**Explanation Of split char alternatively in Two Different file**

#!/bin/bash

This line specifies that the script should be run using the Bash shell.

# Prompt the user to enter the file path

read -p "Enter the file path: " file\_path

This line prompts the user to enter the file path and stores it in the variable file\_path read is basically read input from user

# Check if the file exists

if [ ! -f "$file\_path" ]; then

echo "File not found."

exit 1

Fi

This block of code is conditional statements checks if the file specified by file\_path exists. If the file does not exist, it prints an error message and exits the script with a status code of 1

# Check if the file contains a newline character

if grep -qF $'\n' "$file\_path"; then

echo -e "\n"

else

echo ""

Fi

This code block uses the grep command to check if the file specified by file\_path contains a newline character. If it does, it prints a newline character. Otherwise, it prints an empty string.

# Split the file into two parts

filename=$(basename "$file\_path")

extension="${filename##\*.}"

filename="${filename%.\*}"

file1="${filename}\_part1.${extension}"

file2="${filename}\_part2.${extension}"

This block of code extracts the filename and extension from the file\_path variable. It then creates two new filenames in which splitted character (file1 and file2) by appending "\_part1" and "\_part2" to the base filename, respectively.

# Read the file content

content=$(cat "$file\_path")

This line uses the cat command to read the content of the file specified by file\_path and stores it in the content variable.

# Split the content into two parts

length=${#content}

part1=""

part2=""

for ((i=0; i<$length; i++)); do

char="${content:i:1}"

if grep -qF $'\n' "$file\_path"; then

echo -e "\n"

else

echo ""

fi

if (( i % 2 == 0 )); then

part1+="$char"

else

part2+="$char"

fi

Done

This block of code iterates over each character in the content variable (under a loop there is a conditional statement is performe). It splits the content into two parts (part1 and part2) based on the position of the character. Every even-indexed character is appended to part1, and every odd-indexed character is appended to part2. The if statement inside the loop checks if the file contains a newline character and prints a newline accordingly.

# Write the parts to separate files

echo "$part1" > "$file1"

echo "$part2" > "$file2"

These lines write the contents of part1 and part2 to separate files (file1 and file2, respectively) using the echo command.

echo "File has been split into two parts."

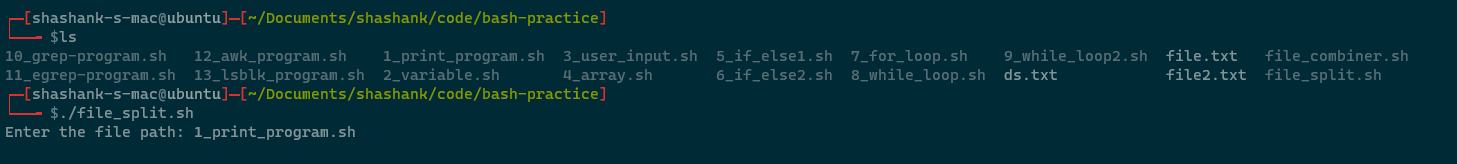
echo "Part 1: $file1"

echo "Part 2: $file2"

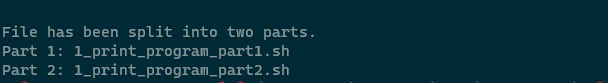
These line of code will print the na,e of output file\_name that is generated while the whole thing perform.

Result I achive from this

1.Run this program



Output of this code



Result of that output  
  
