**Create the Bash script file:** First, create a new file named log-monitor.sh.

**Implement log file monitoring**: Use the tail command to continuously monitor the specified log file for new entries.

**Implement stopping mechanism**: Implement a way to stop the monitoring loop, such as catching the Ctrl+C signal.

**Implement basic log analysis:** Count occurrences of specific keywords or patterns in the log file.

**Generate summary reports:** Generate summary reports based on the analysis.

**Add error handling and logging:** Include error handling to provide feedback on script execution.

**Write README.**md: Provide instructions on how to use and test the script.

By creating the **log-monitor.sh** script file. You can use a text editor like Nano or vim to create and edit the file.

**nano log-monitor.sh**

Once you have the file open in the text editor, let's start writing the script. We'll start with implementing log file monitoring using the tail command. Below is the initial script:

**#!/bin/bash**

**# Function to monitor log file**

**monitor\_log() {**

**# Specify the log file to monitor**

**logfile="/path/to/your/logfile.log"**

**# Start monitoring the log file**

**tail -f "$logfile"**

**}**

**monitor\_log**

This script will continuously monitor the specified log file and display new log entries in real time. However, it doesn't include a mechanism to stop the monitoring loop yet.

To implement a mechanism to stop the monitoring loop, we can catch the Ctrl+C signal (SIGINT) using a trap.

**#!/bin/bash**

**# Function to monitor log file**

**monitor\_log() {**

**# Specify the log file to monitor**

**logfile="/path/to/your/logfile.log"**

**# Trap Ctrl+C to stop monitoring**

**trap 'echo "Monitoring interrupted. Exiting."; exit' INT**

**# Start monitoring the log file**

**tail -f "$logfile"**

**}**

**# Call the function to start monitoring the log file**

**monitor\_log**

To add basic log analysis functionalities, we can enhance the script to perform simple analysis on the log entries. Let's start by counting the occurrences of specific keywords or patterns in the log file.

**#!/bin/bash**

**# Function to monitor log file**

**monitor\_log() {**

**# Specify the log file to monitor**

**logfile="/path/to/your/logfile.log"**

**# Trap Ctrl+C to stop monitoring**

**trap 'echo "Monitoring interrupted. Exiting."; exit' INT**

**# Start monitoring the log file**

**tail -f "$logfile" | while read line; do**

**# Perform log analysis (count occurrences of specific keywords or patterns)**

**# Example: Count occurrences of "ERROR" keyword**

**error\_count=$(echo "$line" | grep -o "ERROR" | wc -l)**

**# Print the log line and the analysis result**

**echo "$line"**

**echo "Error count: $error\_count"**

**done**

**}**

**# Call the function to start monitoring the log file**

**monitor\_log**

Let's enhance the script further by adding the ability to generate summary reports based on the analysis. We can create a function to generate a summary report, which can display information such as the top error messages or other relevant statistics.

**#!/bin/bash**

**# Function to monitor log file**

**monitor\_log() {**

**# Specify the log file to monitor**

**logfile="/path/to/your/logfile.log"**

**# Trap Ctrl+C to stop monitoring**

**trap 'echo "Monitoring interrupted. Exiting."; exit' INT**

**# Start monitoring the log file**

**tail -f "$logfile" | while read line; do**

**# Perform log analysis (count occurrences of specific keywords or patterns)**

**# Example: Count occurrences of "ERROR" keyword**

**error\_count=$(echo "$line" | grep -o "ERROR" | wc -l)**

**# Print the log line and the analysis result**

**echo "$line"**

**echo "Error count: $error\_count"**

**# Append analysis result to summary report**

**echo "Error count: $error\_count" >> summary\_report.txt**

**done**

**}**

**# Function to generate summary report**

**generate\_summary\_report() {**

**# Generate summary report based on analysis results**

**echo "Summary Report:" >> summary\_report.txt**

**echo "----------------" >> summary\_report.txt**

**# Add more summary information as needed**

**}**

**# Call the function to start monitoring the log file**

**monitor\_log**

**# Call the function to generate summary report**

**generate\_summary\_report**