#vi /opt/scripts/system-info.sh

#!/bin/bash

echo -e "-------------------------------System Information----------------------------"

echo -e "Hostname:\t\t"`hostname`

echo -e "uptime:\t\t\t"`uptime | awk '{print $3,$4}' | sed 's/,//'`

echo -e "Manufacturer:\t\t"`cat /sys/class/dmi/id/chassis\_vendor`

echo -e "Product Name:\t\t"`cat /sys/class/dmi/id/product\_name`

echo -e "Version:\t\t"`cat /sys/class/dmi/id/product\_version`

echo -e "Serial Number:\t\t"`cat /sys/class/dmi/id/product\_serial`

echo -e "Machine Type:\t\t"`vserver=$(lscpu | grep Hypervisor | wc -l); if [ $vserver -gt 0 ]; then echo "VM"; else echo "Physical"; fi`

echo -e "Operating System:\t"`hostnamectl | grep "Operating System" | cut -d ' ' -f5-`

echo -e "Kernel:\t\t\t"`uname -r`

echo -e "Architecture:\t\t"`arch`

echo -e "Processor Name:\t\t"`awk -F':' '/^model name/ {print $2}' /proc/cpuinfo | uniq | sed -e 's/^[ \t]\*//'`

echo -e "Active User:\t\t"`w | cut -d ' ' -f1 | grep -v USER | xargs -n1`

echo -e "System Main IP:\t\t"`hostname -I`

echo ""

echo -e "-------------------------------CPU/Memory Usage------------------------------"

echo -e "Memory Usage:\t"`free | awk '/Mem/{printf("%.2f%"), $3/$2\*100}'`

echo -e "Swap Usage:\t"`free | awk '/Swap/{printf("%.2f%"), $3/$2\*100}'`

echo -e "CPU Usage:\t"`cat /proc/stat | awk '/cpu/{printf("%.2f%\n"), ($2+$4)\*100/($2+$4+$5)}' | awk '{print $0}' | head -1`

echo ""

echo -e "-------------------------------Disk Usage >80%-------------------------------"

df -Ph | sed s/%//g | awk '{ if($5 > 80) print $0;}'

echo ""

echo -e "-------------------------------For WWN Details-------------------------------"

vserver=$(lscpu | grep Hypervisor | wc -l)

if [ $vserver -gt 0 ]

then

echo "$(hostname) is a VM"

else

cat /sys/class/fc\_host/host?/port\_name

fi

echo ""

echo -e "-------------------------------Oracle DB Instances---------------------------"

if id oracle >/dev/null 2>&1; then

/bin/ps -ef|grep pmon

then

else

echo "oracle user does not exist on $(hostname)"

fi

echo ""

if (( $(cat /etc/\*-release | grep -w "Oracle|Red Hat|CentOS|Fedora" | wc -l) > 0 ))

then

echo -e "-------------------------------Package Updates-------------------------------"

yum updateinfo summary | grep 'Security|Bugfix|Enhancement'

echo -e "-----------------------------------------------------------------------------"

else

echo -e "-------------------------------Package Updates-------------------------------"

cat /var/lib/update-notifier/updates-available

echo -e "-----------------------------------------------------------------------------"

fi

output

-------------------------------System Information---------------------------

Hostname: daygeek-Y700

uptime: 1:20 1

Manufacturer: LENOVO

Product Name: 80NV

Version: Lenovo ideapad Y700-15ISK

Serial Number: AA0CMRN1

Machine Type: Physical

Operating System: Manjaro Linux

Kernel: 4.19.80-1-MANJARO

Architecture: x86\_64

Processor Name: Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz

Active User: daygeek renu thanu

System Main IP: 192.168.1.6 192.168.122.1

-------------------------------CPU/Memory Usage------------------------------

Memory Usage: 37.28%

Swap Usage: 0.00%

CPU Usage: 15.43%

-------------------------------Disk Usage >80%-------------------------------

Filesystem Size Used Avail Use Mounted on

/dev/nvme0n1p1 217G 202G 4.6G 98 /

/dev/loop0 109M 109M 0 100 /var/lib/snapd/snap/odrive-unofficial/2

/dev/loop1 91M 91M 0 100 /var/lib/snapd/snap/core/6405

/dev/loop2 90M 90M 0 100 /var/lib/snapd/snap/core/7713

-------------------------------For WWN Details-------------------------------

CentOS8.2daygeek.com is a VM

-------------------------------Oracle DB Instances---------------------------

oracle user does not exist on CentOS8.2daygeek.com

-------------------------------Package Updates-------------------------------

13 Security notice(s)

9 Important Security notice(s)

3 Moderate Security notice(s)

1 Low Security notice(s)

35 Bugfix notice(s)

1 Enhancement notice(s)

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

now=$(date)

echo "$now"  
echo "Current date: $now"

$ hs=`hostname`

$ echo $hs

#!/bin/bash

# function to show memory usage

memoryUsage(){

echo "Memory Usage:"

free

read -p "Press any key to Continue...."

}

# function to show disk usage

diskUsage(){

echo "Disk Usage:"

df

read -p "Press any key to Continue...."

}

# function to show menu

show\_menu()

{

clear

echo "++++++++++++ MENU +++++++++++++"

echo "1. Show Memory Usage."

echo "2. Show DIsk Usage."

echo "3. Exit"

echo "+++++++++++++++++++++++++++++++"

}

# function to take input

take\_input()

{

#take the input and store it in choice variable

local choice

read -p "Select the option from above menu: " choice

#using switch case statement check the choice and call function.

case $choice in

1) memoryUsage ;;

2) diskUsage ;;

3) exit 0;;

\*) echo "Enter Valid Option!!"

read -p "Press any key to Continue...."

esac

}

# for loop to call the show\_menu and take\_input function.

while true

do

show\_menu

take\_input

done

# vi /opt/scripts/system-uptime-script.sh

#!/bin/bash

upserver=$(for server in `cat /tmp/servers.txt`

do nc -zvw3 $server 22 | awk '/succeeded/ {print $3}'

done 2>/dev/null)

for host in $upserver

do

echo -n "$host: "

ssh $host uptime | awk '{print $3,$4}' | sed 's/,//'

done | column -t > /tmp/uptime-report.out

cat /tmp/uptime-report.out | mail -s "Linux Servers Uptime Report" "daygeek@gmail.com"

#-q option is to count the number of users and print the logged-in users.

# instead of -q, --count can also be used.

# -v is used to exclude any pattern

who -q | grep -v users

;;

Right angle triangle

#! /bin/ bash

for ((i=1;i<= 9;i++))

do

for ((j=1;j<=$i;j++ ))

do

echo -n "\*"

done

echo

done

left angle triangle

# Program to print the

# given pattern

# Static input for N

N=5

# variable used for

# while loop

i=0

j=0

while [ $i -le `expr $N - 1` ]

do

    j=0

    while [ $j -le `expr $N - 1` ]

    do

        if [ `expr $N - 1` -le `expr $i + $j` ]

        then

          # Print the pattern

          echo -ne "#"

        else

          # Print the spaces required

          echo -ne " "

        fi

        j=`expr $j + 1`

    done

    # For next line

    echo

    i=`expr $i + 1`

done

diamond

i=1

2: while [ $i -le 5 ]

3: do

4: j=5

5: while [ $j -ge $i ]

6: do

7: echo -n " "

8: j=`expr $j - 1`

9: done

10: j=1

11: while [ $j -le $i ]

12: do

13: echo -n "\*"

14: echo -n " "

15: j=`expr $j + 1`

16: done

17: echo " "

18: i=`expr $i + 1`

19: done

20: i=1

21: while [ $i -le 5 ]

22: do

23: j=1

24: while [ $j -le $i ]

25: do

26: echo -n " "

27: j=`expr $j + 1`

28: done

29: j=5

30: while [ $j -ge $i ]

31: do

32: echo -n "\*"

33: echo -n " "

34: j=`expr $j - 1`

35: done

36: echo " "

37: i=`expr $i + 1`

38: done

equilateral triangle

For((I=1;I<=5;I++))  
do  
  for((j=5;j>I;j--))  
   do  
  echo -n " "  
   done  
   for((k=1;k<2\*I;k++))  
    do  
  echo -n "\*"  
     done  
     echo " "  
done

Square rectangle

echo "Size of the square?"

read size

clear

for (( i=0;i<size;i++ ))

do

echo "\*"

done