Linux Assignment

**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

**Explanation:**

**curl -s** [**http://public-dns.info/nameserver/br.csv**](http://public-dns.info/nameserver/br.csv)

curl is used to transfer the data across the network. -s is used to not to show any error messages if it fails to connect.

**cut -d, -f1**

cut removes sections in the file. d represents delimiter and f represents specified number of fields.

The output of above command will be the input to this command as there is a piping. This input is seperated by a delimiter (“,“) and produces one field by ignoring others.

**shuf**

shuf is used to shuffle the data.

The fields produced in the file will be shuffled.

**tail -n 50**

tail gives the specified number of rows from the last in the file

So, the output will be last 50 rows of shuffled data

**xargs -i timeout 1 ping -c1 -w 1 {}**

xargs takes number of arguments as an input

timeout is the time(in seconds) limit to execute the command

ping makes sure that there is a reachability of the host. It sends request packets to network hosts. -c is to specify the number of packets. -w is the waiting time for the reply.

The lastly produced shuffled data consists of ip addresses so ping excutes with each ip address and produces 1 packet from each.

**grep**

grep acts as regular expression. It matches the given pattern in each line of the file, if it consists then it prints the line.

So, here it looks for the pattern “time=” and produces the output.

Then the output will be in this format:

64 bytes from 187.111.34.79: icmp\_seq=1 ttl=243 time=441 ms

**awk '{print substr($7, 6, length($7)) " " substr($4, length($4) -1)}'**

awk is needed to execute the functions.

For example it consists of substr function which gives the substring from the given string. It takes3 arguments. The first argument is the string and then the second argument is the starting index and last argument is the ending index that you wanted to specify.

As we observe we had 8 fields in the above output. First field is represented as $1 and so on.

substr($7, 6, length($7)): produces characters in 7th field starting from 6th index to the last as if the 3rd argument is the last index.

After executing upto this, the output format is: 406 177.200.221.66

This will the input to next command.

**sort -n**

sorts in increasing order.

As the input is in numericals. It sorts in numerical order

**awk '{print $2 " " $1 "ms"}'**

As the input contains 2 fields. First one is $1 and the second one is $2. Then the output of above command will look like: 177.200.221.66 406ms

**head -n 10**

So, the above format will in many number of lines.

head gives the specified number of lines from starting.

The output will be first 10 lines from the above command output.

**Output:**

187.1.175.86 308ms

189.124.138.65 313ms

177.184.220.54 332ms

187.33.230.114 344ms

189.89.165.114 350ms

138.0.53.194 364ms

200.150.113.147 369ms

179.228.67.140 372ms

186.216.62.77 373ms

187.111.27.34 374ms