**CS 312**

**Assignment 2**

**2021193 Hamza Ahmed Zuberi**

**Task 1:**

**A screen shot of a computer program

Description automatically generated**

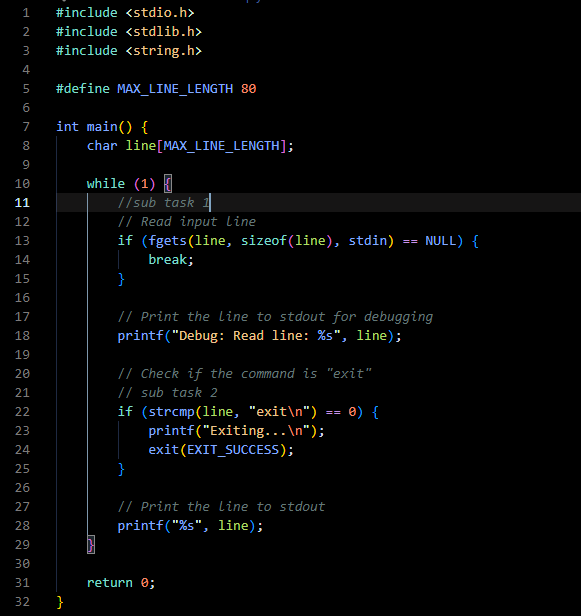
**Output:**

**A screenshot of a computer program

Description automatically generated**

After compiling the code, we debug the code by using the ‘-g’ command. It will ask you to enter input lines one by one. Each line we will enter will be printed to stdout with a debugging message prefixed to it.

**Task 2:**

****

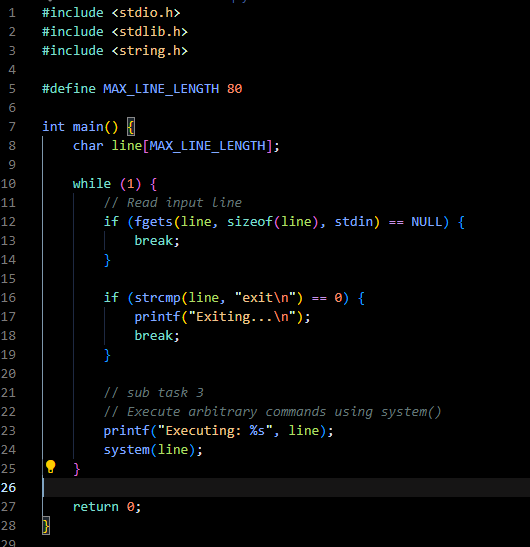
**Output:**

**A computer screen with text on it

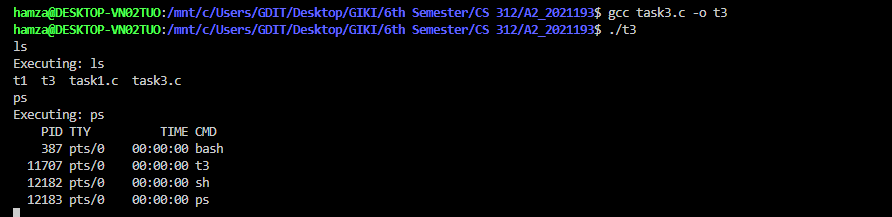
Description automatically generated**

Here we modified the code by adding a “strcmp” function to compare the input string if it matches with the string exit. If a match is found, the program exits with a status of “EXIT\_SUCCESS”, or else it continues to read and print input lines.

**Task 3:**

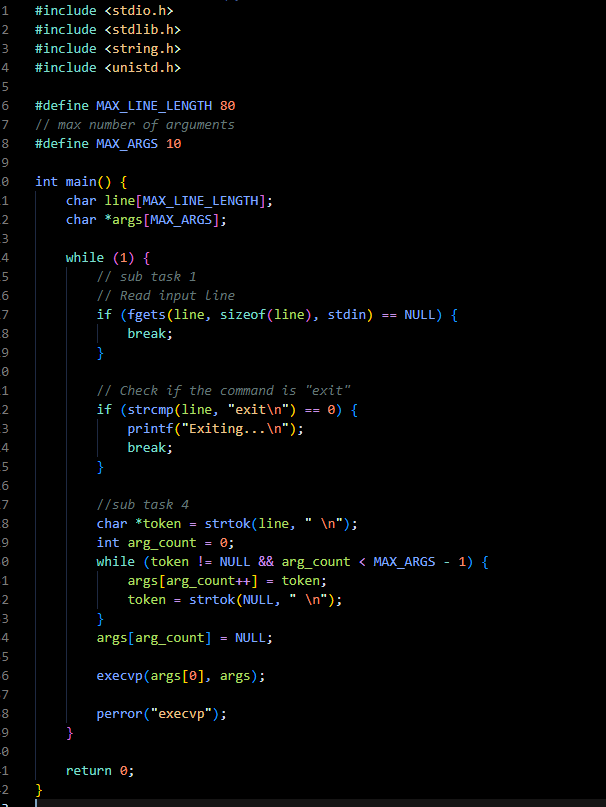
****

**Output:**

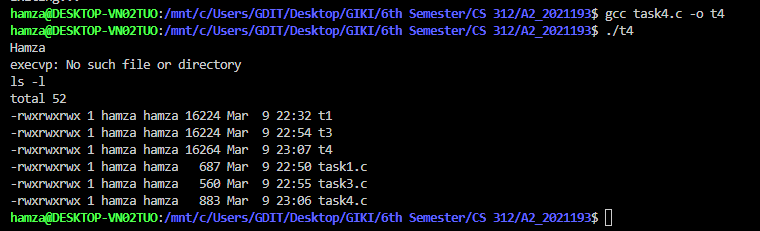
****

After compiling the code, we can use arbitrary commands such as “ls” to list all the files and directories in the current directory or the “ps” command to show the status of the active processes. To exit we can use the exit command as shown earlier.

**Task 4:**

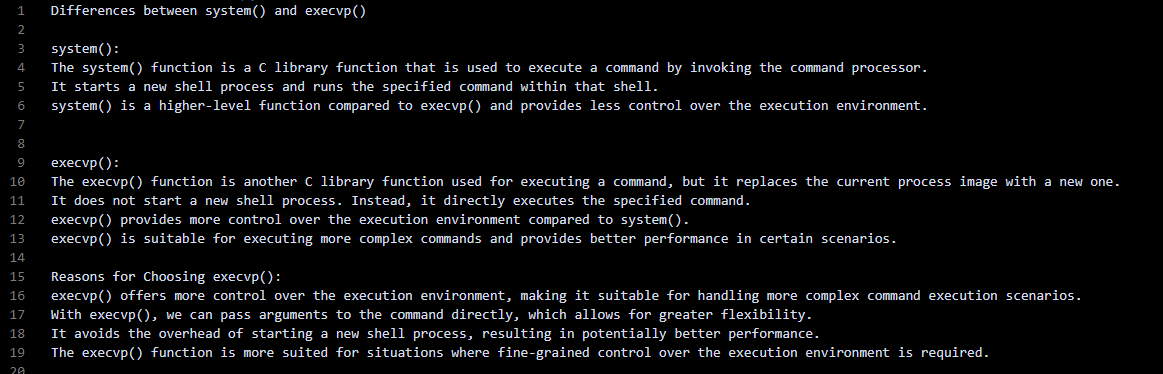
****

**Output:**

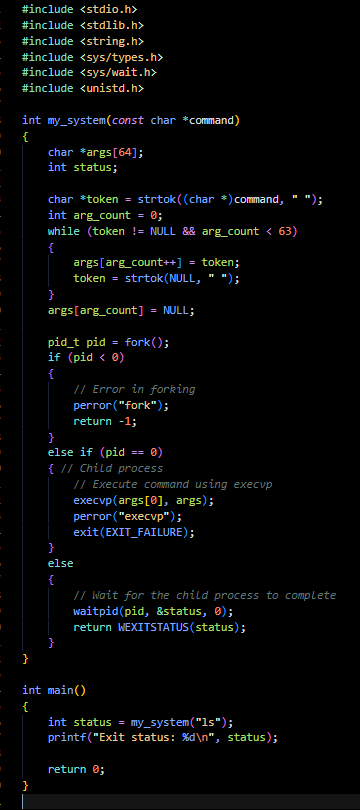
****

Here we replaced the “system()” command with “execvp()” command. Once the program is running, we can enter arbitrary commands such as here we have used “ls -l” to list the files and directories with more details. If we don’t pass the necessary arguments, it prints an error message using “perror” as shows above.

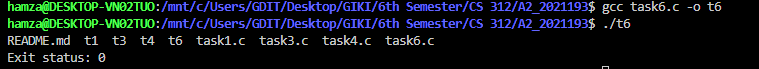
**Task 5:**

****

**Task 6:**

****

**Output:**

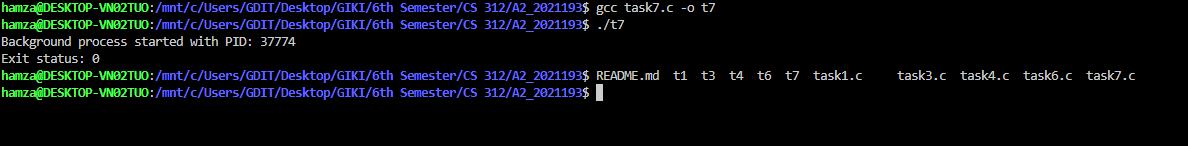
****

This will execute the “ls” command using my\_system function, which internally uses fork() and execvp() to execute the command. After executing the command, it will print the exit status of the command to the console.

**Task 7:**



**Output:**

****

This will execute the “ls &” command in the background. It also shows the PID as well.