# Linux and Bash scripts Homework

**A. Create a script that uses the following keys:**

1. When starting without parameters, it will display a list of possible keys and their description.
2. The ***--all*** key displays the IP addresses and symbolic names of all hosts in the current subnet
3. The ***--target*** key displays a list of open system TCP ports.

The code that performs the functionality of each of the subtasks must be placed in a separate function

Script:

```
#!/bin/bash

#help function
function show_help(){
	echo "This script does next actions:"
	echo "--all			displays the IP addresses and symbolic names of all hosts in the current subnet."
	echo "--target		displays a list of open system TCP ports."
	echo "--help,nothing		show this help"
}

#
function subnet_hosts(){
	echo
	#get our ip address for 2 ways:
	#myIP=`ip route get 8.8.8.8 | head -1 | cut -d' ' -f7`
	myIP=`hostname -I | cut -d' ' -f1`
	echo $myIP
}

#
function show_open_TCP_ports(){
	echo "Scan localhost for open TCP ports:"
	nmap localhost -p1-65535 | sed -n '/PORT/,/Nmap/p' | sed '$d'
}

#check number of args
if [ $# -eq 0 ]; then
	show_help
	exit 0
fi

#select key handler
case $1 in
	"--help"|"-h")
		show_help
	;;
	"--all")
		subnet_hosts
	;;
	"--target")
		show_open_TCP_ports
	;;

	*)
		echo "Error: key unknown, use --help for list of enabled keys"
	;;
esac
```

Results:

```
[rvv@CentOs8 L0-Linux-Bash]$ ./task-a.sh
This script does next actions:
--all           displays the IP addresses and symbolic names of all hosts in the current subnet.
--target        displays a list of open system TCP ports.
--help,nothing  show this help
[rvv@CentOs8 L0-Linux-Bash]$ ./task-a.sh --target
Scan localhost for open TCP ports:
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
631/tcp  open  ipp
6010/tcp open  x11

[rvv@CentOs8 L0-Linux-Bash]$ ./task-a.sh --help
This script does next actions:
--all           displays the IP addresses and symbolic names of all hosts in the current subnet.
--target        displays a list of open system TCP ports.
--help,nothing  show this help
[rvv@CentOs8 L0-Linux-Bash]$
```

**B. Using Apache log example create a script to answer the following questions:**

1. From which ip were the most requests?
2. What is the most requested page?
3. How many requests were there from each ip?
4. What non-existent pages were clients referred to?
5. What time did site get the most requests?
6. What search bots have accessed the site? (UA + IP)

Script:

```
#!/bin/bash

function favoritIP(){
	cut $source -f 1 -d " " | sort |uniq -c | sort -r -n | head -n1
}

function mostReqPage(){
	cat $source | sed -n 's_.*\(http://[^/]*\).*_\1_p' | sort |uniq -c | sort -r -n | head -n1
}

function mostReqByIP(){
	cut $source -f 1 -d " " | sort |uniq -c | sort -r -n
}

function badReq(){
	cat $source | grep 'HTTP/1.0" 404 ' | wc -l
}

function mostReqByTime(){
	echo "--- By hours top 5:"
	cat $source | sed -n 's_.*\[\(.*:[0-2][0-9]\):[0-5][0-9]:[0-5][0-9].*\(http://[^/]*\).*_\1 \2_p' |sort | uniq -c | sort -n -r -k 1 | head -n5
	echo "--- By minutes top 5:"
	cat $source | sed -n 's_.*\[\(.*:[0-2][0-9]:[0-5][0-9]\):[0-5][0-9].*\(http://[^/]*\).*_\1 \2_p' |sort | uniq -c | sort -n -r -k 1 | head -n5
}

function UAIP(){
	cat $source |  sed -n 's_^\(.*\) - -.* "-" "\(.*\)(.*;\(.*[Bb]ot.*\);.*_\1\t\2\t\3_p' | sort -k 1 | uniq | sort -k 2 #work 2

	#this commnd with less conditions
	#cat example_log.log |  sed -n 's_^\(.*\) - -.*;\(.*[Bb]ot.*\);.*_\1\t\2_p' | sort -k 1 | uniq | sort -k 2 #work fine
}
if [ $# -eq 0 ]
	then
	echo "Must be one argument - source filename!"
	exit 0
fi
source=$1
if [ $# -eq 2 ]
	then
	dest=$2
else
	dest="/dev/stdout"
fi

echo "Please select quastion for solve:"
echo "1. From which ip were the most requests?"
echo "2. What is the most requested page?"
echo "3. How many requests were there from each ip?"
echo "4. What non-existent pages were clients referred to?"
echo "5. What time did site get the most requests?"
echo "6. What search bots have accessed the site? (UA + IP)"

echo "q - for Quit"
exec 6<&1
while : ; do
  exec 1<&6
  echo -n "Your choise:"
  read UserSelect
  exec 1>>$dest
  case $UserSelect in
	1) echo "Calculete and find IP witch most requests"
	  favoritIP
	  ;;
	2) echo "Most requests page is:"
	  mostReqPage
	  ;;
	3) echo "Requests by IP:"
	  mostReqByIP
	  ;;
	4) echo -n "Non-existent pages were clients referred is "
	  badReq
	  ;;
	5) echo "Time when site get the most requests"
	  mostReqByTime
```

```
            ;;
        6) echo "Search bots who have accessed the site (UA + IP)"
            UAIP
            ;;
        [qQ]) exit 0
    ;;
        *) echo "Invalid input, Please select 1-6 or "q" for exit"
            ;;
  esac
done
```

Results:

```
[rvv@CentOs8 L0-Linux-Bash]$ ./task-b.sh example_log.log result.txt
Please select quastion for solve:
1. From which ip were the most requests?
2. What is the most requested page?
3. How many requests were there from each ip?
4. What non-existent pages were clients referred to?
5. What time did site get the most requests?
6. What search bots have accessed the site? (UA + IP)
q - for Quit
Your choise:1
Your choise:2
Your choise:3
Your choise:4
Your choise:5
Your choise:6
Your choise:q
[rvv@CentOs8 L0-Linux-Bash]$
```

Task b : 1-3

```
[rvv@CentOs8 L0-Linux-Bash]$ ./task-b.sh example_log.log
Please select quastion for solve:
1. From which ip were the most requests?
2. What is the most requested page?
3. How many requests were there from each ip?
4. What non-existent pages were clients referred to?
5. What time did site get the most requests?
6. What search bots have accessed the site? (UA + IP)
q - for Quit
Your choise:1
Calculete and find IP witch most requests
     29 94.78.95.154
Your choise:2
Most requests page is:
    297 http://example.com
Your choise:3
Requests by IP:
     29 94.78.95.154
     21 95.31.14.165
     19 176.108.5.105
     16 31.7.230.210
     14 144.76.76.115
     12 217.69.133.239
     11 66.102.9.35
     11 5.255.251.28
     11 217.69.133.234
     11 188.123.232.29
     10 91.121.209.185
     10 46.158.68.55
      9 93.170.253.156
      9 5.135.154.105
      9 217.69.133.236
      8 91.206.110.87
      8 82.193.140.164
      8 66.102.9.32
      8 217.69.133.235
      8 213.80.162.114
      8 195.24.255.94
      7 66.102.9.38
      7 31.173.84.130
      6 95.65.45.111
      6 66.249.66.204
```

Task b : 4-5

```
      1 109.126.143.53
      1 107.23.224.100
Your choise:4
Non-existent pages were clients referred is 31
Your choise:5
Time when site get the most requests
--- By hours top 5:
     90 25/Apr/2017:12 http://example.com
     62 25/Apr/2017:11 http://example.com
     60 25/Apr/2017:09 http://example.com
     59 25/Apr/2017:10 http://example.com
     31 25/Apr/2017:11 http://go.mail.ru
--- By minutes top 5:
      6 25/Apr/2017:13:02 http://example.com
      6 25/Apr/2017:11:36 http://go.mail.ru
      5 25/Apr/2017:10:02 http://example.com
      5 25/Apr/2017:09:16 http://example.com
      5 25/Apr/2017:09:02 http://example.com
Your choise:6
Search bots who have accessed the site (UA + IP)
164.132.161.40  Mozilla/5.0      AhrefsBot/5.2
164.132.161.63  Mozilla/5.0      AhrefsBot/5.2
164.132.161.85  Mozilla/5.0      AhrefsBot/5.2
217.182.132.183 Mozilla/5.0      AhrefsBot/5.2
136.243.34.71   Mozilla/5.0      bingbot/2.0
207.46.13.109   Mozilla/5.0      bingbot/2.0
207.46.13.128   Mozilla/5.0      bingbot/2.0
207.46.13.3     Mozilla/5.0      bingbot/2.0
40.77.167.19    Mozilla/5.0      bingbot/2.0
66.249.66.194   Mozilla/5.0      Googlebot/2.1
66.249.66.199   Mozilla/5.0      Googlebot/2.1
66.249.66.204   Mozilla/5.0      Googlebot/2.1
```

**C. Create a data backup script that takes the following data as parameters:**

1. Path to the syncing directory.
2. The path to the directory where the copies of the files will be stored.

In case of adding new or deleting old files, the script must add a corresponding entry to the log file indicating the time, type of operation and file name. [The command to run the script must be added to crontab with a run frequency of one minute]

Script:

```
#!/bin/bash

if [ $# -lt 2 ]
        then
        echo "Error: Must be minimum 2 arguments: task-c.sh SourceDir DestignationDir [logfile]"
        exit 1
fi

source=$1
dest=$2
logfile="./rsync.log"
if [ $# -ge 3 ]
        then
        logfile=$3
fi

if ! [ -d $source ]
        then
        echo "Error: Source dir does not exist!"
        exit 1
fi
if ! [ -d $dest ]
        then
        echo "Error: Destignation dir does not exist!"
        exit 1
fi

rsync -carv --delete --log-file=$logfile $source $dest
```

Results:

```
[rvv@CentOs8 L0-Linux-Bash]$
[rvv@CentOs8 L0-Linux-Bash]$ ./task-c.sh ~/sourcedir1 ~/destdir1 log-rsync1
sending incremental file list
sourcedir1/
sourcedir1/result.txt
sourcedir1/testfile3

sent 22,497 bytes  received 59 bytes  45,112.00 bytes/sec
total size is 831,549  speedup is 36.87
[rvv@CentOs8 L0-Linux-Bash]$
[rvv@CentOs8 sourcedir1]$ ./task-c.sh ~/sourcedir1 ~/destdir1 log-rsync1
bash: ./task-c.sh: No such file or directory
[rvv@CentOs8 sourcedir1]$
[rvv@CentOs8 L0-Linux-Bash]$ ./task-c.sh ~/sourcedir1 ~/destdir1 log-rsync1
sending incremental file list
deleting sourcedir1/result.txt
sourcedir1/

sent 374 bytes  received 46 bytes  840.00 bytes/sec
total size is 820,501  speedup is 1,953.57
[rvv@CentOs8 L0-Linux-Bash]$ ./task-c.sh ~/sourcedir1
Error: Must be minimum 2 arguments: task-c.sh SourceDir DestignationDir [logfile]
[rvv@CentOs8 L0-Linux-Bash]$ ./task-c.sh ~/sourcedir1 ~/destdir2
Error: Destignation dir does not exist!
[rvv@CentOs8 L0-Linux-Bash]$ ./task-c.sh ~/sourcedir2 ~/destdir1
Error: Source dir does not exist!
[rvv@CentOs8 L0-Linux-Bash]$ ./task-c.sh ~/sourcedir1 ~/destdir1
sending incremental file list

sent 371 bytes  received 18 bytes  778.00 bytes/sec
total size is 820,501  speedup is 2,109.26
[rvv@CentOs8 L0-Linux-Bash]$
```

```
[rvv@CentOs8 L0-Linux-Bash]$ ./task-c.sh ~/sourcedir1 ~/destdir1
sending incremental file list
sourcedir1/
sourcedir1/example_log.log
sourcedir1/result.txt
sourcedir1/testfile3
sourcedir1/screenshots/
sourcedir1/screenshots/task-a-2.jpg
sourcedir1/screenshots/task-b.jpg
sourcedir1/screenshots/task-b_1-3_result.jpg
sourcedir1/screenshots/task-b_4-6_result.jpg

sent 832,429 bytes  received 165 bytes  1,665,188.00 bytes/sec
total size is 831,536  speedup is 1.00
[rvv@CentOs8 L0-Linux-Bash]$
[rvv@CentOs8 L0-Linux-Bash]$ ./task-c.sh ~/sourcedir1 ~/destdir1
sending incremental file list

sent 414 bytes  received 18 bytes  864.00 bytes/sec
total size is 831,536  speedup is 1,924.85
[rvv@CentOs8 L0-Linux-Bash]$
[rvv@CentOs8 L0-Linux-Bash]$ ./task-c.sh ~/sourcedir1 ~/destdir1
sending incremental file list
sourcedir1/
sourcedir1/testfile3

sent 11,395 bytes  received 40 bytes  22,870.00 bytes/sec
total size is 831,536  speedup is 72.72
[rvv@CentOs8 L0-Linux-Bash]$
[rvv@CentOs8 L0-Linux-Bash]$ ./task-c.sh ~/sourcedir1 ~/destdir1
sending incremental file list
sourcedir1/

sent 375 bytes  received 21 bytes  792.00 bytes/sec
total size is 820,488  speedup is 2,071.94
[rvv@CentOs8 L0-Linux-Bash]$
[rvv@CentOs8 L0-Linux-Bash]$ ./task-c.sh ~/sourcedir1 ~/destdir1
sending incremental file list
deleting sourcedir1/result.txt

sent 372 bytes  received 43 bytes  830.00 bytes/sec
```

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name  command to be executed
0-59 * * * * rvv /home/rvv/git_projects/l1/L0-Linux-Bash/task-c.sh ~/sourcedir1 ~/destdir1 /home/rvv/git_projects/l1/L0-Linux-Bash/log-rsync1

```