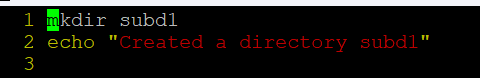
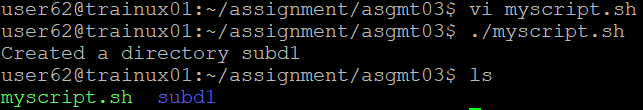
# **PIPESANDREDIRECTIONOPERATORS\_ASSIGNMENT**

1. **Create a script named “myscript” in current directory to do the following.**
   1. **create a dir named “subd1” in current directory:**
      1. First create a myscript.sh using vi command and write the command mkdir subd1 in myscript to create a subd1 in current directory, save it using esc: wq. And run script using ./myscript.sh then subd1 is created.

Output: The content in myscript.sh



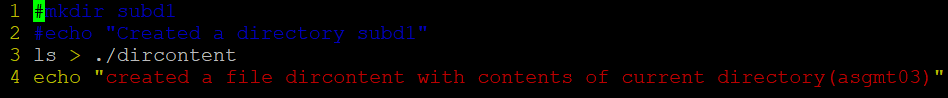


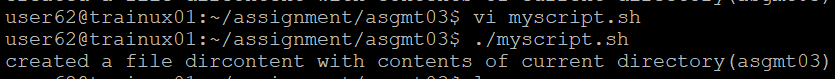
* 1. **create a file named “dircontent” with contents of the current directory:**

1. To create a file dircontent at same time with contents of the current directory we can use the redirection operator “>” stdout. Write

ls > ./dircontent in myscript.

Output: The content in myscript.sh



****

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

1. To display the contents of the file “dircontent" we use cat command. Write “cat dircontent” in myscript.

Output: The content in myscript.sh

A blue text on a black background

Description automatically generated

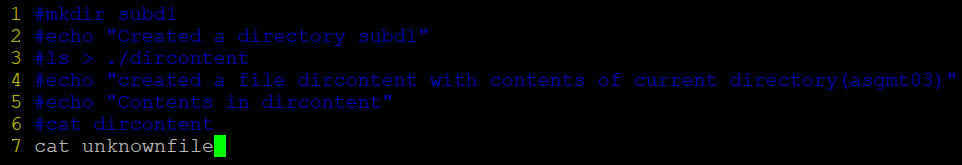
A computer screen with white text

Description automatically generated

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

1. To display the contents of a non-existing file unknounfile, we use “cat” command. Write cat unknownfile in myscript.sh.

Output: The content in myscript.sh



A black background with white text

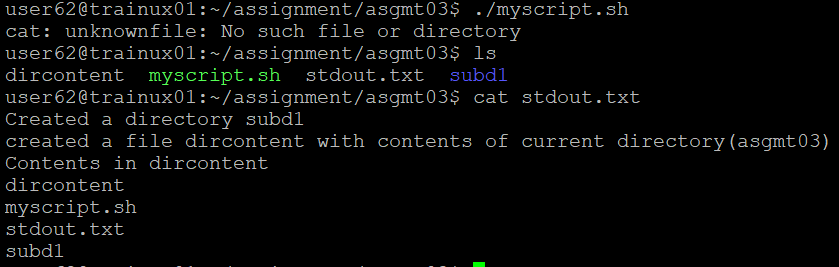
Description automatically generated

1. **Run the script and validate the output in following cases**
   * + 1. **Redirect only the stdout to an o/p file named stdout.txt:**
     1. To redirect only the stdout to an output file stdout.txt, we use redirection operator “>” and “exec” command. Write “exec > stdout.txt” at top of the script, when we run the script this command writes only the output to the stdout.txt file without displaying in terminal.

Output: The content in myscript.sh

A black background with red text

Description automatically generated



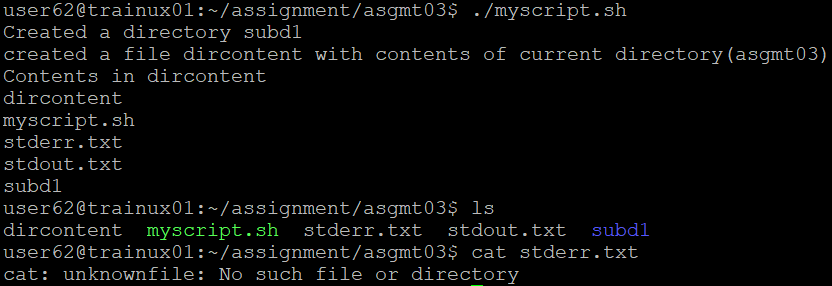
* + - 1. **Redirect only the stderr to an o/p file named stderr.txt:**

1. To redirect only the stderr to an output file stderr.txt, we use redirection operator “2>” and “exec” command. Write “exec 2> stderr.txt” at the top of the script, when we run the script this command writes only the standard error to the stderr.txt file without displaying in terminal.

Output: The content in myscript.sh

A black screen with red text

Description automatically generated

****

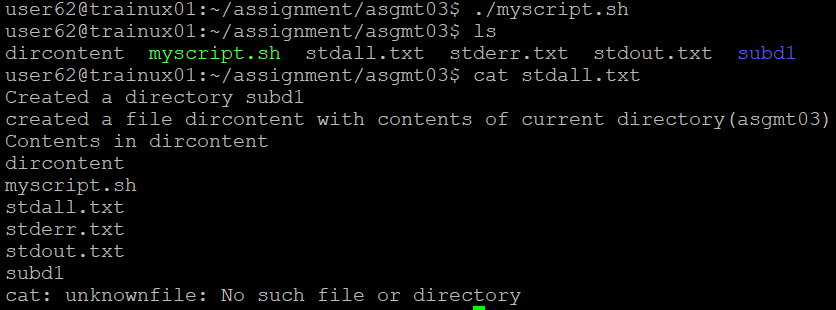
* + - 1. **Redirect both stdout and stderr to an o/p file named stdall.txt:**

1. To redirect both stdout and stderr to an output file stdall.txt, we use redirection operator “&” and “exec” command. Write “exec > stdall.txt 2>&1” at the top of the script, when we run the script this command writes both the output and standard error to the stdall.txt file without displaying in terminal.

Output: The content in myscript.sh

A black screen with red text

Description automatically generated

****

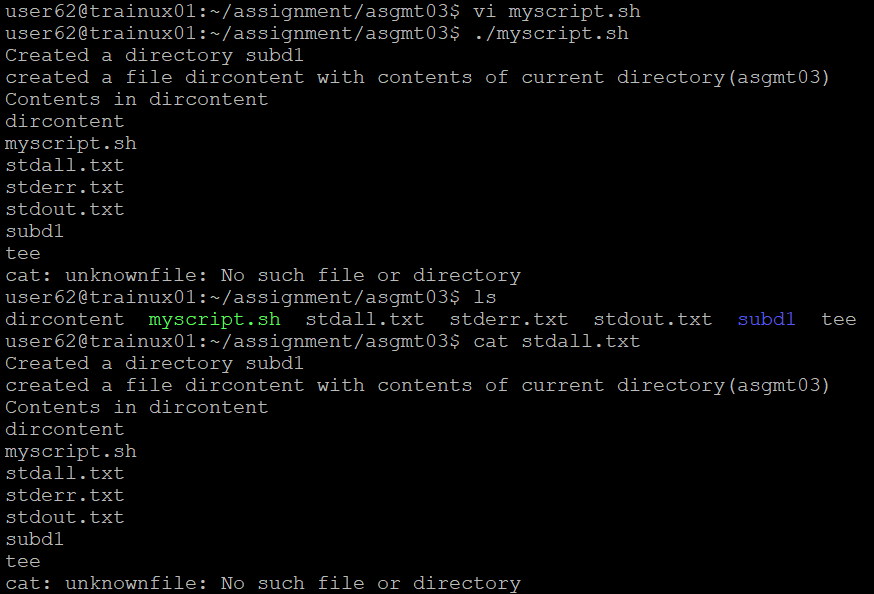
* + - 1. **Display all o/p and error and redirect both stdout and stderr to an o/p file named stdall.txt:**

1. To both display and redirect the output and error we the “tee” command. Write exec > >(tee stdall.txt) 2>&1, when we run the script the output and error displayed and stored in stdall.txt file.

Output: The content in myscript.sh

A screenshot of a computer

Description automatically generated



1. **Redirect the output of command below using pipe (|) to wc and get the output. ls -l?**
2. To redirect the output of command ls -l using pipe to wc, we write $ ls -l | wc and press enter to get output.

Output:

