

# Assignment submission : LINUX COMMANDS

## Command :

```
$ curl -s http://public-dns.info/nameserver/br.csv | cut -d, -f1 | shuf | tail -n 50 | xargs -i timeout 1 ping -c1 -w 1 {} | grep "time=" | awk '{print substr($7, 6, length($7)) " " substr($4, 1, length($4) -1)}' | sort -n | awk '{print $2 " " $1 "ms"}' | head -n 10
```

I created a shell script file called commands.sh whose content output through cat (concatenate) command is shown below.

```
zadmin@zadmin:~/temp$ cat commands.sh
#!/bin/bash

#First script
#Neeraj

pwd
ls

curl -s http://public-dns.info/nameserver/br.csv | cut -d, -f1 | shuf | tail -n 50 | xargs -i timeout 1 ping -c1 -w 1 {} | grep "time=" | awk '{print substr($7, 6, length($7)) " " substr($4, 1, length($4) -1)}' | sort -n | awk '{print $2 " " $1 "ms"}' | head -n 10
```

I changed the file's permissions to executable, through chmod command. Upon executing the file, I got the following output.

```
zadmin@zadmin:~/temp$ ./commands.sh
/home/zadmin/temp
commands.sh
201.76.162.156 313ms
186.230.29.10 325ms
177.67.201.203 374ms
168.181.111.20 396ms
200.195.132.210 401ms
191.7.219.114 423ms
200.169.8.1 432ms
179.183.29.42 478ms
186.194.224.82 507ms
189.3.153.109 508ms
zadmin@zadmin:~/temp$
```

Since this is a pipeline and filtered output, I'll explain each of the commands in their respective order.

Commands and Command line options / arguments :

Curl : Transfers and downloads content from a URL

-s : makes curl silent i.e. omits progress meter and error messages

*Syntex : curl -OPTIONS url*

Cut : Prints selected parts of lines from each file to standard output. If no file is given, then read from standard input.

-d, : delimiter. Input items are terminated by the specified character. For our command, it is ','

-f1 : displays only the first column

*Syntex : cut -OPTIONS file*

Shuf : Generate random permutations of the piped/standard input. If no file is given, then read from standard input.

*Syntex : shuf -OPTIONS file*

Tail : outputs the last parts of the file. If no file is given, then read from standard input.

-n 50 : outputs the last 50 lines as input to the next piped command. By default it gives the last 10 lines.

*Syntex : tail -OPTIONS file*

Xargs : execute command lines from the standard input, which is the output from tail for this command.

-i : Since no replace string is given, it works the same as I{ }.

*Syntex : xargs -OPTIONS command -arguments*

Timeout : runs a command with a given time limit. For our command, given duration is 1.

*Syntex : timeout -OPTION duration*

Ping : send ICMP ECHO\_REQUEST to network hosts. Upon completion it shows a summary of packets transmitted, received, % loss and time taken.

-c1 w 1 {} : Stops after sending 1 ECHO\_REQUEST packet. With w/deadline option, ping waits for 1 ECHO\_REPLY packet until the timeout expires.

*Syntex : ping -OPTIONS destination*

'Grep "time=" : Global regular expression print, searches for a "time=" in each file, and print lines that match the pattern. It is used as a filtering command to omit the lines that don't match the pattern. The matching text is highlighted.

*Syntex : grep -OPTION pattern file*

Awk : Pattern scanning and processing language

```
{print substr($7, 6, length($7)) " " substr($4, 1, length($4) -1)}  
{print $2 " " $1 "ms"}
```

Above arguments prints out the strings at position 6, to display the complete IP address, all but the colon at the end, hence the length-1. Although if we want to print the whole string, we need not give the complete length as the argument. And the second argument prints the time and ms.

Sort : Sort the lines of text file or standard input in this case.

-n : compare according to string numerical value.

*Syntax : sort -OPTIONS file*

Head : Output the first parts of the file. If no file is given, then read from standard input.

-n 10 : By default it prints the first 10 lines, hence we need not give this argument.

*Syntax : head -OPTIONS file*