I used chmod +x q1.sh to compile the code and then used bash to execute the code.

Below is shown the output of the code.

A screen shot of a computer program

Description automatically generated

These alerts are also being stored in the Arham.log file.

alert() #just printing alert here

{

message="Alert!!"

echo "$message" | tee -a "$log\_file" # i used tee -a so that the terminal also shows the echo statements and then they are aslo written in the log file named arham.log

}

**\*\*\* the above is the alert function which simply prints alert! On the terminal and writes it down to the log file**

**# setting the threshold values below**

disk\_threshold=30

cpu\_threshold=30

memory\_threshold=10

log\_file="/home/arham/Desktop/OS thoery/Assignment # 1/arham.log"

max\_log\_size=10M

while true; do

**# data is stored in the file after each print in terminal**

**# Disk usage calculations and then writing alert call**

disk\_usage=$(df -h --output=pcent / | sed '1d' | awk '{printf "%.2f", $1}') **#df displays disk usage and used output=pcent to output the % only**

if (( $(echo "$disk\_usage > $disk\_threshold" | bc -l) )); then **# checking the threshold using if , if is doing the same thing throughout the code**

alert "Disk usage exceeds $disk\_threshold%: $disk\_usage"

echo "Disk usage value: $disk\_usage" | tee -a "$log\_file"

echo "Disk usage exceeded" | tee -a "$log\_file"

fi

**# CPU Usage Monitoring and then calling alert**

cpu\_usage=$(top -bn1 | grep "Cpu(s)" | sed "s/.\*, \*\([0-9.]\*\)%\* id.\*/\1/")

if (( $(echo "$cpu\_usage > $cpu\_threshold" | bc -l) )); then

alert "CPU usage exceeds $cpu\_threshold%: $cpu\_usage%"

echo "CPU usage value: $cpu\_usage" | tee -a "$log\_file"

echo "CPU usage exceeded" | tee -a "$log\_file"

fi

**# Memory Usage Monitoring and then alert is called if it is below threshold, in the above we were checking if it was above threshold**

free\_mem=$(free --mega | awk '/Mem/{print $4}') **# free mega displays the free memory**

total\_mem=$(free --mega | awk '/Mem/{print $2}')

memory\_usage=$(echo "scale=2; ($total\_mem - $free\_mem) \* 100 / $total\_mem" | bc)

if (( $(echo "$memory\_usage < $memory\_threshold" | bc -l) )); then

alert "Memory usage falls below $memory\_threshold%: $memory\_usage%"

echo "Memory usage value: $memory\_usage" | tee -a "$log\_file"

echo "Memory usage exceeded" | tee -a "$log\_file"

fi

**# Log Rotation means that new file is made when old one is filled up to 10mb**

if [ -f "$log\_file" ]; then # -f tells us if the path to the file is the one we need or not

log\_size=$(du -b "$log\_file" | awk '{print $1}')

if [ "$log\_size" -gt "$(($(echo "$max\_log\_size" | tr -d '[:alpha:]') \* 1024))" ]; then

mv "$log\_file" "$log\_file.old"

touch "$log\_file"

alert "Log file rotated: $log\_file"

fi

fi

sleep 10s

done