Linux administration with bash. Home task

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

```
sdrv@U:~/SServe$ cat taskA.sh
#!/bin/bash

empty () {
echo  -e "--all key displays the IP addresses and symbolic names of all hosts in the current
 subnet\n--target key displays a list of open system TCP ports"
}

all () {
nmap -sn 10.10.10.* | grep for | tr -d 'Nmap scan report for'
}

target () {
nmap -sT localhost | grep open
}

if [ "$#" == "0" ]
then
 empty
 exit 1
elif [ "$1" == "--all" ]
then
 all
elif [ "$1" == "--target" ]
then
 target
fi
```

```
sdrv@U:~/SServe$ ./taskA.sh
--all key displays the IP addresses and symbolic names of all hosts in the current subnet
--target key displays a list of open system TCP ports
```

```
sdrv@U:~/SServe$ ./taskA.sh --all
10.10.10.1
U1(10.10.10.17)
U2(10.10.10.18)
```

```
sdrv@U:~/SServe$ ./taskA.sh --target
22/tcp open  ssh
53/tcp open  domain
```

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

1. From which ip were the most requests?

```
user@sdrvadim-vbox:~/SServe$ cat apache_log.txt | awk '{ print $1 ; }' | sort | uniq -c |
sort -n -r | head -n 1
     62 157.55.39.250
```

2. What is the most requested page?

```
user@sdrvadim-vbox:~/SServe$ cat apache_log.txt | awk '{ print $7 ; }' | sort | uniq -c |
sort -n -r | head -n 1
      8 /sitemap1.xml.gz
```

3. How many requests were there from each ip?

```
user@sdrvadim-vbox:~/SServe$ cat apache_log.txt | awk '{ print $1 ; }' | sort | uniq -c | sort -n -r
     62 157.55.39.250
     61 46.29.2.62
     34 207.46.13.48
     10 178.76.227.154
      7 176.59.119.104
      4 157.55.39.174
      3 37.140.141.30
      2 66.249.78.58
      2 217.69.134.29
      2 157.55.39.182
      1 95.108.158.190
      1 93.158.178.129
      1 66.249.78.72
      1 66.249.78.65
      1 66.249.69.39
      1 5.255.253.74
      1 5.255.253.45
      1 217.69.134.39
      1 217.69.134.15
      1 217.69.134.13
      1 217.69.134.12
      1 217.69.134.11
      1 213.87.151.38
      1 185.53.44.186
```

4. What non-existent pages were clients referred to?

```
user@sdrvadim-vbox:~/SServe$ cat apache_log.txt | awk '{ print $4 ; }' | sort | uniq -c |
sort -n | tail
      5 [30/Sep/2015:02:25:50
      5 [30/Sep/2015:02:25:52
      5 [30/Sep/2015:02:25:53
      5 [30/Sep/2015:02:25:54
      5 [30/Sep/2015:02:26:23
      5 [30/Sep/2015:02:26:24
      5 [30/Sep/2015:02:26:51
      5 [30/Sep/2015:02:26:52
      5 [30/Sep/2015:02:26:53
      5 [30/Sep/2015:02:26:55
```

5. What time did site get the most requests?

```
user@sdrvadim-vbox:~/SServe$ cat apache_log.txt | grep "error404" | awk '{ print $1,$15 ;
}' | tr -d '+)"'
157.55.39.174 http://www.bing.com/bingbot.htm
157.55.39.250 http://www.bing.com/bingbot.htm
207.46.13.48 http://www.bing.com/bingbot.htm
```

6. What search bots have accessed the site? (UA + IP)

```
user@sdrvadim-vbox:~/SServe$ cat apache_log.txt | grep "bot"| awk -F '"' '{ print $1,$6 }' | sed -e 's/\[[^][
]*\]//g' | awk -F '-' '{ print $1 $3 $4 $5 }' | sort | uniq
157.55.39.174    Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)
157.55.39.174    Mozilla/5.0 (iPhone; CPU iPhone OS 7_0 like Mac OS X) AppleWebKit/537.51.1 (KHTML, like Geck
o) Version/7.0 Mobile/11A465 Safari/9537.53 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)
157.55.39.182    Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)
157.55.39.182    Mozilla/5.0 (iPhone; CPU iPhone OS 7_0 like Mac OS X) AppleWebKit/537.51.1 (KHTML, like Geck
o) Version/7.0 Mobile/11A465 Safari/9537.53 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)
157.55.39.250    Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)
185.53.44.186    Mozilla/5.0 (compatible; XoviBot/2.0; +http://www.xovibot.net/)
207.46.13.48    Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)
207.46.13.48    Mozilla/5.0 (iPhone; CPU iPhone OS 7_0 like Mac OS X) AppleWebKit/537.51.1 (KHTML, like Gecko
) Version/7.0 Mobile/11A465 Safari/9537.53 (compatible; bingbot/2.0;  http://www.bing.com/bingbot.htm)
217.69.134.11    Mozilla/5.0 (compatible; Linux x86_64; Mail.RU_Bot/Fast/2.0; +http://go.mail.ru/help/robots)
217.69.134.12    Mozilla/5.0 (compatible; Linux x86_64; Mail.RU_Bot/Fast/2.0; +http://go.mail.ru/help/robots)
217.69.134.13    Mozilla/5.0 (compatible; Linux x86_64; Mail.RU_Bot/Fast/2.0; +http://go.mail.ru/help/robots)
217.69.134.15    Mozilla/5.0 (compatible; Linux x86_64; Mail.RU_Bot/Fast/2.0; +http://go.mail.ru/help/robots)
217.69.134.29    Mozilla/5.0 (compatible; Linux x86_64; Mail.RU_Bot/Fast/2.0; +http://go.mail.ru/help/robots)
217.69.134.39    Mozilla/5.0 (compatible; Linux x86_64; Mail.RU_Bot/Fast/2.0; +http://go.mail.ru/help/robots)
37.140.141.30    Mozilla/5.0 (compatible; YandexBot/3.0; +http://yandex.com/bots)
5.255.253.45    Mozilla/5.0 (compatible; YandexBot/3.0; +http://yandex.com/bots)
5.255.253.74    Mozilla/5.0 (compatible; YandexBot/3.0; +http://yandex.com/bots)
66.249.69.39    Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)
66.249.78.58    Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)
66.249.78.58    Mozilla/5.0 (iPhone; U; CPU iPhone OS 4_1 like Mac OS X; enus) AppleWebKit/532.9 (KHTML, like
 Gecko) Version/4.0.5 Mobile/8B117 Safari/6531.22.7 (compatible; MediapartnersGoogle/2.1; +http://www.google.
com/bot.html)
66.249.78.65    Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)
93.158.178.129    Mozilla/5.0 (compatible; YandexBot/3.0; +http://yandex.com/bots)
95.108.158.190    Mozilla/5.0 (compatible; YandexBot/3.0; +http://yandex.com/bots)
```

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]

```
sdrv@U:~/SServe$ cat backup.sh
#!/bin/bash
touch /home/student/$(date +"%Y-%m-%d.%H:%M:%S".log)
rsync -azP --stats /home/student/SServe/ /home/backup/ --log-file="/home/student/$(date +"%Y
-%m-%d.%H:%M:%S".log)"
```

```
 GNU nano 2.2.6        File: /tmp/crontab.4qaU44/crontab

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command

* * * * * /home/student/SServe/backup.sh
```