

Testing for initial piping implementation:

```
buffalo@makg_pc:~/COMP304_Project_1$ gcc -o shell-ish shellish-skeleton.c
buffalo@makg_pc:~/COMP304_Project_1$ ./shell-ish
buffalo@makg_pc:/home/buffalo/COMP304_Project_1 shellish$ ls | grep shell
grep: cannot access 'shell': No such file or directory
buffalo@makg_pc:/home/buffalo/COMP304_Project_1 shellish$ ls -la | grep shellish
grep: cannot access 'shellish': No such file or directory
buffalo@makg_pc:/home/buffalo/COMP304_Project_1 shellish$ ls | wc
new.txt new2.txt new4.txt notes output.txt shell-ish shellish-skeleton.c tests
buffalo@makg_pc:/home/buffalo/COMP304_Project_1 shellish$ ls
new.txt new2.txt new4.txt notes output.txt shell-ish shellish-skeleton.c tests
buffalo@makg_pc:/home/buffalo/COMP304_Project_1 shellish$ ls | grep notes
Skeleton_code_analysis.pdf part1_notes.pdf
buffalo@makg_pc:/home/buffalo/COMP304_Project_1 shellish$
buffalo@makg_pc:~/COMP304_Project_1$ nano shellish-skeleton.c
buffalo@makg_pc:~/COMP304_Project_1$ gcc -o shell-ish shellish-skeleton.c
buffalo@makg_pc:~/COMP304_Project_1$ ./shell-ish
buffalo@makg_pc:/home/buffalo/COMP304_Project_1 shellish$ ls | wc
      8      8      79
buffalo@makg_pc:/home/buffalo/COMP304_Project_1 shellish$ ls | grep notes
notes
```

Note: First few tests were not giving the expected output.

Issue: in the **snprintf** function of “pid_pipe_right” the path was not correctly implemented.

changed: `snprintf(full_path, sizeof(full_path), %s/%s, directories, command->name);`

to: `snprintf(full_path, sizeof(full_path), %s/%s, directories, command->next->name);`

Recursive calls do not work.

Fix: Added a **command->next->next** check that creates a new process_command call if true.

```
buffalo@makg_pc:~/COMP304_Project_1$ nano shellish-skeleton.c
buffalo@makg_pc:~/COMP304_Project_1$ gcc -o shell-ish shellish-skeleton.c
buffalo@makg_pc:~/COMP304_Project_1$ ./shell-ish
buffalo@makg_pc:/home/buffalo/COMP304_Project_1 shellish$ ls -la | grep notes | wc
1      9      54
buffalo@makg_pc:/home/buffalo/COMP304_Project_1 shellish$ ls | random_command
ls: command not found
buffalo@makg_pc:/home/buffalo/COMP304_Project_1 shellish$ ls | grep
Usage: grep [OPTION]... PATTERNS [FILE]...
Try 'grep --help' for more information.
buffalo@makg_pc:/home/buffalo/COMP304_Project_1 shellish$ ls -la | grep shellax | wc
0      0      0
buffalo@makg_pc:/home/buffalo/COMP304_Project_1 shellish$ cat shellish-skeleton.c | grep include | grep sys | wc
1      2      22
```