boot/efi: EFI System Partition
* /run/user/1000: User runtime directory

c. Navigating to the directory where C code resides and running the du command:

A black rectangle with white dots

Description automatically generated