**Task 1:** Please design a script that writes the numbers from 1 - 10 in random order. Each number should appear only once. You can use bash only. Please provide tests for the script, along with documentation which should include the following:

* Build instructions
* Usage
* Description
* Known limitations / bugs

**Steps to execute the bash script of task1:**

1. Clone and download my repository into your local machine / or directly download the bash script executable file into your local machine from the given link:

<https://github.com/hari680/Task-Adjust.git>

1. Please find my bash executable file ‘task1randomnumbers.sh’
2. Open any command prompt🡪 for eg: Git Bash
3. Change directory and navigate to the path where you have placed the ‘task1randomnumbers.sh’ file
4. Provide executable permission to the file in order to make it executable:

# chmod u+x task1randomnumbers.sh

1. Run the executable script file

# ./task1randomnumbers.sh

1. Output appears as shown below:

10

4

2

8

5

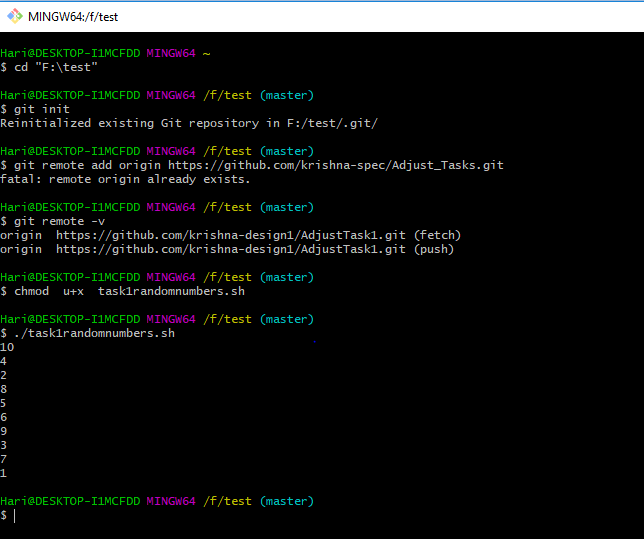
6

9

3

7

1

Screenshot for reference:

**Solution:**

1. Create a text file and write the script then save and exit:

#! /bin/bash

for (( i=1;i<=10;i++ )) do echo $RANDOM $i; done | sort -k1|cut -d " " -f2 | head -10

1. Output appears as shown below

10

4

2

8

5

6

9

3

7

1

**Known limitations / bugs:**

If there is no executable permission to file it cannot be executed. Hence use chmod command to get out of this problem.

**Task 2**: Imagine a server with the following specs:

* 4 times Intel(R) Xeon(R) CPU E7-4830 v4 @ 2.00GHz
* 64GB of ram
* 2 tb HDD disk space
* 2 x 10Gbit/s nics

The server is used for SSL offloading and proxies around 25000 requests per second. Please let us know which metrics are interesting to monitor in that specific case and how would you do that? What are the challenges of monitoring this?

1. First we need to setup the server with 4 times Intel(R) Xeon(R) CPU E7-4830 v4 @ 2.00GHz, 64GB of ram , 2 tb HDD disk space and 2 x 10Gbit/s nics.
2. Basically, It works this way, the **proxy** server you use for the **SSL offloading** acts as the **SSL** terminator, which also acts as an edge device. When a client attempts to connect to a website, the client connects to the **SSL** terminator—that connection is HTTPS.

**Metrics to monitor:**

**Server general metrics:**

* Sever running state and Load average of the server.
* User Utilization, System Utilization, i/o utilization.
* CPU

CPU used

CPU load

CPU temperature

* Memory usage
* I/O

Disk space usage and performance of disk

IO read, IO write

* Network (nics)

Network traffic

TCP connection states

**Application metrics :**

SSL certificate validate status

**How to monitor metrics?**

**Method 1: Monitoring with the monitoring tool or with command prompt**

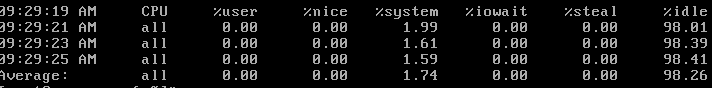
1. First I will know the server running state and load average of server i.e 1st min, 5th min and 15th min load by using **#uptime** command

#uptime



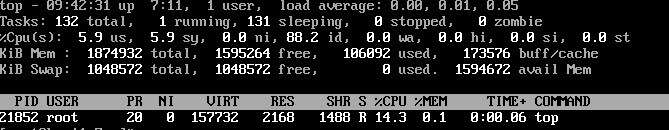
1. Next I will run the **#sar iterations interval** command to know the user utilization, system utilization and i/o utilization.

#sar 2 3



1. Next to identify CPU information I will run the **#cat /proc/cpuinfo or #top or #sar** **or #htop** commands

#top | head -8



1. I will identify the memory information i.e memory used and memory free by using following commands #free –m or #vmstat or #cat /proc/meminfo

#cat /proc/meminfo

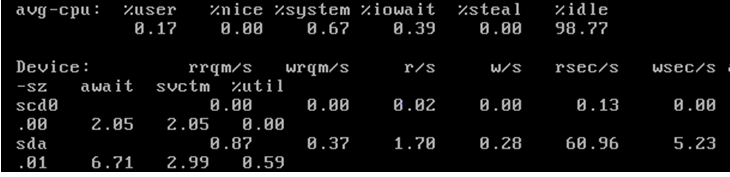


#free –m



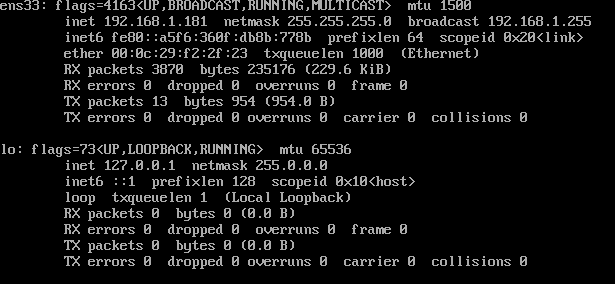
1. Next verify I/O operations of device and its performance i.e i/o read waiting and i/o write waiting percentages by using #iostat –xtc command. And df –h to know the disk space used.

#iostat –xtc



1. Next I will run #ifconfig –a command to identify all connected nic and its status i.e. whether online or not , up, broadcasting, observe TX and RX packets along with error and drop count.

#ifconfig –a



1. Verify the SSL certificate validate status

By using monitoring tool if the metric value is over limit field, it will trig the related trigger and it send an alert message through SMS or an email.

**Challenges of monitoring**

1. Figure out important metrics to monitor.
2. Observe the outputs of command carefully and take necessary steps.
3. Metrics monitoring tool should not affect the server's performance.
4. Set proper trigger warning value to minimize false alarm.
5. Monitoring the metrics with commands should not affect the other running services, process and jobs.