

Simple Shell

OPERATING SYSTEMS – LAB1

DR. HICHAM ELMONGUI

Nevine Said

COMPUTER AND COMMUNICATION TERM 7 | ID: 4500

Code organization

Parsing

The code starts with parsing the string inputted by the user in the shell. This is done using a famous C library called readline. It has some linker error when run on linux though. Therefore, I had to add it to the build options in linker tab.

If cd command

After parsing, the program checks if this is a cd command. If it is a cd command, it calls the responsible function cd() which gets the current directory.

If exit command

This condition checks if it exit. If so, it will terminate the shell using the exit() system call or return 0;.

Fork()

After that, the fork() command is used. Now we have a child process.

In case of a command with no arguments or with arguments, we wait for it to execute. Commands that end with & will be nested.

Execvp()

Executes commands by passing the array of instructions.

Functions:

Parsing function

```
char **get_instr(char *in);
```

The function allocates 10 blocks -as in our project we won't need more- then it separates the string by the space. It puts those tokens (separated strings) in an array.

cd command

```
int cd(char *dir);
```

this function is just responsible for delivering the current directory.

Handler

```
void handler(int s);
```

It is used to send the SIGCHLD to the parent process when the child terminates.

Main

Contains while(1) which is the main loop. The code is executed here. Fork() and execvp() are used in this function.

Screenshots

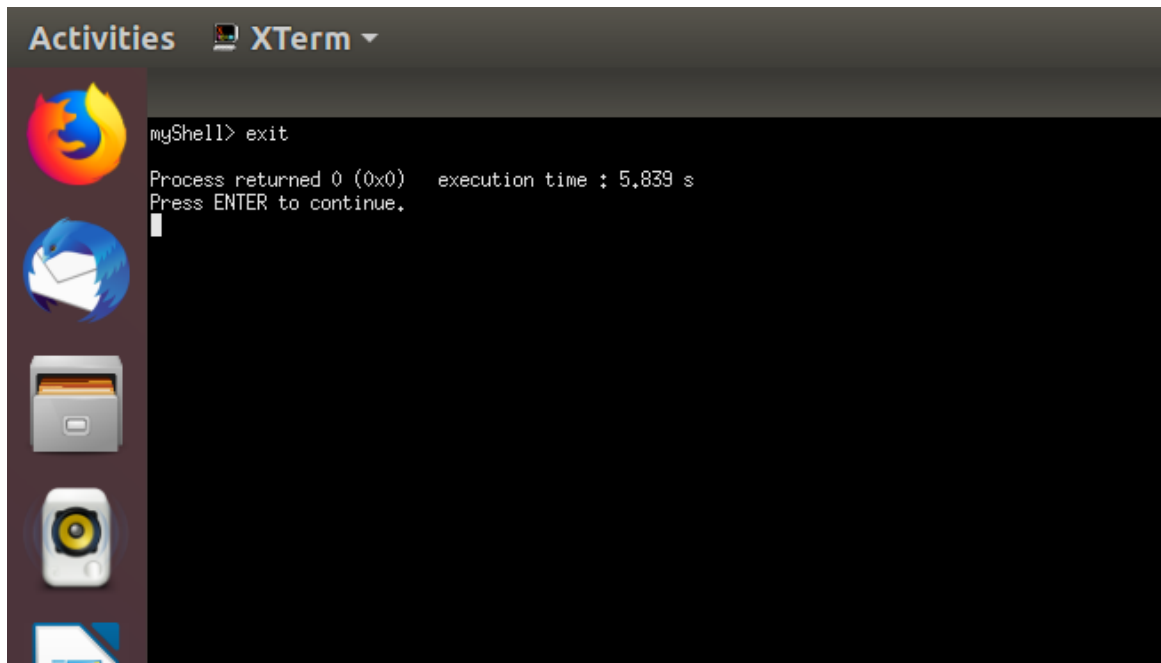


Figure 1: Exit command

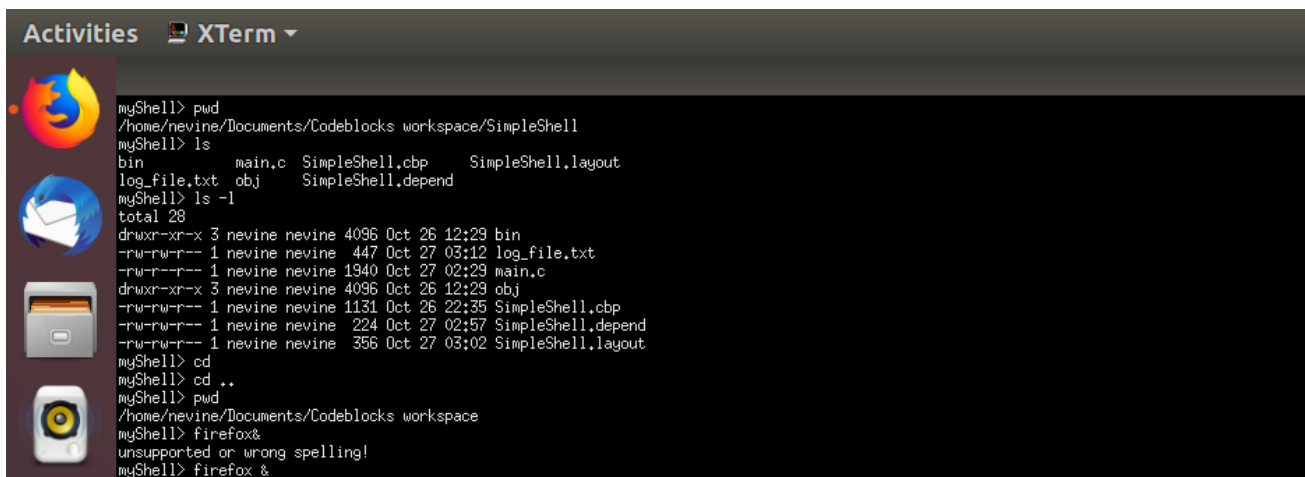


Figure 2: Some commands such as pwd, ls and ls-l

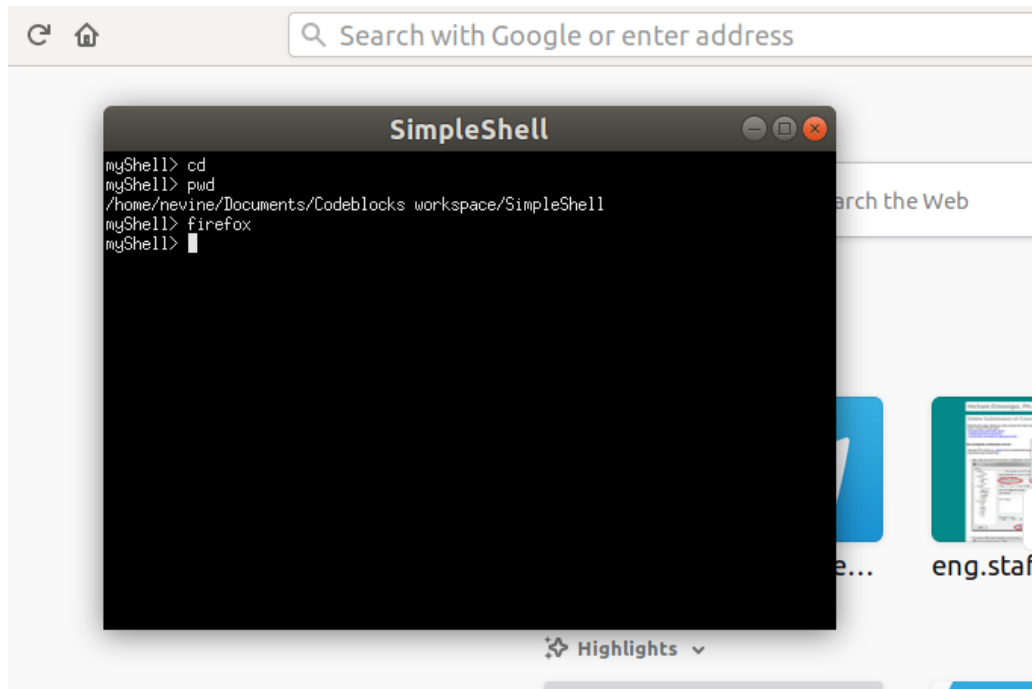


Figure 3: Running firefox

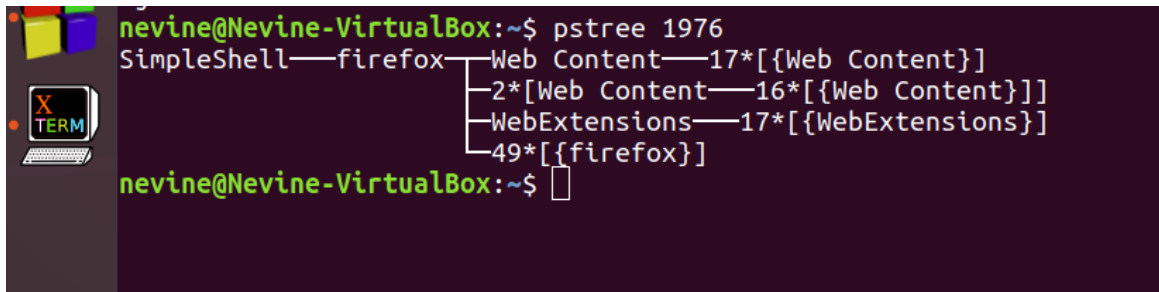


Figure 4: Processes heirarchy