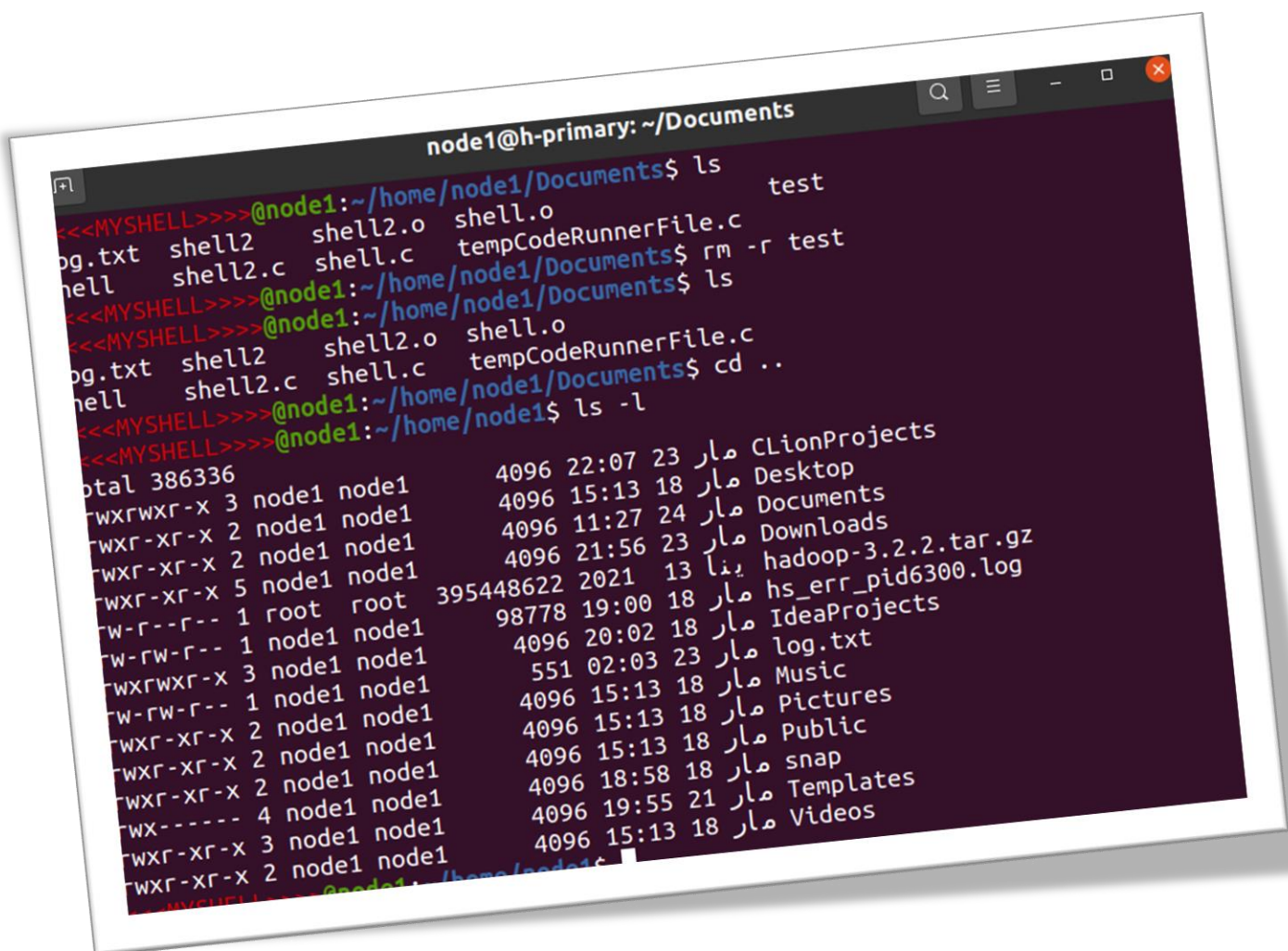


Operating System

Lab 1

Simple Shell



```
node1@h-primary: ~/Documents
<<<MYSHELL>>>@node1:~/home/node1/Documents$ ls
pg.txt  shell2.o  shell.o  tempCodeRunnerFile.c
hell    shell2.c  shell.c
<<<MYSHELL>>>@node1:~/home/node1/Documents$ rm -r test
<<<MYSHELL>>>@node1:~/home/node1/Documents$ cd ..
<<<MYSHELL>>>@node1:~/home/node1$ ls -l
total 386336
-rwxrwxr-x 3 node1 node1 4096 22:07 23 مار CLionProjects
-rwxr-xr-x 2 node1 node1 4096 15:13 18 مار Desktop
-rwxr-xr-x 2 node1 node1 4096 11:27 24 مار Documents
-rwxr-xr-x 5 node1 node1 4096 21:56 23 مار Downloads
-rw-r--r-- 1 root  root  395448622 2021 13 ينا hadoop-3.2.2.tar.gz
-rw-rw-r-- 1 node1 node1 98778 19:00 18 مار hs_err_pid6300.log
-rwxrwxr-x 3 node1 node1 4096 20:02 18 مار IdeaProjects
-rw-rw-r-- 1 node1 node1 551 02:03 23 مار log.txt
-rwxr-xr-x 2 node1 node1 4096 15:13 18 مار Music
-rwxr-xr-x 2 node1 node1 4096 15:13 18 مار Pictures
-rwxr-xr-x 2 node1 node1 4096 15:13 18 مار Public
-rwxr-xr-x 2 node1 node1 4096 18:58 18 مار snap
-rwx----- 4 node1 node1 4096 19:55 21 مار Templates
-rwxr-xr-x 3 node1 node1 4096 15:13 18 مار Videos
-rwxr-xr-x 2 node1 node1
```

Name: Omar Mohamed El-Sayed Metmwah

ID: 19016082

Problem To Be Solved

It is required to implement a Unix shell program. A shell is simply a program that conveniently allows you to run other programs.

The shell must support the following commands:

- "exit"**: which terminates the shell
- A command with no arguments, ex **"ls"**
- A command with arguments, ex **"ls -l"**
- A command, with or without arguments, executed in the background using **&**.
- "cd"**: covering all cases(**cd | cd ~ | cd .. | cd absolute path | cd relative path**)
- "echo"**: prints the input after evaluating all expressions
- "export"**: set values to variables and print variables values.

Organization Of Code

The flow of the code mainly run in `shell()` function, but first I had to prepare the environment by:

- moving the directory to the current directory of the opening shell.

```
void setupEnvironment()
{
    char cwd[1024];
    getcwd(cwd, sizeof(cwd));
    chdir(cwd);
}
```

- set the action for the parent after receiving end signal from the child "SIGCHLD" signal. And the action we chose to happen is to wait to kill the child and make sure there is no zombie process and also to write into log file to count the number of launched processes after finish.

```
signal(SIGCHLD, onChildExit);
```

```
void onChildExit()
{
    FILE *f = fopen("log.txt", "a");
    int stat;
    wait(&stat);
    fprintf(f, "%s", "Child Terminat\n");
    fclose(f);
}
```

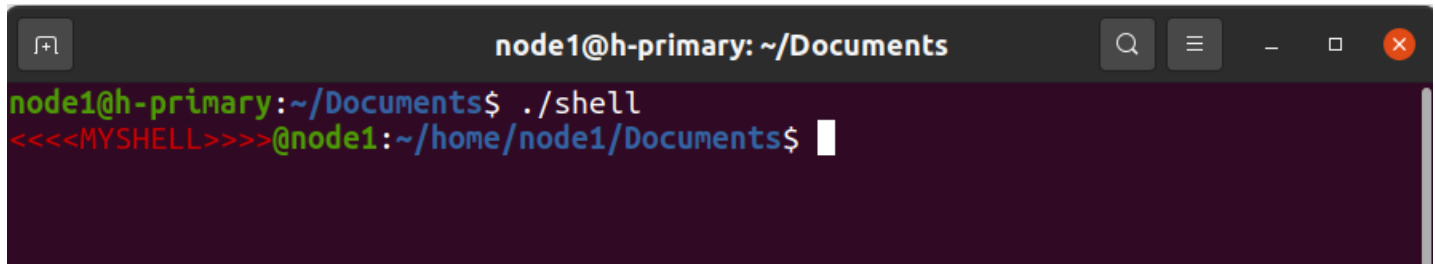
- after preparing everything we ran the `shell()` function which almost contains most of the logic to control the shell.

Its logic of flow is abstracted in this pseudocode:

```
function shell()
    while command_is_not_exit
        print_directory()
        read_input()
        evaluate_environment_variables_in_expression_if_any
        prepare_command_by_parsing();
    switch(input_type):
        case shell_builtin:
            execute_shell_builtin();
        case executable_or_error:
            execute_command();
```

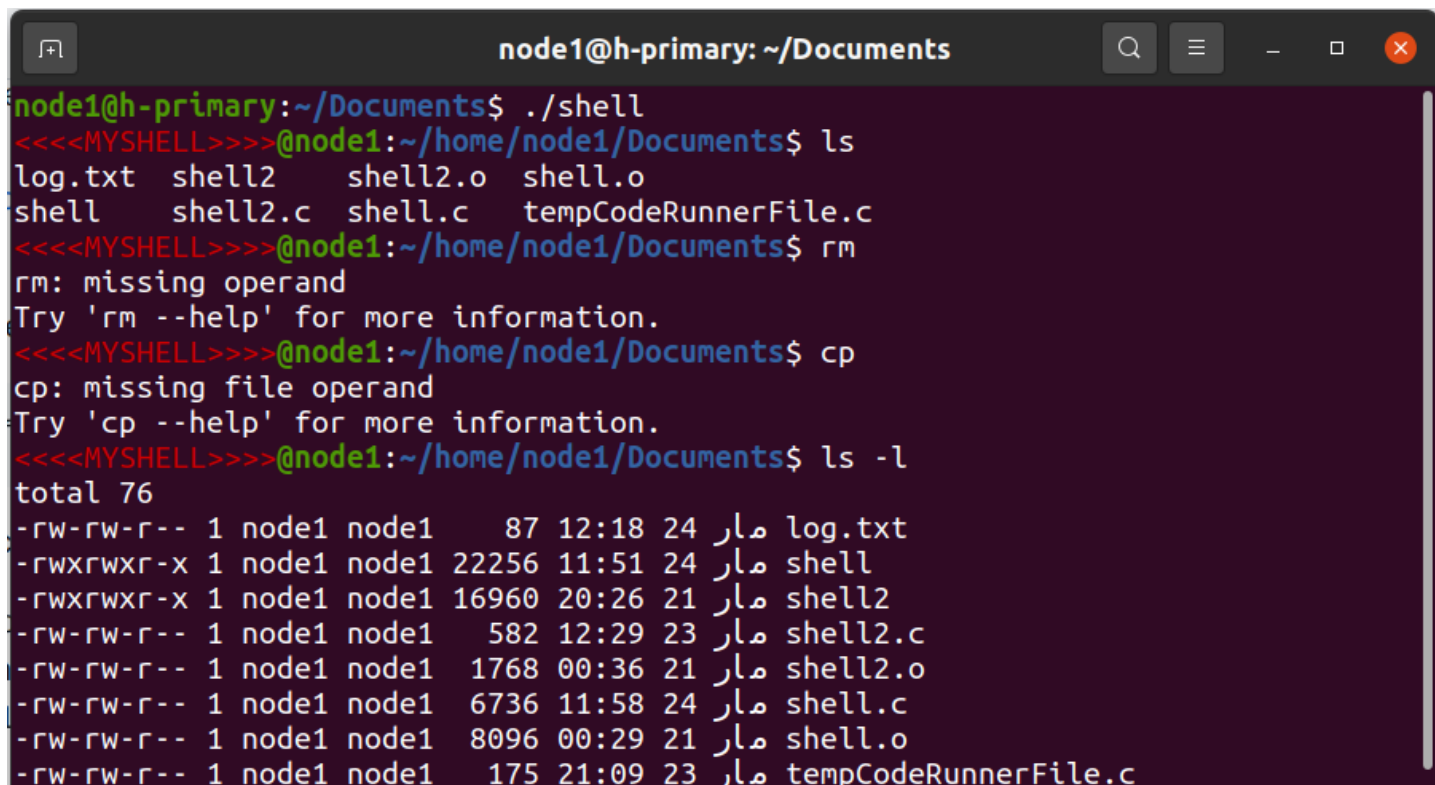
Sample Runs

•start my shell program



```
node1@h-primary: ~/Documents
node1@h-primary:~/Documents$ ./shell
<<<<MYSHELL>>>>@node1:~/home/node1/Documents$
```

•Test commands with no arguments and with arguments



```
node1@h-primary:~/Documents$ ./shell
<<<<MYSHELL>>>>@node1:~/home/node1/Documents$ ls
log.txt  shell2  shell2.o  shell.o
shell    shell2.c  shell.c  tempCodeRunnerFile.c
<<<<MYSHELL>>>>@node1:~/home/node1/Documents$ rm
rm: missing operand
Try 'rm --help' for more information.
<<<<MYSHELL>>>>@node1:~/home/node1/Documents$ cp
cp: missing file operand
Try 'cp --help' for more information.
<<<<MYSHELL>>>>@node1:~/home/node1/Documents$ ls -l
total 76
-rw-rw-r-- 1 node1 node1  87 12:18 24 مار log.txt
-rwxrwxr-x 1 node1 node1 22256 11:51 24 مار shell
-rwxrwxr-x 1 node1 node1 16960 20:26 21 مار shell2
-rw-rw-r-- 1 node1 node1  582 12:29 23 مار shell2.c
-rw-rw-r-- 1 node1 node1 1768 00:36 21 مار shell2.o
-rw-rw-r-- 1 node1 node1 6736 11:58 24 مار shell.c
-rw-rw-r-- 1 node1 node1 8096 00:29 21 مار shell.o
-rw-rw-r-- 1 node1 node1  175 21:09 23 مار tempCodeRunnerFile.c
```

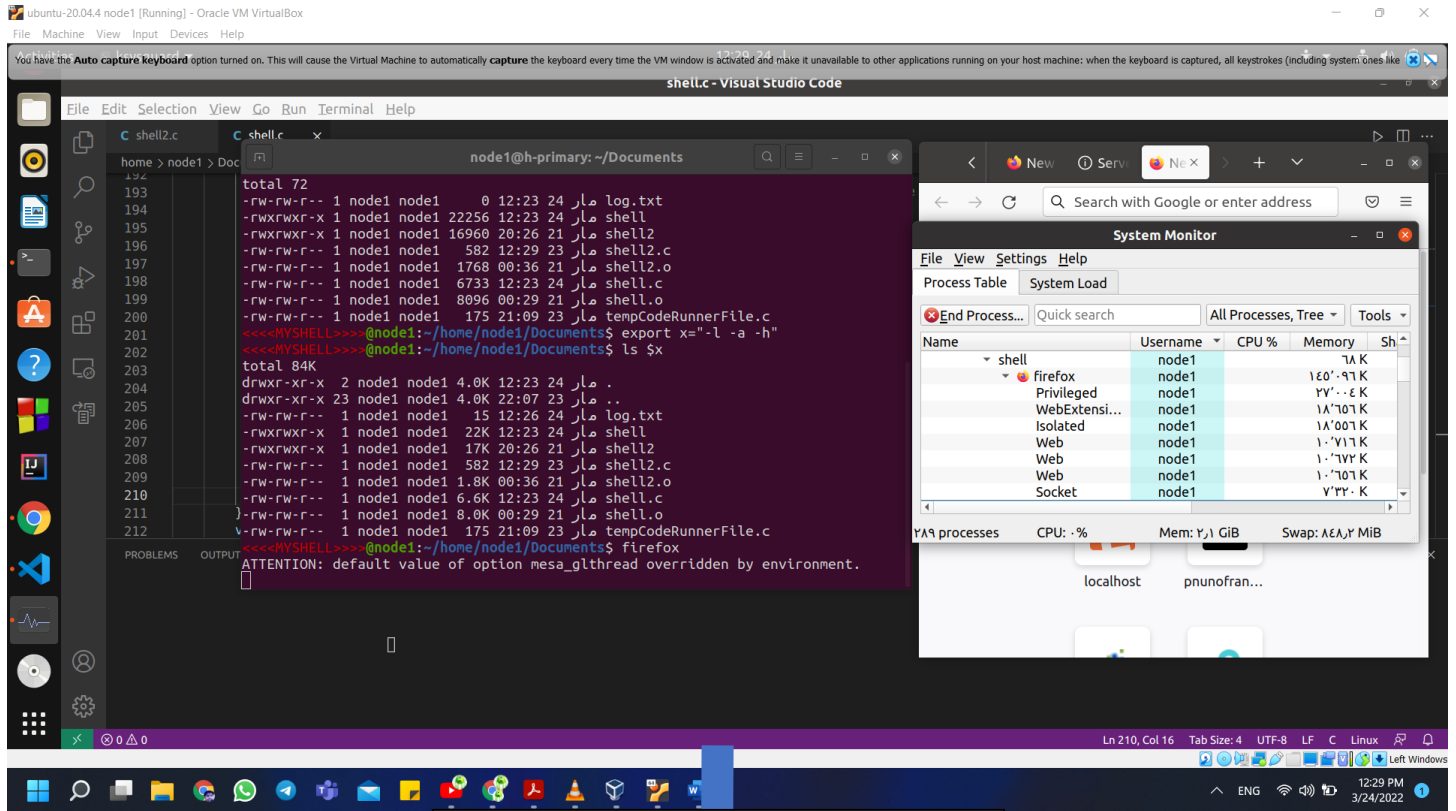
•Test cd cases

```
<<<<MYSHELL>>>>@node1:~/home/node1/Documents$ cd
<<<<MYSHELL>>>>@node1:~/home/node1/Documents$ cd ..
<<<<MYSHELL>>>>@node1:~/home/node1$ cd /home/node1/Documents
<<<<MYSHELL>>>>@node1:~/home/node1/Documents$ cd ~
<<<<MYSHELL>>>>@node1:~/home/node1$ ls
CLionProjects  Downloads          IdeaProjects  Pictures  Templates
Desktop        hadoop-3.2.2.tar.gz log.txt       Public    Videos
Documents      hs_err_pid6300.log Music          snap
<<<<MYSHELL>>>>@node1:~/home/node1$ cd Documents
<<<<MYSHELL>>>>@node1:~/home/node1/Documents$ cd ../Downloads
<<<<MYSHELL>>>>@node1:~/home/node1/Documents$ cd ../Downloads
<<<<MYSHELL>>>>@node1:~/home/node1/Downloads$
```

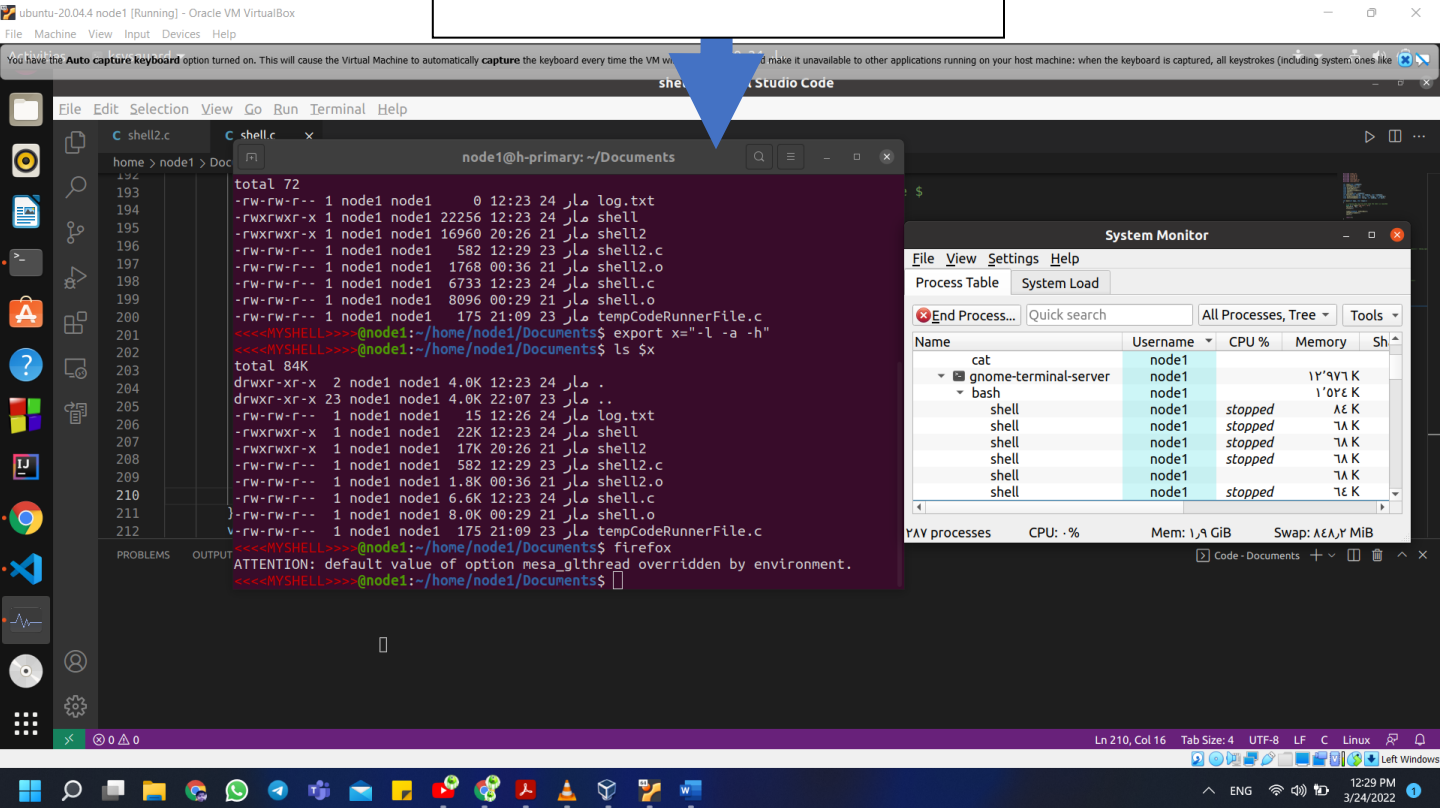
•Test “echo” and “export”

```
<<<<MYSHELL>>>>@node1:~/home/node1/Documents$ export x="omar"
<<<<MYSHELL>>>>@node1:~/home/node1/Documents$ echo "$x"
omar
<<<<MYSHELL>>>>@node1:~/home/node1/Documents$ echo "hello $x"
hello omar
<<<<MYSHELL>>>>@node1:~/home/node1/Documents$ echo "hello"
hello
<<<<MYSHELL>>>>@node1:~/home/node1/Documents$ echo "hello $x mohamed"
hello omar mohamed
<<<<MYSHELL>>>>@node1:~/home/node1/Documents$ export x=-l
<<<<MYSHELL>>>>@node1:~/home/node1/Documents$ ls $x
total 72
-rw-rw-r-- 1 node1 node1 0 12:23 24 مار log.txt
-rwxrwxr-x 1 node1 node1 22256 12:23 24 مار shell
-rwxrwxr-x 1 node1 node1 16960 20:26 21 مار shell2
-rw-rw-r-- 1 node1 node1 582 12:29 23 مار shell2.c
-rw-rw-r-- 1 node1 node1 1768 00:36 21 مار shell2.o
-rw-rw-r-- 1 node1 node1 6733 12:23 24 مار shell.c
-rw-rw-r-- 1 node1 node1 8096 00:29 21 مار shell.o
-rw-rw-r-- 1 node1 node1 175 21:09 23 مار tempCodeRunnerFile.c
<<<<MYSHELL>>>>@node1:~/home/node1/Documents$ export x="-l -a -h"
<<<<MYSHELL>>>>@node1:~/home/node1/Documents$ ls $x
total 84K
drwxr-xr-x 2 node1 node1 4.0K 12:23 24 مار .
drwxr-xr-x 23 node1 node1 4.0K 22:07 23 مار ..
-rw-rw-r-- 1 node1 node1 15 12:26 24 مار log.txt
-rwxrwxr-x 1 node1 node1 22K 12:23 24 مار shell
-rwxrwxr-x 1 node1 node1 17K 20:26 21 مار shell2
-rw-rw-r-- 1 node1 node1 582 12:29 23 مار shell2.c
-rw-rw-r-- 1 node1 node1 1.8K 00:36 21 مار shell2.o
-rw-rw-r-- 1 node1 node1 6.6K 12:23 24 مار shell.c
-rw-rw-r-- 1 node1 node1 8.0K 00:29 21 مار shell.o
-rw-rw-r-- 1 node1 node1 175 21:09 23 مار tempCodeRunnerFile.c
<<<<MYSHELL>>>>@node1:~/home/node1/Documents$
```


- **Test a command, with or without arguments, executed in the background using &**



Note that the shell didn't continue till the firefox is closed when we don't use &



“rm” and then with “rm -r” and then use “ls” again to make sure it is removed then make it again so this add **3 more terminated children** to log file.

And also, I need to mention that the “attention” message that appears after opening firefox is always shown even with the base shell of Ubuntu, It might be because of some new releases or something else but this has no thing with my shell.

<https://youtu.be/yqf7q92BMMo>