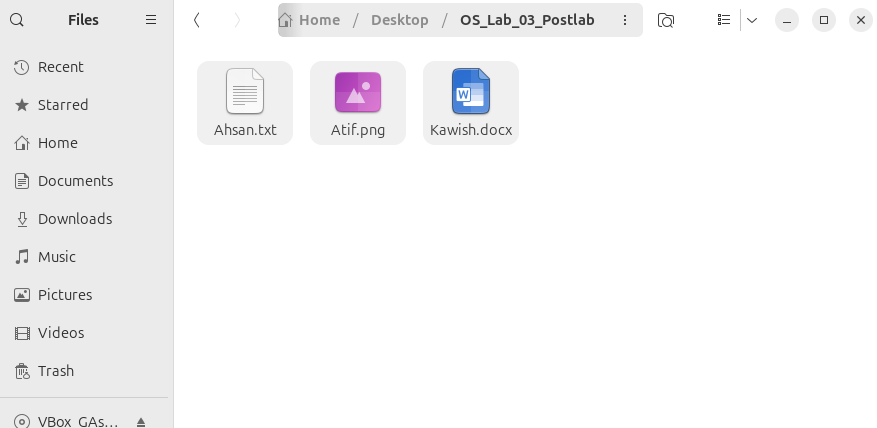
**OS LAB 03**

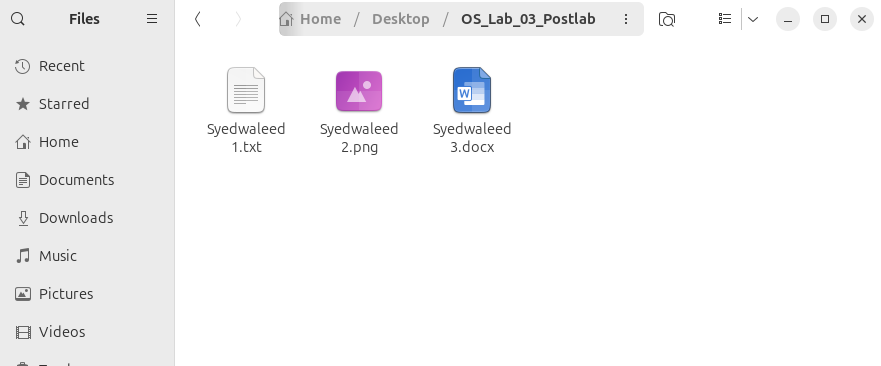
**Shell Scripting in Linux**

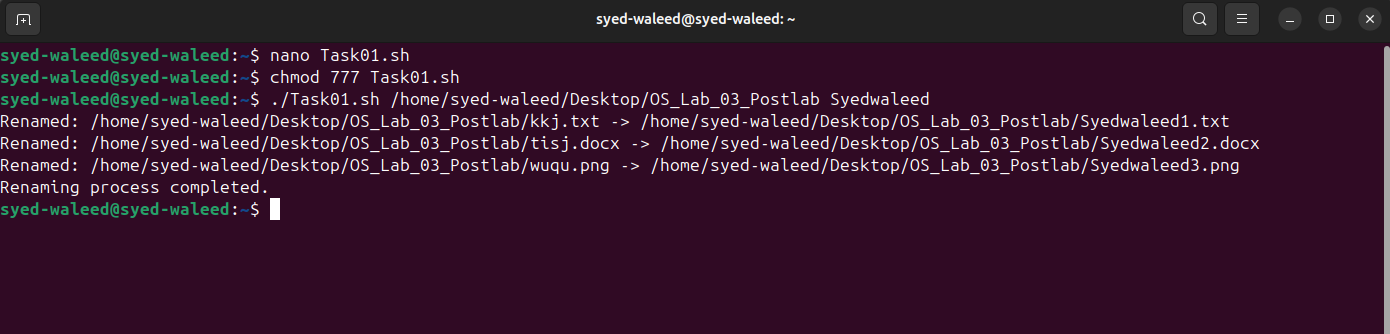
**Taskno\_01:**

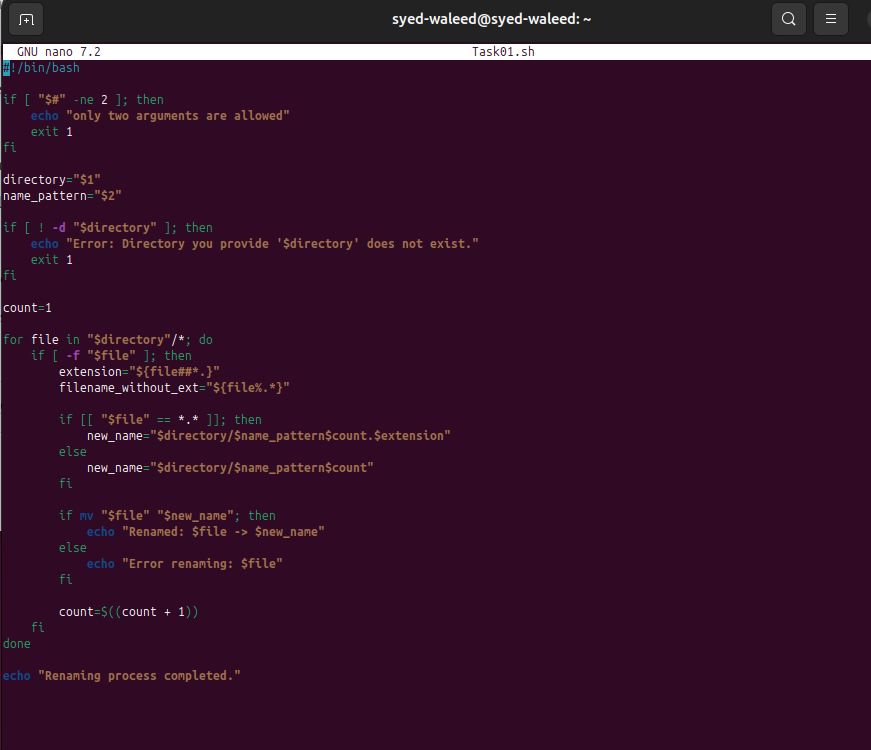
**Before Changing name :**



**After changing name:**







**Code:**

#!/bin/bash

if [ "$#" -ne 2 ]; then

echo "only two arguments are allowed"

exit 1

fi

directory="$1"

name\_pattern="$2"

if [ ! -d "$directory" ]; then

echo "Error: Directory you provide '$directory' does not exist."

exit 1

fi

count=1

for file in "$directory"/\*; do

if [ -f "$file" ]; then

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

filename\_without\_ext="${file%.\*}"

if [[ "$file" == \*.\* ]]; then

new\_name="$directory/$name\_pattern$count.$extension"

else

new\_name="$directory/$name\_pattern$count"

fi

if mv "$file" "$new\_name"; then

echo "Renamed: $file -> $new\_name"

else

echo "Error renaming: $file"

fi

count=$((count + 1))

fi

done

**Explanation:**

>if [ "$#" -ne 2 ]; to check only two arguments are input

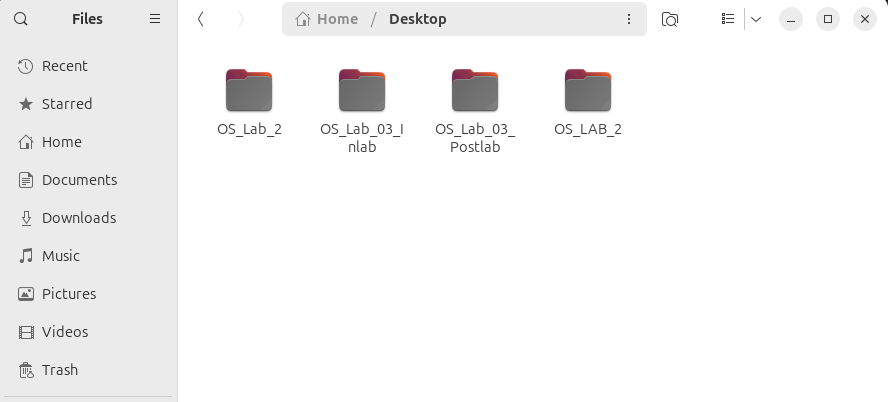
>directory="$1" name\_pattern="$2" to save arguments

>for file in "$directory"/\*; for loop to rename all filesin the directory

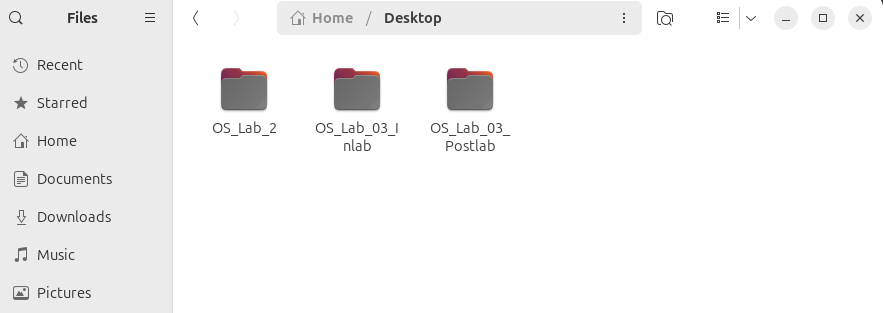
>extension="${file##\*.}" filename\_without\_ext="${file%.\*}" to save extentions and file names separately in different variables

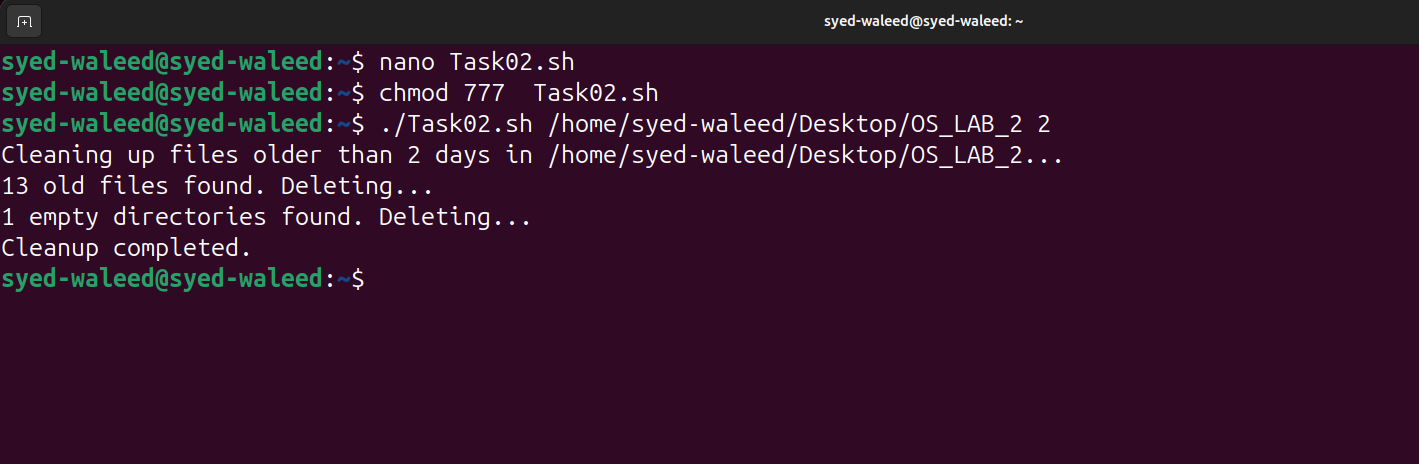
**Taskno\_02:**

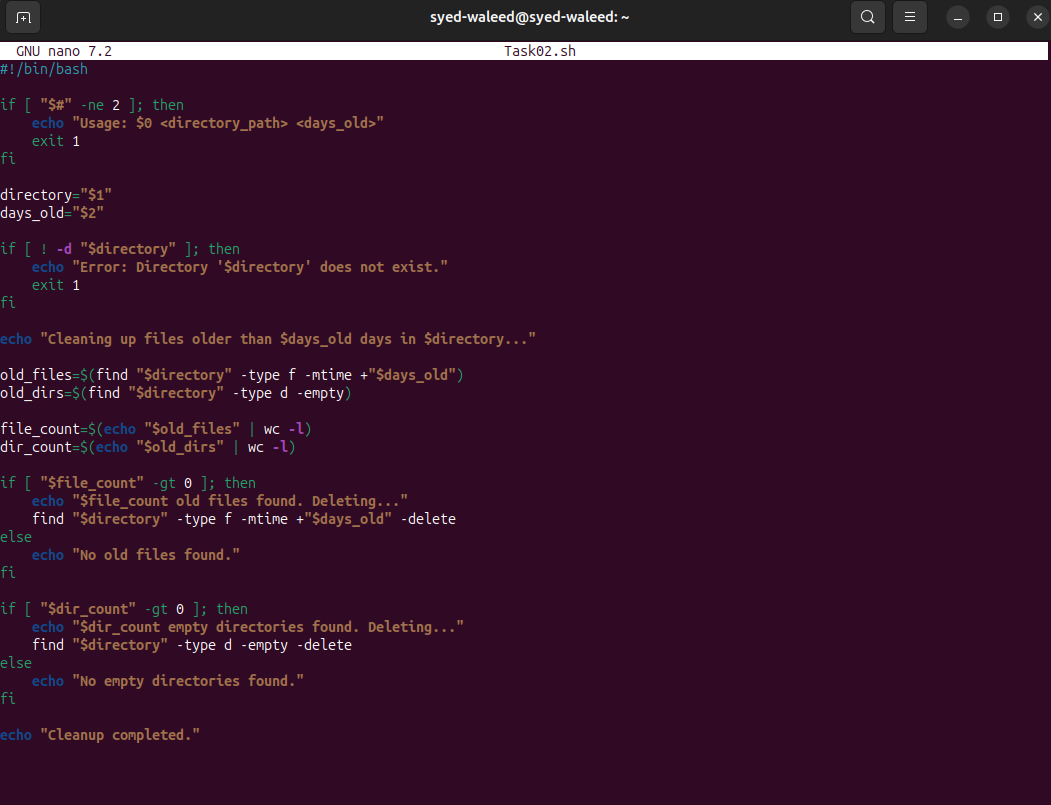
**Before Deleting**

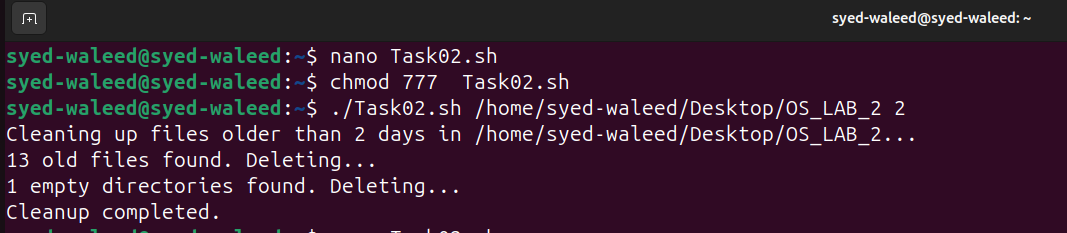


**After Deleting:**









**Code:**

#!/bin/bash

if [ "$#" -ne 2 ]; then

echo "Usage: $0 <directory\_path> <days\_old>"

exit 1

fi

directory="$1"

days\_old="$2"

if [ ! -d "$directory" ]; then

echo "Error: Directory '$directory' does not exist."

exit 1

fi

echo "Cleaning up files older than $days\_old days in $directory..."

old\_files=$(find "$directory" -type f -mtime +"$days\_old")

old\_dirs=$(find "$directory" -type d -empty)

file\_count=$(echo "$old\_files" | wc -l)

dir\_count=$(echo "$old\_dirs" | wc -l)

if [ "$file\_count" -gt 0 ]; then

echo "$file\_count old files found. Deleting..."

find "$directory" -type f -mtime +"$days\_old" -delete

else

echo "No old files found."

fi

if [ "$dir\_count" -gt 0 ]; then

echo "$dir\_count empty directories found. Deleting..."

find "$directory" -type d -empty -delete

else

echo "No empty directories found."

fi

echo "Cleanup completed."

**Explanation:**

>if [ "$#" -ne 2 ]; To check only 2 Arguments as input

>if [ ! -d "$directory" ]; To check Directory exist or not

>old\_files=$(find "$directory" -type f -mtime +"$days\_old")

Used find to search in directory, -type f means we are searching for file type and –mtime +$days\_old will search files given days old and this all will save in old\_files

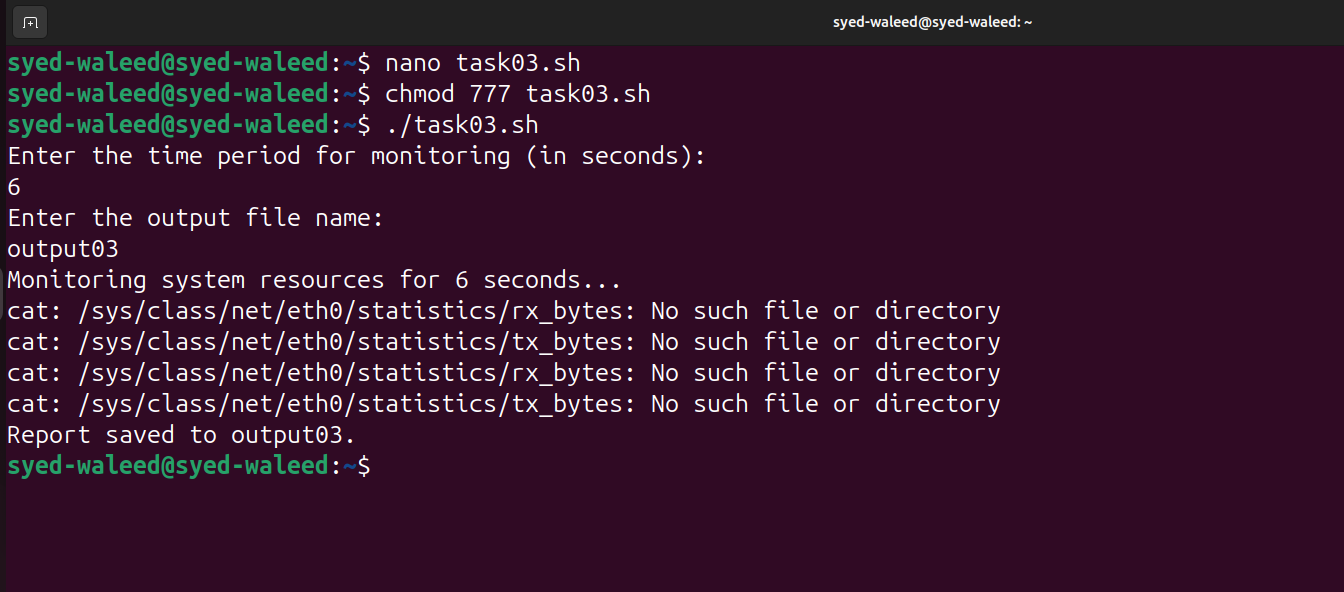
>file\_count=$(echo "$old\_files" | wc -l)

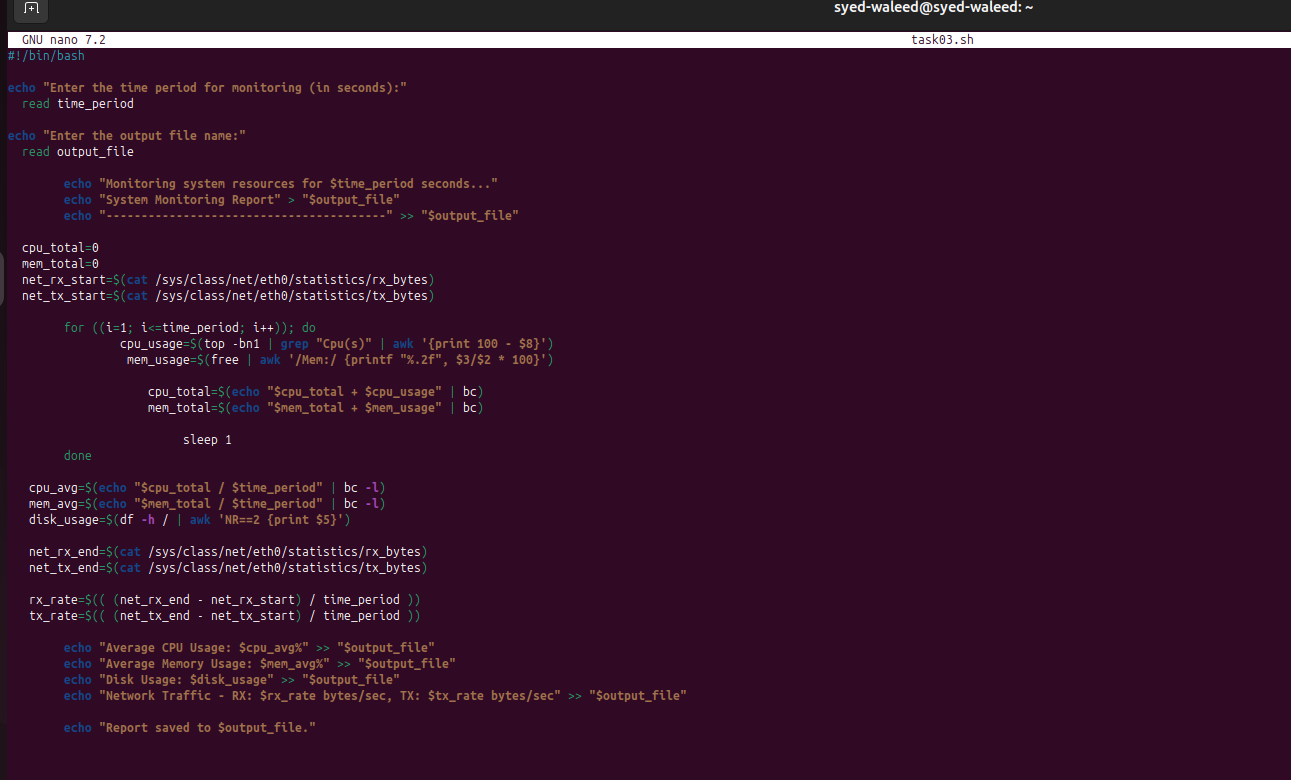
dir\_count=$(echo "$old\_dirs" | wc -l)

this will count how many directories and files are there

>-delete to delete

**Taskno\_03:**







**Code:**

#!/bin/bash

echo "Enter the time period for monitoring (in seconds):"

read time\_period

echo "Enter the output file name:"

read output\_file

echo "Monitoring system resources for $time\_period seconds..."

echo "System Monitoring Report" > "$output\_file"

echo "----------------------------------------" >> "$output\_file"

cpu\_total=0

mem\_total=0

net\_rx\_start=$(cat /sys/class/net/eth0/statistics/rx\_bytes)

net\_tx\_start=$(cat /sys/class/net/eth0/statistics/tx\_bytes)

for ((i=1; i<=time\_period; i++)); do

cpu\_usage=$(top -bn1 | grep "Cpu(s)" | awk '{print 100 - $8}')

mem\_usage=$(free | awk '/Mem:/ {printf "%.2f", $3/$2 \* 100}')

cpu\_total=$(echo "$cpu\_total + $cpu\_usage" | bc)

mem\_total=$(echo "$mem\_total + $mem\_usage" | bc)

sleep 1

done

cpu\_avg=$(echo "$cpu\_total / $time\_period" | bc -l)

mem\_avg=$(echo "$mem\_total / $time\_period" | bc -l)

disk\_usage=$(df -h / | awk 'NR==2 {print $5}')

net\_rx\_end=$(cat /sys/class/net/eth0/statistics/rx\_bytes)

net\_tx\_end=$(cat /sys/class/net/eth0/statistics/tx\_bytes)

rx\_rate=$(( (net\_rx\_end - net\_rx\_start) / time\_period ))

tx\_rate=$(( (net\_tx\_end - net\_tx\_start) / time\_period ))

echo "Average CPU Usage: $cpu\_avg%" >> "$output\_file"

echo "Average Memory Usage: $mem\_avg%" >> "$output\_file"

echo "Disk Usage: $disk\_usage" >> "$output\_file"

echo "Network Traffic - RX: $rx\_rate bytes/sec, TX: $tx\_rate bytes/sec" >> "$output\_file"

echo "Report saved to $output\_file."

**Explanation:**

>’>’ to create output file in overwritemode

>net\_rx\_start=$(cat /sys/class/net/eth0/statistics/rx\_bytes) net\_tx\_start=$(cat /sys/class/net/eth0/statistics/tx\_bytes) to compute network speed

>cpu\_usage=$(top -bn1 | grep "Cpu(s)" | awk '{print 100 - $8}') it Extracts the current CPU usage by subtracting the idle percentage ($8 field) from 100 using the top command.

> mem\_usage=$(free | awk '/Mem:/ {printf "%.2f", $3/$2 \* 100}') here free command to get memory usage and $3/$2 \* 100 gives the percentage of usage

>cpu\_total=$(echo "$cpu\_total + $cpu\_usage" | bc) mem\_total=$(echo "$mem\_total + $mem\_usage" | bc) to sum

>bc to calculate the floating nmbrs

-----------------------------------------------------------------------------------------------------------------------------------------