Course Project

DeVry University
College of Engineering and Information Sciences

Operating Systems
Module 3 Linux Shell Scripts

Introduction

A shell script is a file that contains Linux commands and special constructs. It is widely used to perform and automate administrative tasks, by combining a lengthy sequence of commands into one script. A script is interpreted and executed by the Linux shell. Therefore, any command that can be entered on the Linux command line can be part of a shell script. A shell script is often created as an ASCII text file by using a text editor program.

In this module of the course project, you will create and execute a shell script by experimenting with the standard input, user-defined variables, redirection, file permissions, and environment variables.

Procedure

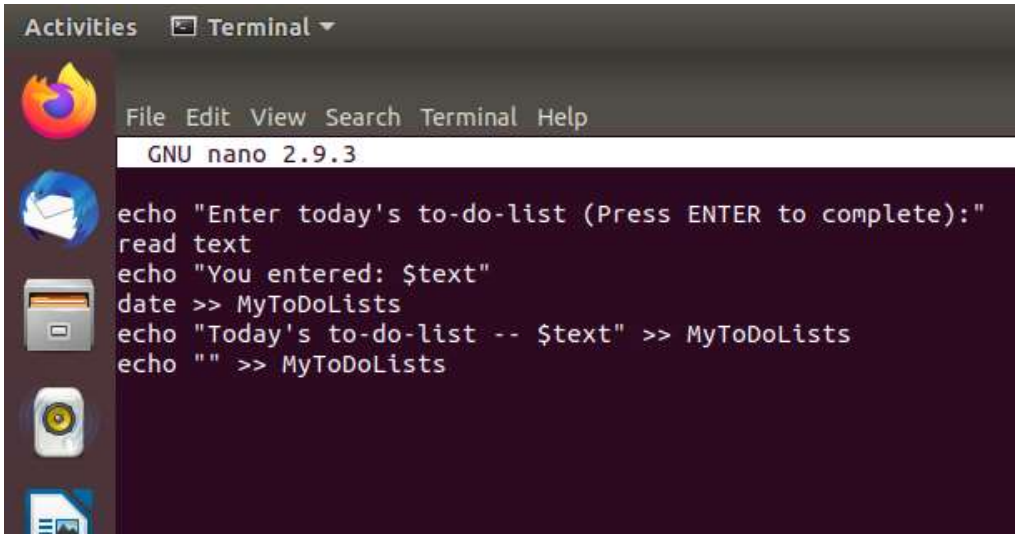
A. Create a shell script.
B. Change script file permissions.
C. Set the PATH variable.
D. Make the PATH variable permanent.

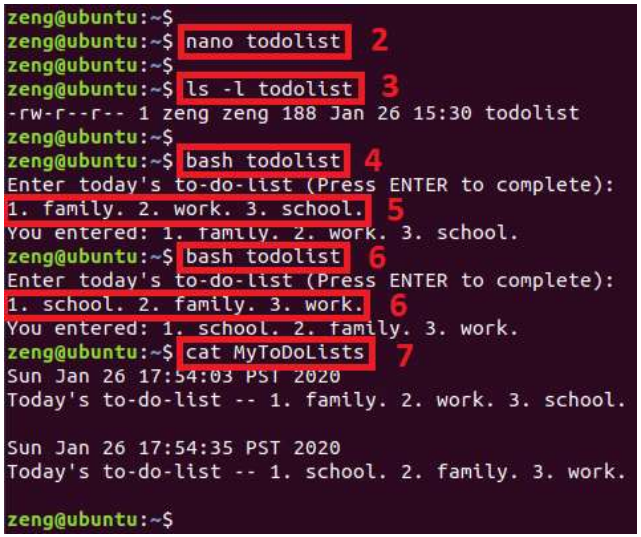
Deliverable

Course Project Module 3 PowerPoint Template

A. Create a shell script

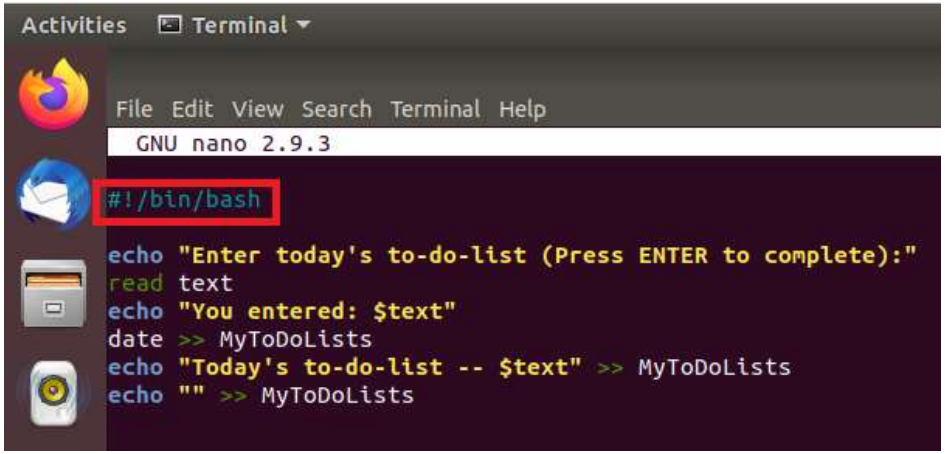
1. Start only the *Ubuntu* VM in Hyper-V Manager. In the Terminal window, enter **cd ~** to return to the /home/student directory. Verify the directory.
2. Enter **nano todolist** at the command prompt. Type in the script as shown in the screen capture below. Use the left, right, up, and down arrow keys to move the cursor in the nano editor. Double check the script before pressing **^x** (ctrl + x), answering **y** to question "Save modified buffer?", and pressing **Enter** to keep the same script name. Note that Linux treats upper-case and lower-case letters differently! Enter **cat todolist** to verify the script content.
3. Enter **ls –l todolist** to check the file permissions. Notice that no execute permission is set for the script file.
4. Execute the script by entering **bash todolist**. If there are error messages displayed, go to Step 2 and double check every character including spaces!
5. Enter **1. family. 2. work. 3. school.** at the prompt and press the **Enter** key.
6. Execute the script again and enter **1. school. 2. family. 3. work.** at the prompt and press the **Enter** key.
7. Enter **cat MyToDoLists** to view the two entries you entered before.
8. Open the Course Project Module 3 PowerPoint Template and answer three questions there.

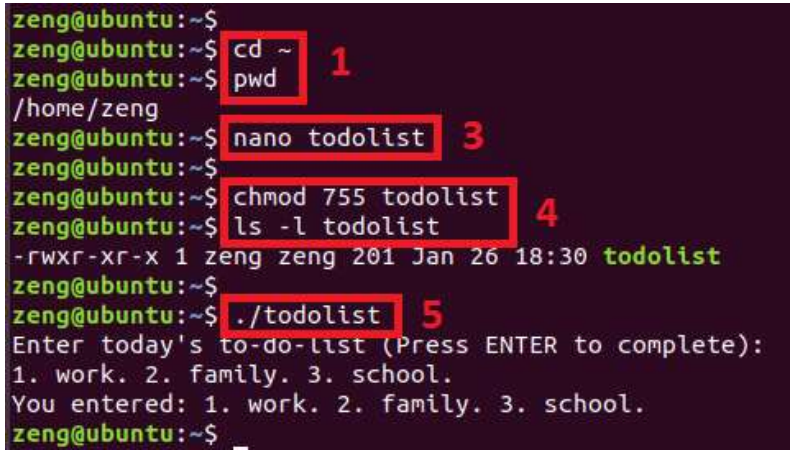




B. Change script file permissions

1. In the Terminal window, enter **cd ~** to return to the /home/student directory. Use the **pwd** command to verify the current directory.
2. The **bash todolist** command calls the bash executable or binary directly to interpret and execute the script. Another way to execute it is to make the shell script executable and declare the hashpling or shebang line (i.e., #!/bin/bash) in the script. This line specifies the pathname to the shell binary that interprets the script.
3. Enter **nano todolist** to edit the script. Add **#!/bin/bash** as the first line of the script. In the nano editor, press **^x** (ctrl + x), answer **y** to question "Save modified buffer?", and press **Enter** to keep the same script name.
4. Make the script executable by entering the **chmod 755 todolist** command. Type **ls –l todolist** to verify file permissions.
5. Enter **./todolist** to execute the script by calling its name. Note that the . (dot) character represents the current directory, and ./todolist is the relative path name of the script. Enter **1. work. 2. family. 3. school.** at the prompt and press the **Enter** key.
6. Take a screenshot of the output in Step 5. Paste it in the Course Project Module 3 PowerPoint Template.





C. Set the PATH variable

1. In the Terminal window, enter **cd ~** to return to the /home/student directory. Use the **pwd** command to verify the current directory.

2. PATH is an environment variable that contains a list of directories. When a command or script is called without an absolute or relative pathname, the shell searches the directories in the list for the executable file. If the file is not found, a "command not found" error message is displayed.

3. Enter **todolist** at the command prompt to observe the error message.

4. Enter **echo $PATH** to display the current directories in the list.

5. Add your home directory where the todolist script is to the PATH variable by using the **PATH=$PATH:/home/student** command. Verify the change by entering the **echo $PATH** command. Notice that your input is slightly different from what's shown in the screenshot.

6. Execute the script by entering **todolist** (instead of ./todolist). Enter **1. school. 2. work. 3. family.** at the prompt and press the **Enter** key.

7. Take a screenshot of the output in Step 6. Paste it in the Course Project Module 3 PowerPoint Template.

D. Make the PATH variable permanent

1. Right click on the Ubuntu Desktop to open a NEW Terminal window. Enter **cd ~** to return to the /home/student directory. Use the **pwd** command to display the current directory.
2. Enter **todolist** at the command prompt to run the script. It returns the "command not found" error message. The previous change to the PATH variable does not carry over from one shell window to the next!
3. One way of setting the PATH variable permanently is to modify the hidden .bashrc file which runs every time you open a new terminal window.
4. Enter **ls –a .bash*** to list the .bashrc file. Make a copy of it by entering the **cp .bashrc .bashrc.old** command, in case you need to reverse changes.
5. Enter **nano .bashrc** to open the file. Add a new line **export PATH=$PATH:/home/student** to the end of the file (your input is slightly different from what's shown in the screenshot below). Press **^x** (ctrl + x), answer **y** to question "Save modified buffer?", and press **Enter** to keep the same file name.
6. To active the change, use the **source .bashrc** command.
7. Open another terminal window. Run the **todolist** script. Take a screenshot of both Terminal windows (as shown in the bottom right of this page). Paste it in the Course Project Module 3 PowerPoint Template. **Shut down the Ubuntu VM. Exit the Lab environment following the proper procedure.**