COMP304 - Operating Systems

Project #1 Shellfyre

March 31, 2022

Deniz Erdogan - 69572
Oya Suran - 69337

Contents

1 cdh

For this part, we implemented a mechanism that writes the path of the changed directories into a file whenever cd is called. Following this, cdh takes the last 10 lines of this file, displays them on the terminal and asks the user to choose one of them. We have completed all the parts but getting the user input. For some reason that we were not able to understand, regular I/O methods did not seem to work.

To tackle this issue, we implemented it in such a way that it always chooses the second last directory for demonstration purposes. A sample run can be seen from Figure 1.

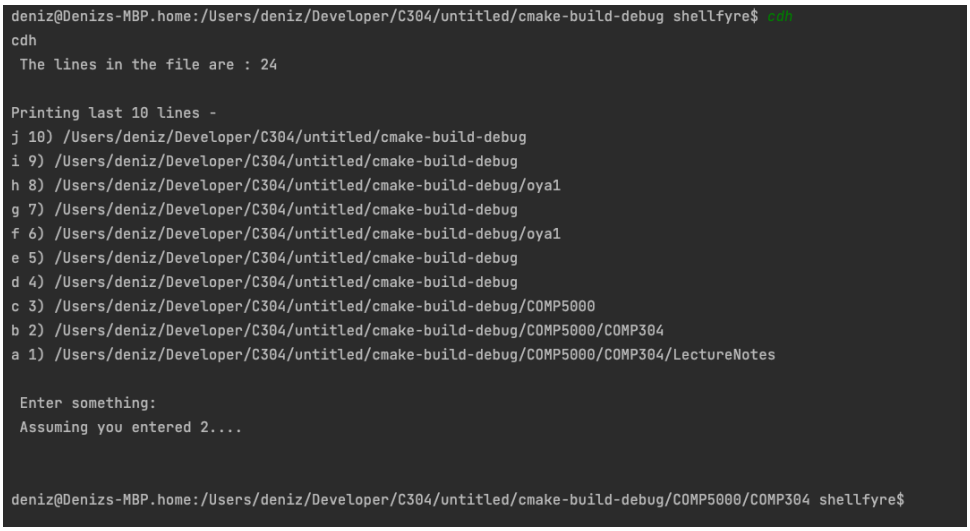


Figure 1: *cdh* demo

2 joker

For this command, we have managed to create correct cronjobs that curl the joke from the address and display it as a notification every 15 minutes. The corresponding cronjob entry is as follows:

*/15 * * * * XDG_RUNTIME_DIR=/run/user/$(id -u) notify-send "$(curl https://icanhazdadjoke.com/)"

After this, we add this among our cronjobs with system calls. A sample notification is as follows:

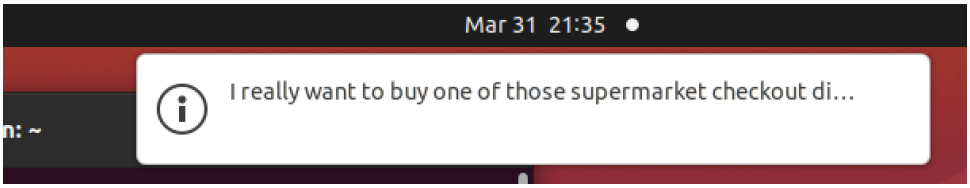


Figure 2: Sample Notification

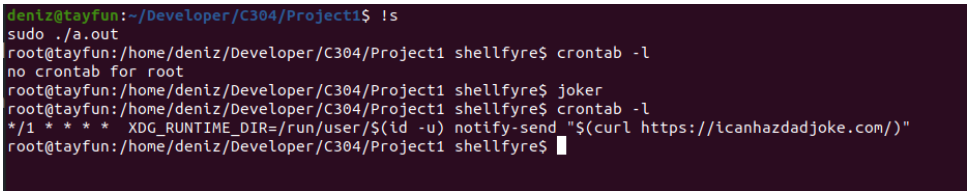


Figure 3: Added Cronjob

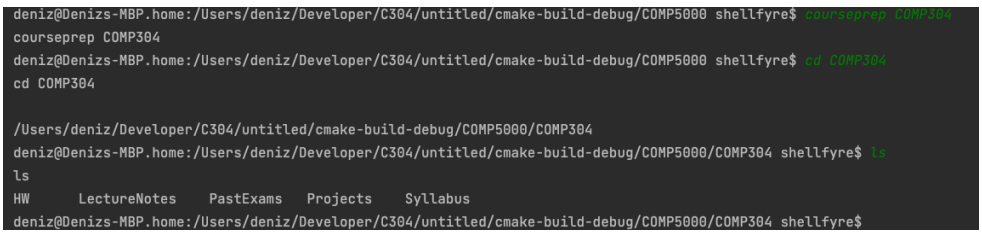
3 Custom Commands

3.1 courseprep

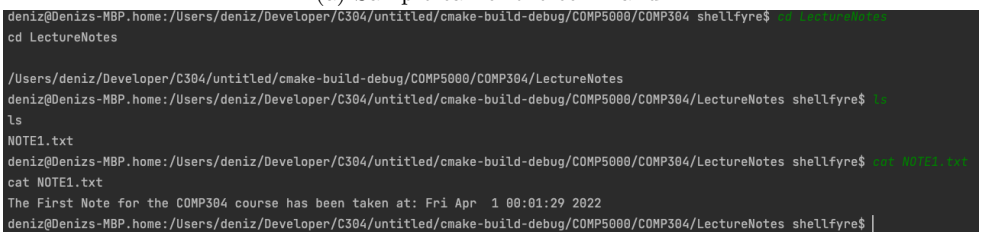
This command is one that can automate a highly repetitive task that every student faces in the beginnings of each semester. In the first days of a semester, most people create folders in an organized manner with very similar subfolders. For instance, a folder with the name of a course with subfolders as follows:

- Homeworks
- Past Exams
- Syllabus
- Lecture Notes
- Projects

Our command *courseprep* takes one argument as the name of the course, create a main folder with that name the necessary subfolders inside. Also, it add a new text file in the lecture notes folder with the name of the course and the date of the creation inside as can be seen from Figure 4.



(a) Sample call of the command.



(b) Created structure and note.

Figure 4: *courseprep* demo

3.2 didemunatsays

This command takes an arbitrary number of arguments of words. Then, it concatenates those and prints them along with an ASCII character portrait of our professor, Didem Unat. A

sample run can be seen from the Figure below. The ASCII portrait is obtained from this website: https://www.text-image.com/convert/ascii.html

Figure 5: *didemunatsays* demo

4 take

To implement take correctly we needed to handle recursive case where user gives new directories inside new directories. To do that, we needed to get the name of these directories hence the parsing the argument. Then with mkdir command we try to create the first directory if it can be created, we move on to other tokenized parts. We do this until we get out of the while loop in that case there is no more directories to create. After this process we change into newest created directory. A sample run can be seen below:

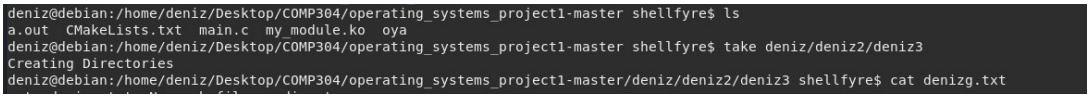


Figure 6: Sample run.

5 filesearch

For filesearch we have several different cases we check whether user gives -r or -o flags. For recursive procedure we added a help function to ease our implementation. We first go into this function with our current directory indicated with '.' . We read the directory we are in and try to see if there is another directory we can go into if there is we recurse in that directory as well. While searching we also see if the files have the name we want to find. Since there might be some privilege differences we check if we can open a directory for this purpose we also print their D values.

We also used 'xdg-open' command to open any file in its default viewer and we tested in Ubuntu. When the user enters the command *filesearch ¡filename¿ -r -o* , it searches recursively and opens

the matching files accordingly.

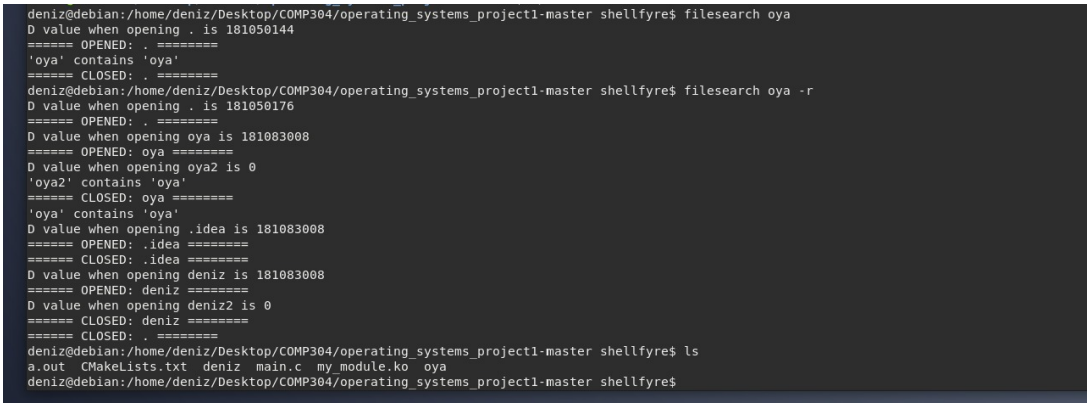


Figure 7: Sample Run

6 pstraverse

We implemented the first parts:

- If user asks for pstraverse for the first time we load the module with system call and we change our loaded module flag to true.
- If user asks for pstraverse again we do not load the module again.
- When user types exit if the module is loaded, we remove it.

We tried to take the argument of dfs or bfs from user, but we could not implement correctly how to pass this parameter to Kernel Module.

A sample run of this command can be seen from the figure below:

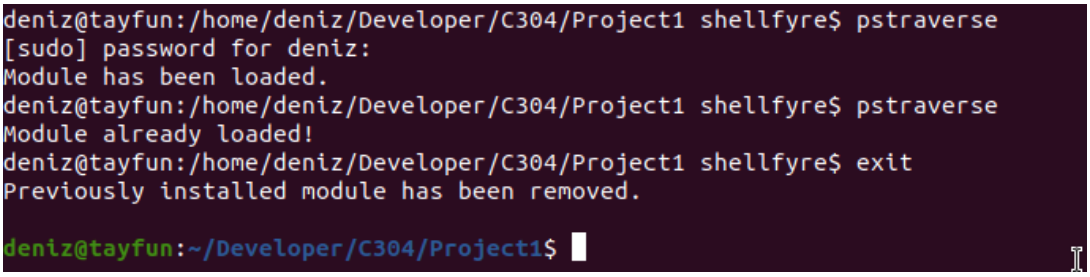


Figure 8: Sample run