```
                    program
                    directory
            ┌──────────┴──────────┐
          Dir0                  shell*
    ┌───────┼───────┬───────────┐
   t1      t2       t2        Dir1
```

The object of this assignment is to gain experience with some advanced programming techniques like process creation and control, file descriptors, signals and possibly pipes. To do this, a team of 5 students will be working together to write their own command shell - much like csh, bsh or the DOS command shell. During the presentation, each student has to:

  I.   run their part of the code

  II.   talk about their code development difficulties

  III.   share suggestions to improve their part of codes.

  IV.   be ready for 1 question their code

Each member the team will write the code for one of the following customized commands:

1. **shell***: the environment that will execute the rest of nee command. the shell won't terminate unless the exit command gets executed. The shell displays a different command prompt (i.e. something other than **$** and **>**).

2. **tree***: this new command will create a directory and call it Dir0. Then it will change the working directory to Dir0, and create three empty text files namely; t1.txt, t2.txt,and t3.txt, and one empty directory , called Dir1,inside it.

3. **list***: this new command will clear the terminal screen and print a detailed list of all content of the current directory (similar to ls -l) to the terminal and t1.txt. Finally, it will change the name of text file to tree.txt.

4. **path***: this new command will print the path of the current directory to the terminal and t2.txt, and change the name of text file to path-info.txt. Concatenate the content of tree.txt and path.txt into t3.txt and change the last to log.txt. Finally, delete tree and path text files

5. **exit***: this new command will print a list the last 4 commands to the terminal, a detailed list of all content of the current directory (similar to ls -l). Finally, it will wait for the user to hit the "return" key to terminate the shell and return the control to the original shell program on your machine.