Project 03 :
Command-line Executor

Objectives:

In this project, you will gain experience handling basic functionalities of an OS like file redirection, process & program execution, & inter-process communication.

Requirements:

For this project you are not bounded to specific programming constraints as long as your project demonstrates the following :

1) File Redirection (Input Output(Truncate/Append) Error).
    Example System Calls : open(), creat(), dup() & dup2() .

2) Process Creation & Program Execution
    Example System Calls : fork() & execlp() execvp() wait() exit()

3) Inter Process Communication .
    Example System Calls : pipe(),close()

4) Signal Handlers
    Example System Calls : sigaction()

The project can be a group project with up-to three members.

Addition Requirement :

For this project you have to use a version control system. If you're not familiar with what a VCS is, it's basically a standard way of managing your code. You can read more about it on the internet.
You can use BitBucket to manage your git repositories as it provides private repositories out of the box for free.

Note : Contribution of each member should be highlighted in the source code. Each member should be committing to the repository with his/her account. Marks will be deducted if usage of VCS is not satisfactory. Like uploading the project to the VCS the day before submission or no timely commits.

I have attached a sample program demonstrating all the system calls

Grading Criteria:

Your grade on this assignment will depend upon:

1. This project is worth 50% of your Total Project Marks.
2. Following the coding standards (posted on GitHub).
3. Fulfilling project requirements.
4. Performing the required processing correctly and displaying the correct results
5. Handling all corner cases correctly
6. If your project does not compile, it's an AUTOMATIC ZERO, no exceptions
7. Performing validation checks on your input. There should be no crashes or incorrect results due to improper input. However, for any numeric arguments, you can always assume that your program will be only tested with numbers, NOT any other non-numeric input
8. During testing, your program should not crash; marks will be deducted for any crashes during testing.