Assignment 1: Ensure the script checks if a specific file (e.g., myfile.txt) exists in the current directory. If it exists, print "File exists", otherwise print "File not found".

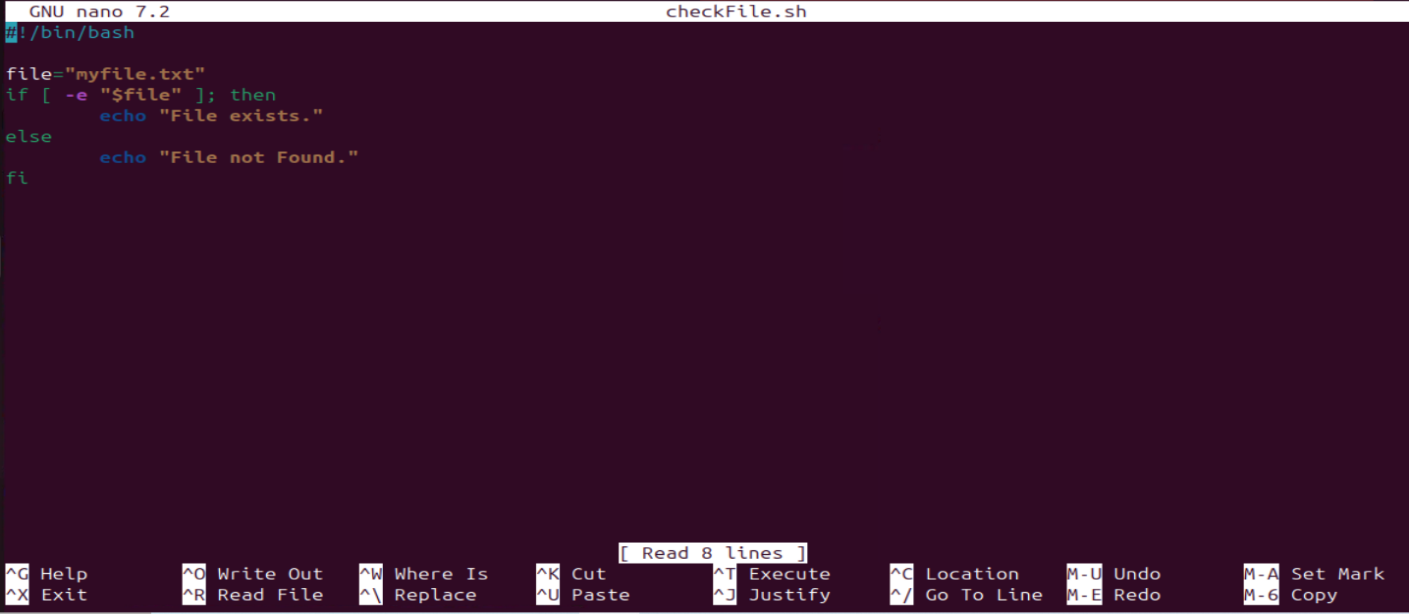
# Shell script for checking a file in current working dir:

* Create a new file named checkFile.sh and provide it ‘execute’ permissions.

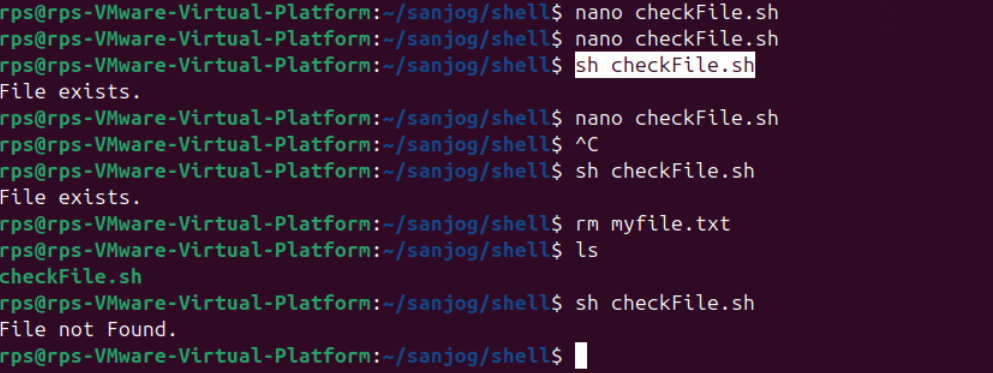
touch chechFile.sh

chmod 755 checkFile.sh

* using nano or other editor make changes to sh file and write code accordingly.



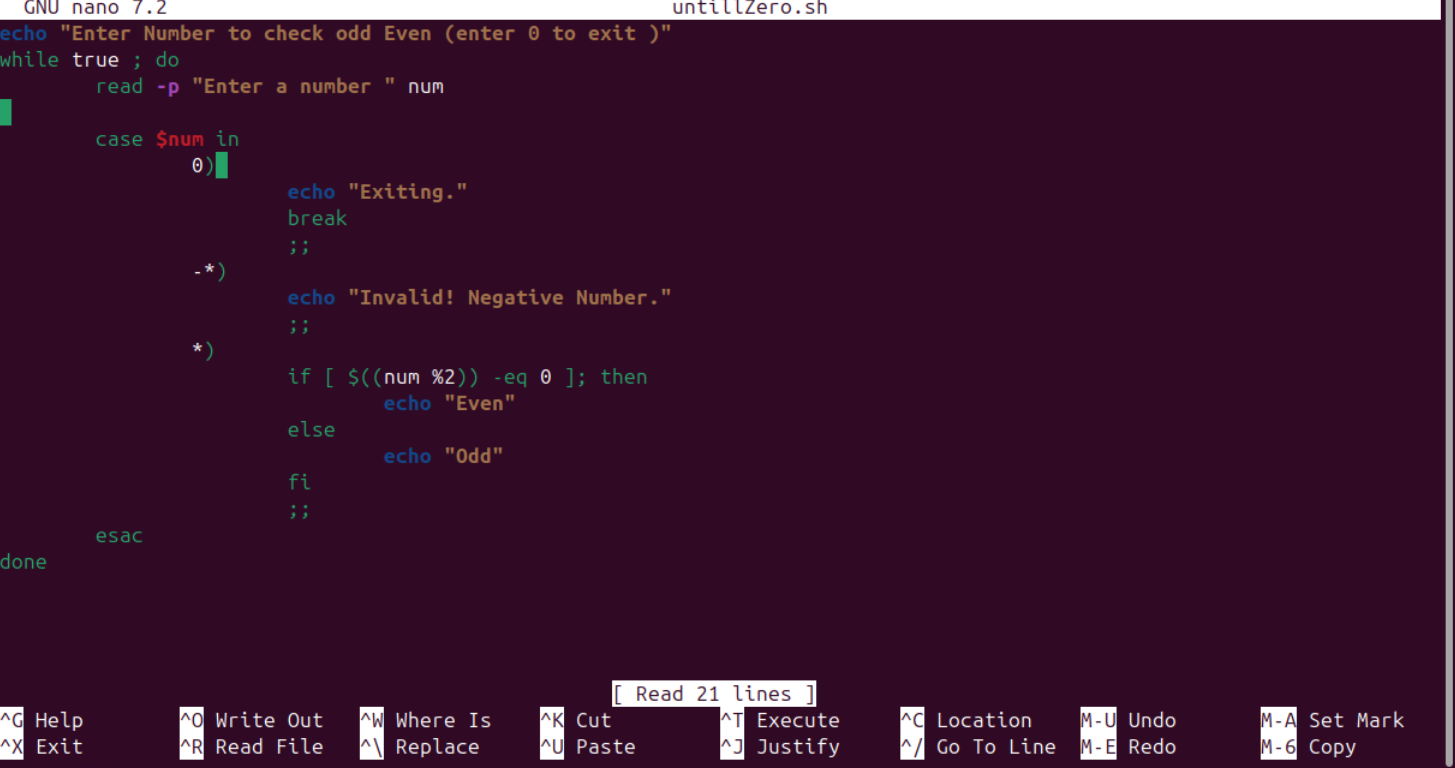
* Execute command.



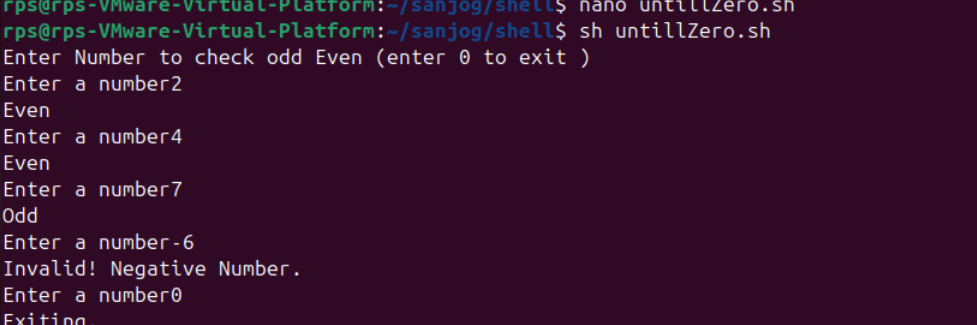
Assignment 2: Write a script that reads numbers from the user until they enter '0'. The script should also print whether each number is odd or even.

# Shell script to read number till they write ‘0’ and check Odd & Even:

* The shell code is shown below.



* Execution:

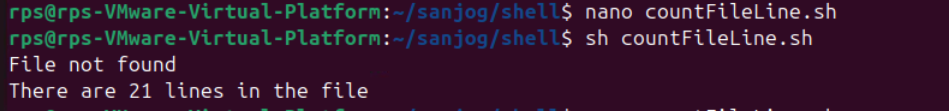


Assignment 3: Create a function that takes a filename as an argument and prints the number of lines in the file. Call this function from your script with different filenames.

# Shell script to get number of lines in file using function:

* Code and execution shown below:

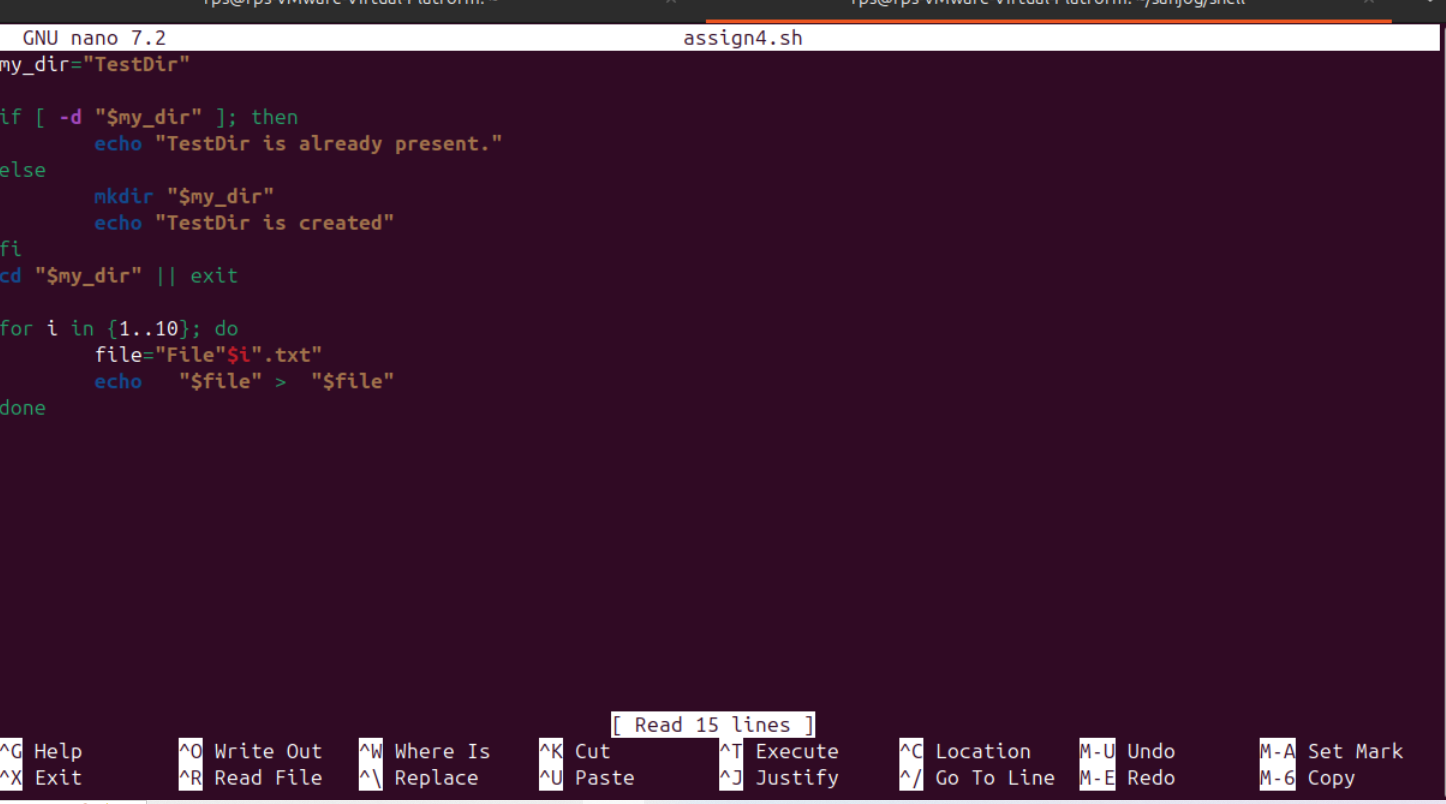


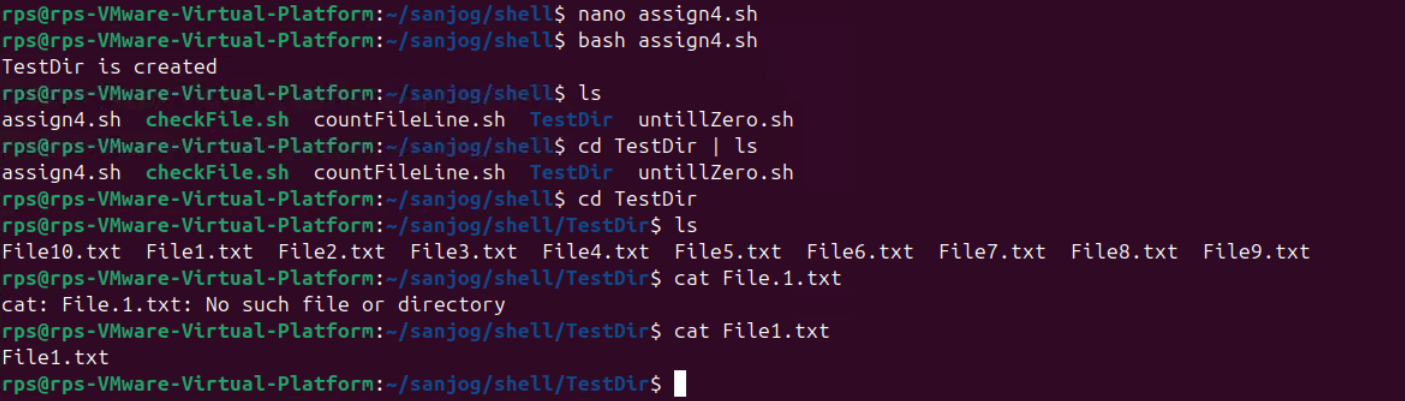


Assignment 4: Write a script that creates a directory named TestDir and inside it, creates ten files named File1.txt, File2.txt, ... File10.txt. Each file should contain its filename as its content (e.g., File1.txt contains "File1.txt").

# Shell script to make files and its content:

* The code and its execution is shown below:



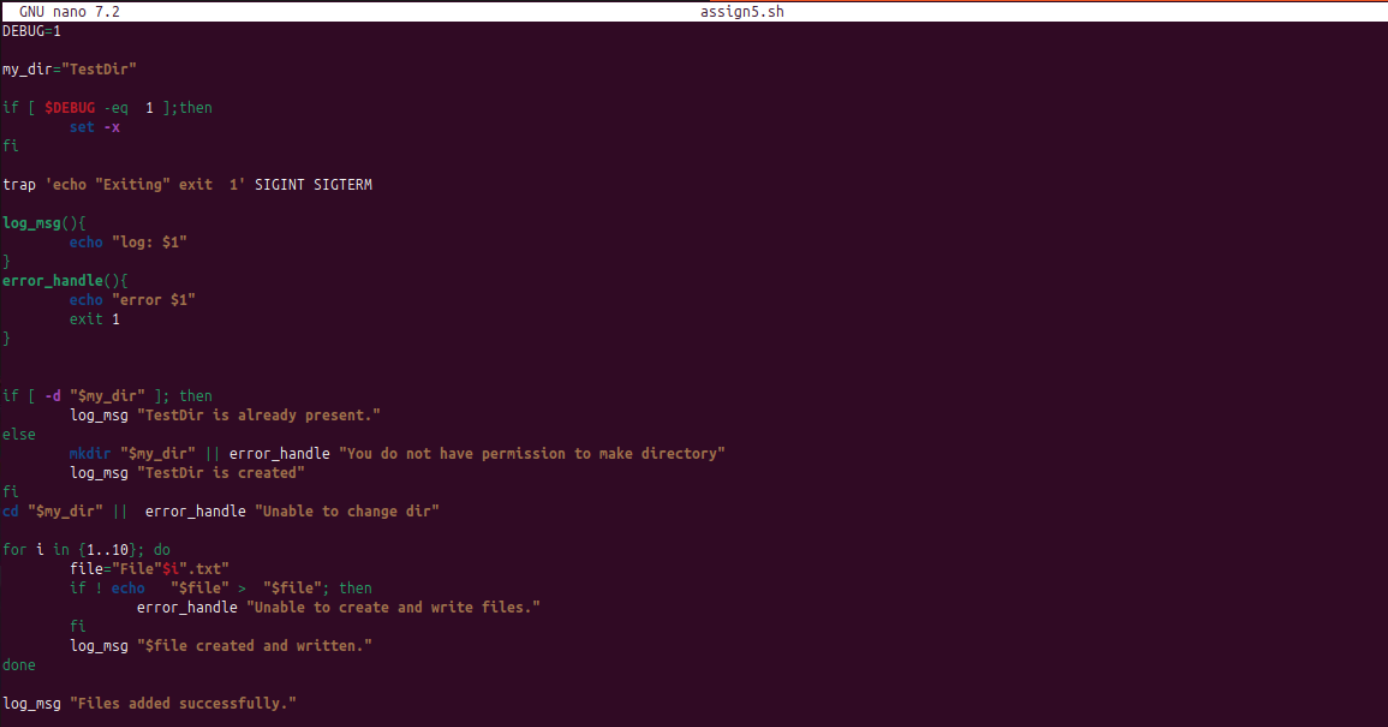


Assignment 5: Modify the script to handle errors, such as the directory already existing or lacking permissions to create files.

Add a debugging mode that prints additional information when enabled.

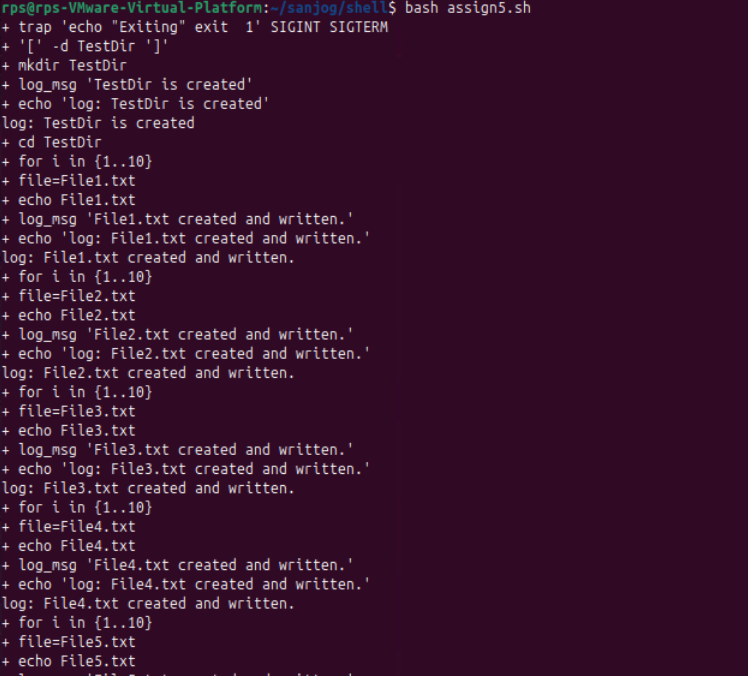
# Shell script to for logging and error handling:

* The code and its execution is shown below:

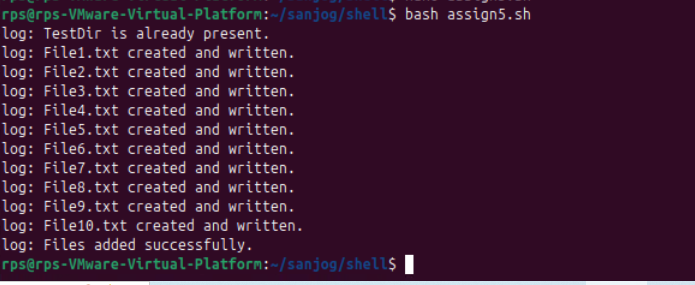




# Execute using Debug=1:



# Execute using Debug=0:



Assignment 6: Given a sample log file, write a script using grep to extract all lines containing "ERROR". Use awk to print the date, time, and error message of each extracted line.

Data Processing with sed

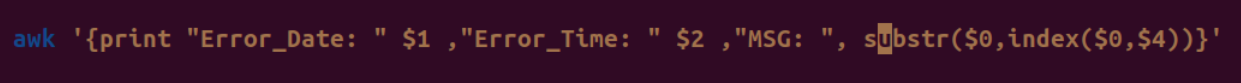
# Extracting Error and using grep and awk command:

* **grep**: it searches file for given cmd
* Now to search “Error” msg on we use condition as follows

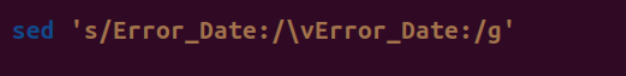
grep pattern filename

ex. grep “ERROR” sample\_logs.log

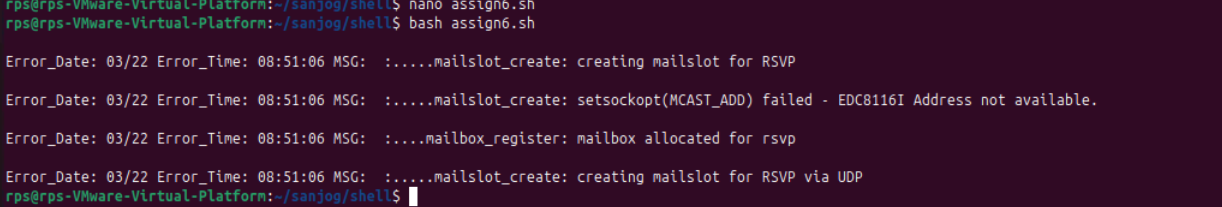
* **|(pipe):** this will return output of previous cmd (grep cmd here) and works as input to RHS.
* **awk:** this command will make structure of given input by mainly using it part or columns.
* **$1**: Refers to the first field or column (i.e., Date for my file) in a line of input from grep.
* **$2:** Refers to the second field which is Time here
* **index ($0, $4):** This will get us this index of word after $3 which is “ERROR” or “INFO” word column
* **substr ($0, index ($0, $4)):** it returns sub string of each line from index from 4th column to end of line.



* After getting the required data we can now process to show better to user using sed
* sed: it is a stream editor used for text transformations.



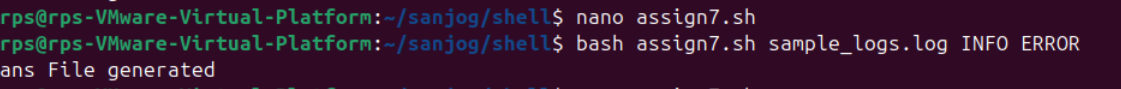
* s: Stands for substitution.
* Error\_Date: Target text to search.
* \v: vertical tab.
* \vError\_Date: add a vertical tab before Error Date text.
* g: Applies the substitution globally to all matches in the input like regular regex.

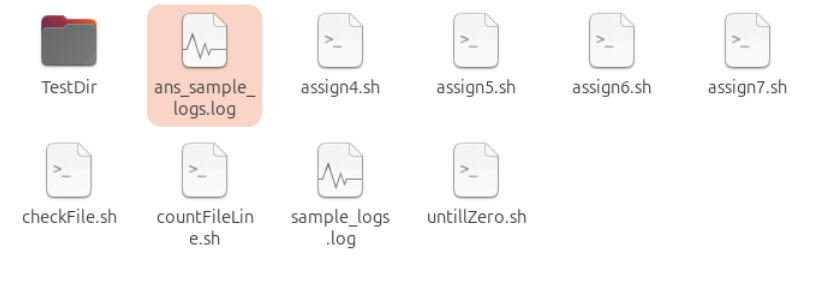


Assignment 7: Create a script that takes a text file and replaces all occurrences of "old\_text" with "new\_text". Use sed to perform this operation and output the result to a new file.

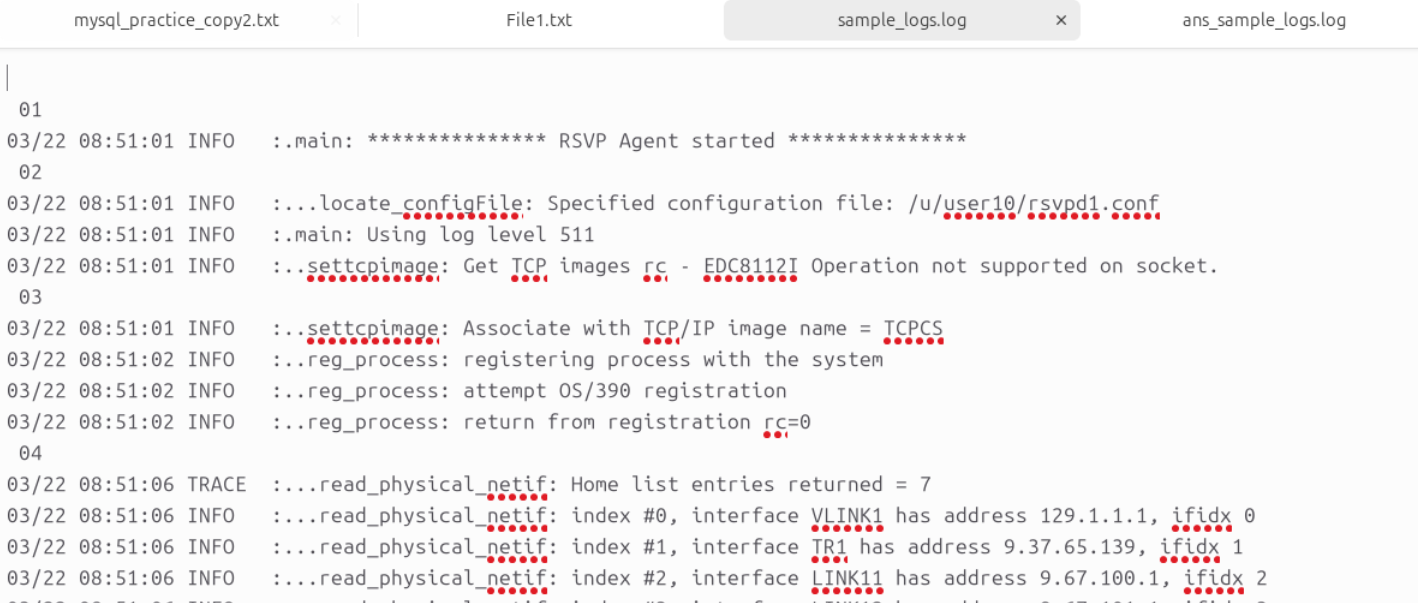
# Replace words from one file and generate Ans:



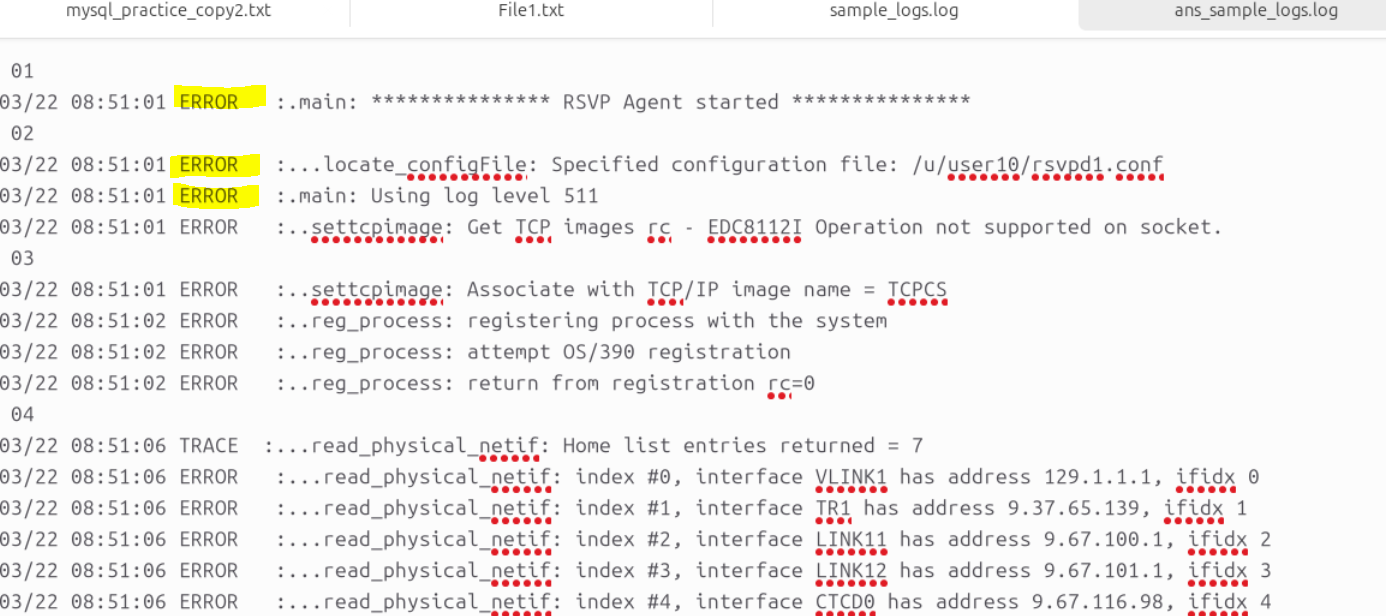




* Sample\_log file



* Output File ans\_sample\_logs



Important Note:

* Ref: <https://www.ibm.com/docs/en/zos/2.4.0?topic=problems-example-log-file> for Sample\_log file.
* Snipping tool for cropping, copying and taking ss of outputs and codes