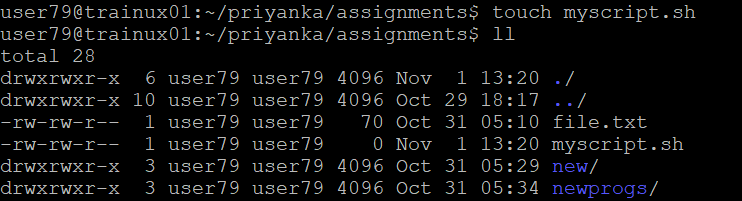
**ASSIGNMENT -3**

**Pipes and Redirection Operators**

1. **Create a script named “myscript” in current directory to do the following.**

First create ‘myscript.sh’ in current directory by using ‘touch’ command.



1. **create a dir named “subd1” in current directory**

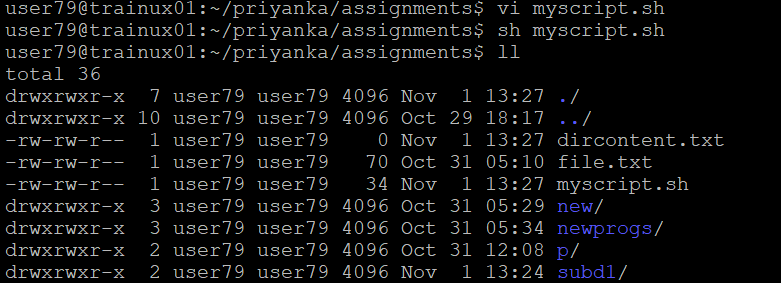
Create a sub directory in current directory by using ‘myscript.sh’. First we have to open ‘myscript.sh’ by using ‘vi editor’ in that write the command ‘mkdir subd1’. It creates the subdirectory ‘subd1’. To execute the ‘myshell.sh’ we have to write ‘ sh filename’.

A screenshot of a computer screen

Description automatically generated

1. **create a file named “dircontent” with contents of the current directory**

We know that to create a text file we have to use ‘touch’ command. So, open ‘myscript.sh’ by using vi editor in that create a textfile name ‘dircontent’. Then run the shell script by using command ‘sh myscript.sh’.



1. **display the contents of the file “dircontent”**

To display the contents of the file we have to use ‘cat’ command. So, open the myscript by using vi editor and then write the cat command in that then close the shell script. After execute the shell script by using ‘sh myscript.sh’. It displays the content in the file.

A black screen with white text

Description automatically generated

1. **display the contents of a non-existing file named “unknownfile”**

When we execute the command ‘cat unknown.txt’ in myscript.sh it gives an error that is no such file or directory because there is no such a file is exist in the current directory.

**A black background with white text

Description automatically generated**

These are commands are used in ‘myscript.sh’ for above cases.

**A screen shot of a computer

Description automatically generated**

1. **Run the script and validate the output in following cases**
2. **Redirect only the stdout to an o/p file named stdout.txt**

To redirect only the standard output (stdout) to an output file named ‘stdout.txt’, we have to use ‘exec > stdout.txt’ command.

A black background with white text

Description automatically generated

First, open ‘myscript.sh’ by using vi editor. Then write the commands in it as shown below.

A screen shot of a computer screen

Description automatically generated

After run the ‘myscript.sh’ by using ‘sh’ command then it creates a file ‘stdout.txt’ it contains the output. While running the ‘myscript.sh’ in terminal it gives the error and the output is contained in ‘stdout.txt’.

A black background with white text

Description automatically generated

1. **Redirect only the stderr to an o/p file named stderr.txt**

To redirect only the standard error(stderr) to an output file named ‘stderr.txt’, we have to the command ‘exec 2> stderr.txt’

A black background with white text

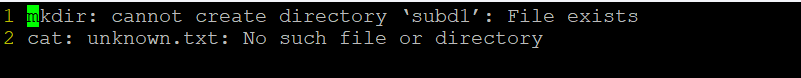
Description automatically generated

Open the ‘myscript.sh’ by using vi editor then write commands shown below. After we have to run the ‘myscript.sh’ by using ‘sh’ command.

A screen shot of a computer screen

Description automatically generated

After it creates a file ‘stderr.txt’ which contains the error output of the ‘myshell.sh’. While running in the terminal it gives actual output as shown and the error output is stored in ‘stderr.txt’. The output in ‘stderr.txt’ is shown below.



1. **Redirect both stdout and stderr to an o/p file named stdall.txt**

You can redirect both standard output(stdout) and standard error(stderr) to an output file named ‘stdall.txt’ by using “exec >stdall.txt 2>&1”.

A black background with white text

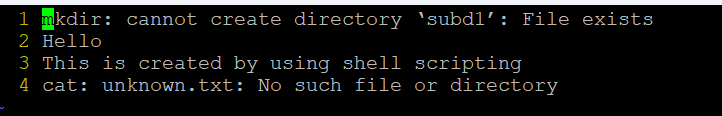
Description automatically generated

Open the ‘myscript.sh’ and write commands shown below to execute.

A screen shot of a computer screen

Description automatically generated

After execute the ‘myscript.sh’ and then it creates the file ‘stdall.txt’ which contains both error and the output. Then open the ‘stdall.txt’ the output is shown in below.



1. **Display all o/p and error and also redirect both stdout and stderr to an o/p file named stdall.txt**

To display all o/p and error in terminal and also redirect both stdout and stderr to an o/p file named stdall.txt, we use the ‘exec > >(tee stdall.txt) 2>&1’.

* The below screenshot shows the commands in ‘myscript.sh’.

**A black screen with white text

Description automatically generated**

A screen shot of a computer screen

Description automatically generated

* The below screenshot shows the stdall.txt file.

**A black background with white text

Description automatically generated**

1. **Redirect the output of command below using pipe (|) to wc and get the output .**

**ls -l**

To redirect the output of the command ‘ls -l’ using a pipe to ‘wc’ to get the output count, we have to use the command ‘ls -l |wc’.

A black background with white text

Description automatically generated

This command will list the files in long format using ‘ls -l’ and then count the number of lines in the output using ‘wc’.