**Linux assignment**

**The given command is**

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

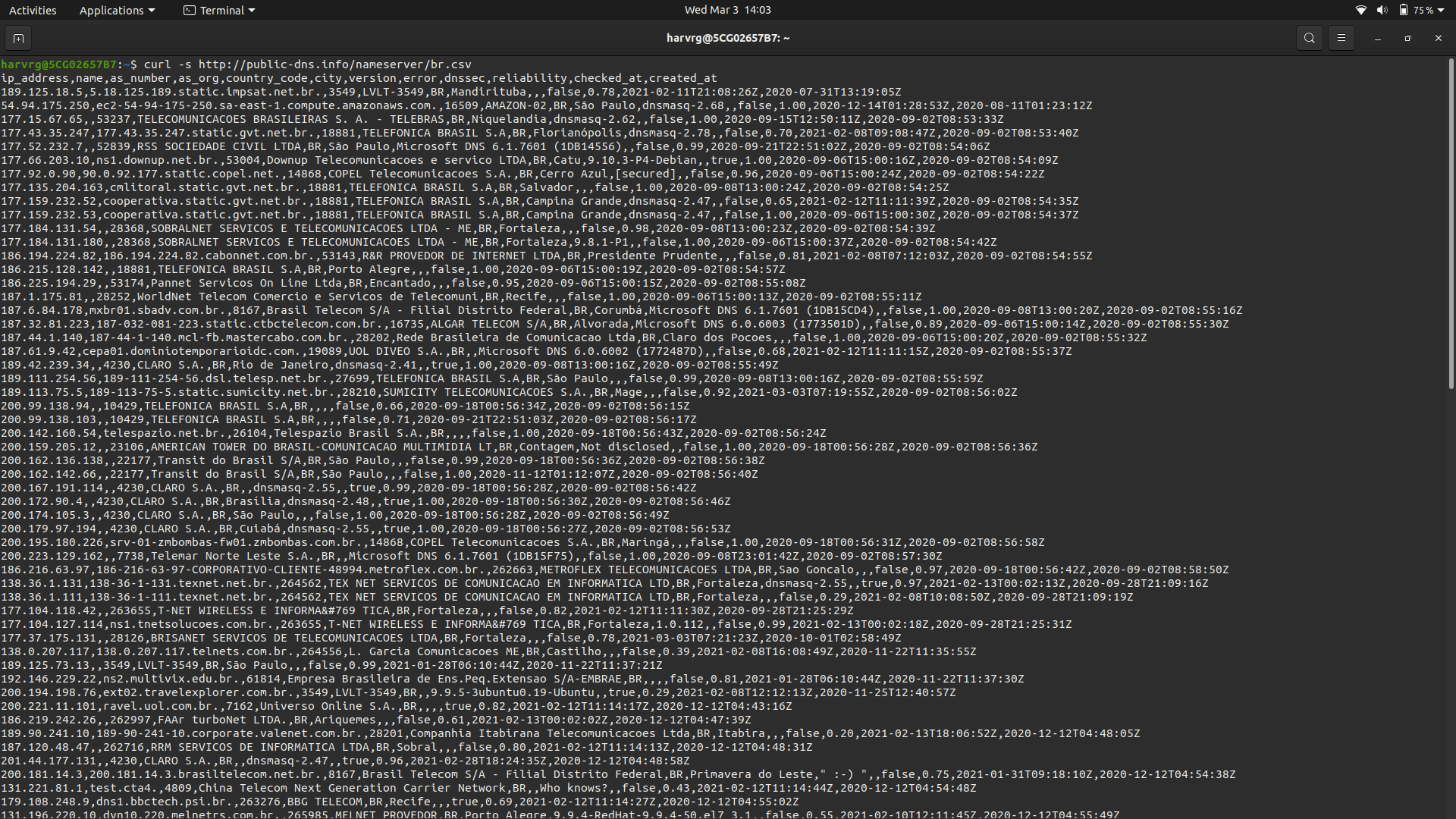
The given command is the combination of multiple commands which are joined with the help of pipes ( | ), these pipe will help the commands to pass its output to other command as a input

So now let’s break the commands and get to know each command and its output which is passed as standard input for the next command

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

This command is used for data transfer to the servers, which is also known as sending the requests to the servers with some protocols such as HTTP, HTTPS, FTP etc.,

Here we are getting the data present in the url and the output is passed to the next command through the pipe

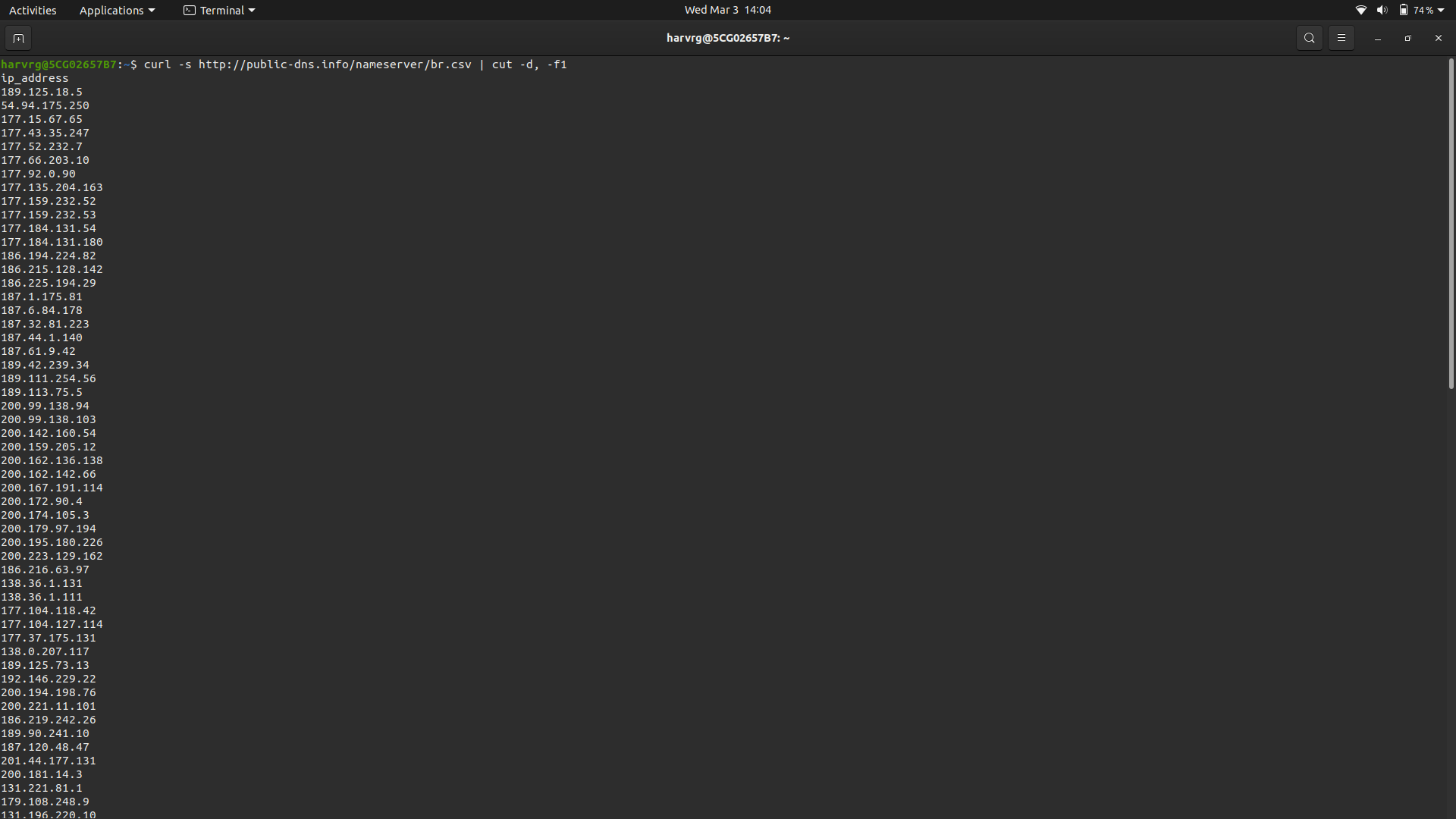


Above is the data which is present in the provided url, the data is sent to the next command

1. **Cut -d, -f1:**

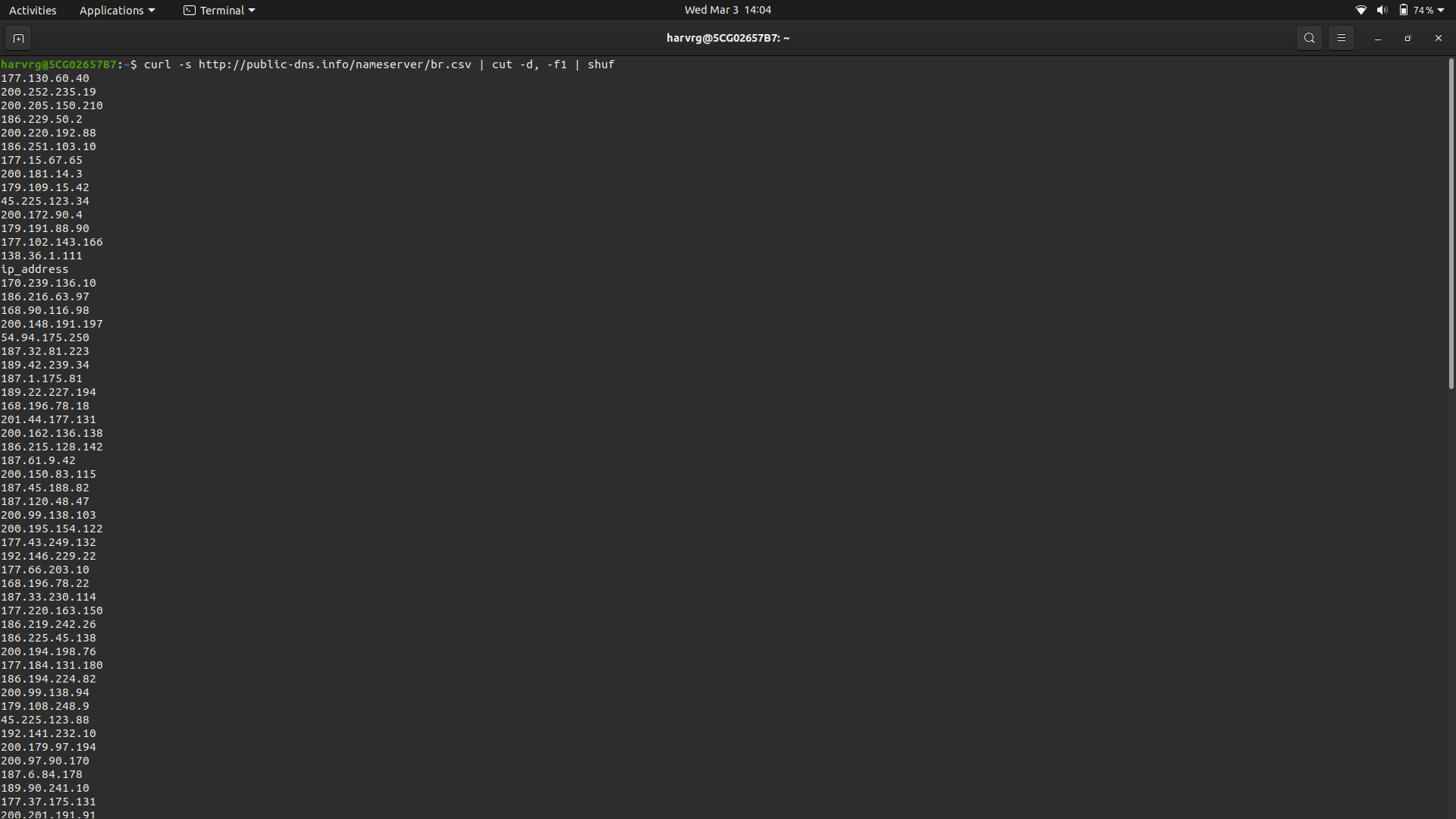
This command is used for cutting the provided data and the output is given to the standard output, this command can also perform the split operations on the provided data

Here in the above command we are splitting the data with a delimiter “,” and we are selecting the 1st field after splitting so the output will be as shown in figure after using this command



1. **Shuf:**

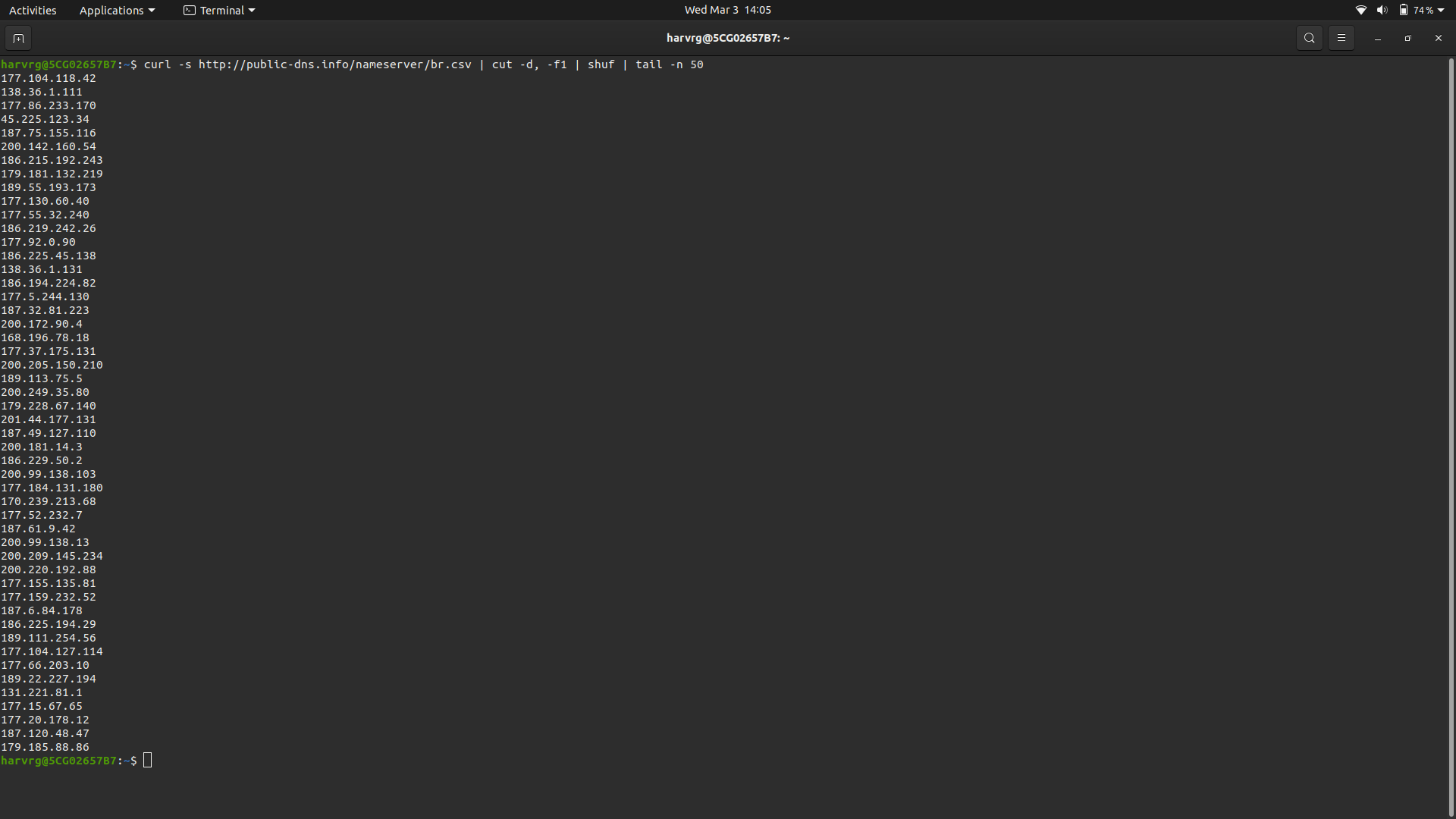
This command is used to randomise the order of the input, this will generate a shuffled order of the input which we have provided



1. **tail -n 50:**

This command is used to display the last n number of the input data, the n can be specified by using the option -n

Here we are displaying the last 50 number of lines from the input, so here is the output after using this command

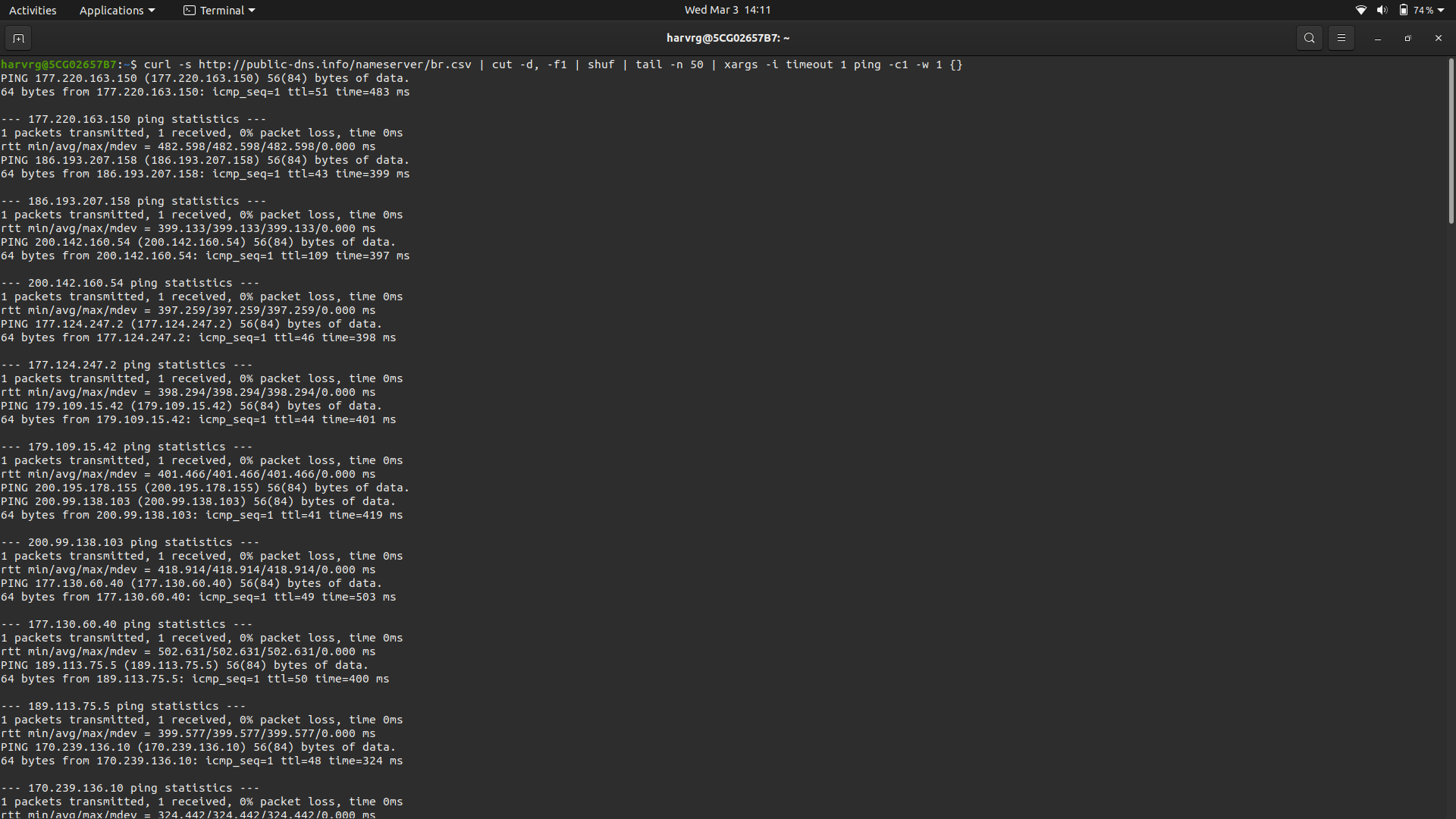


1. **Xargs -i timeout 1 ping -c1 -w 1 {}:**

Xargs reads the items from the standard input and gives that input to a specified command, the -i flag is used to replace the ‘{}’ with input data recursively, we can also mention a string after the -i so we can replace that string with the data but the default is ‘{}’

1. Ping command is used to get the ping from a specific address which is given through the standard input, -c1 flag is used to obtain only one column when generating the ping from the server, -w flag is used to give a timeout to the ping command so here we are providing 1sec as the timeout

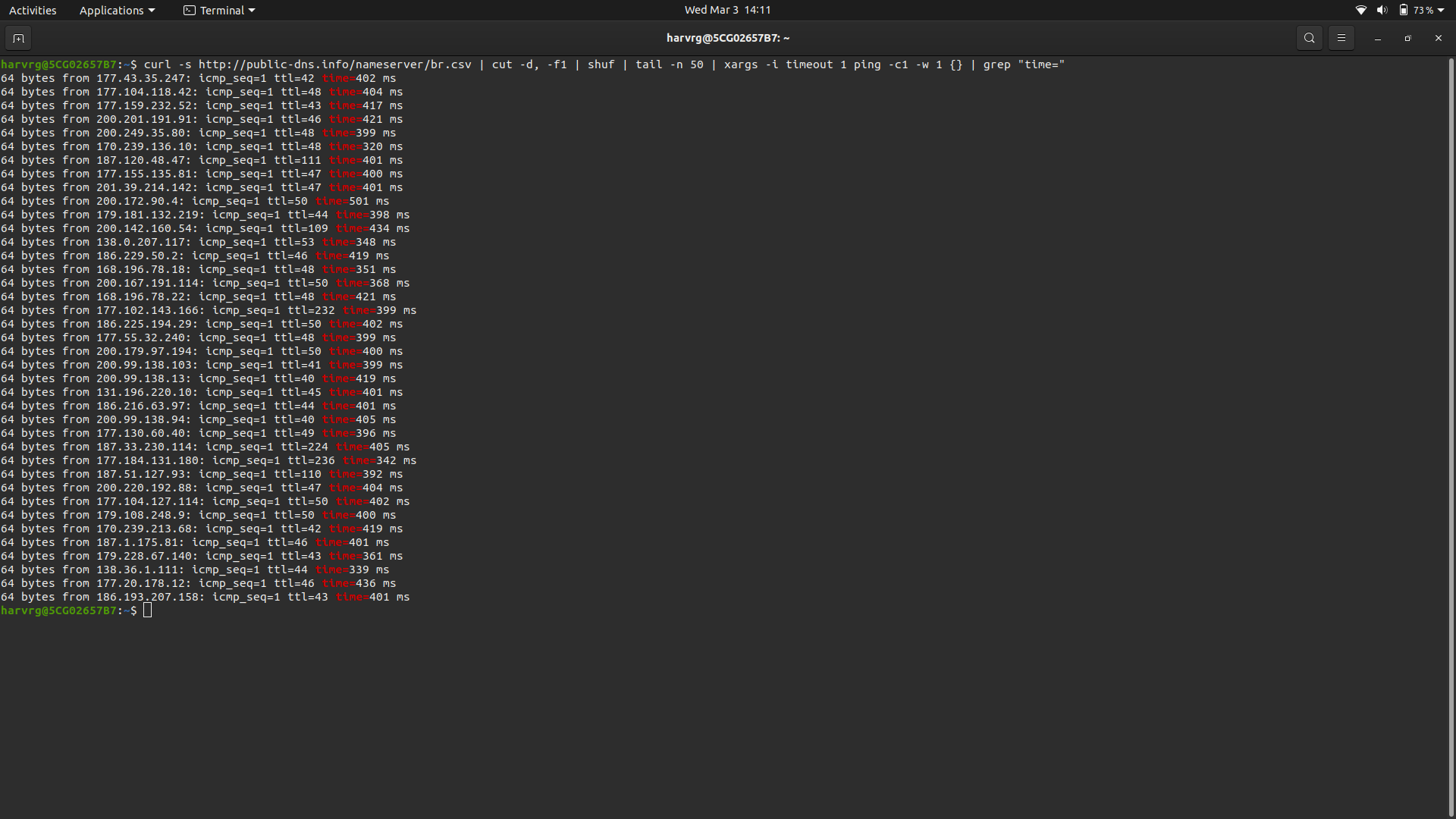
Here is the output of the command after execution.



**6. Grep “time=” :**

This command is used to match the given string in the data, and display the lines which contain that string.

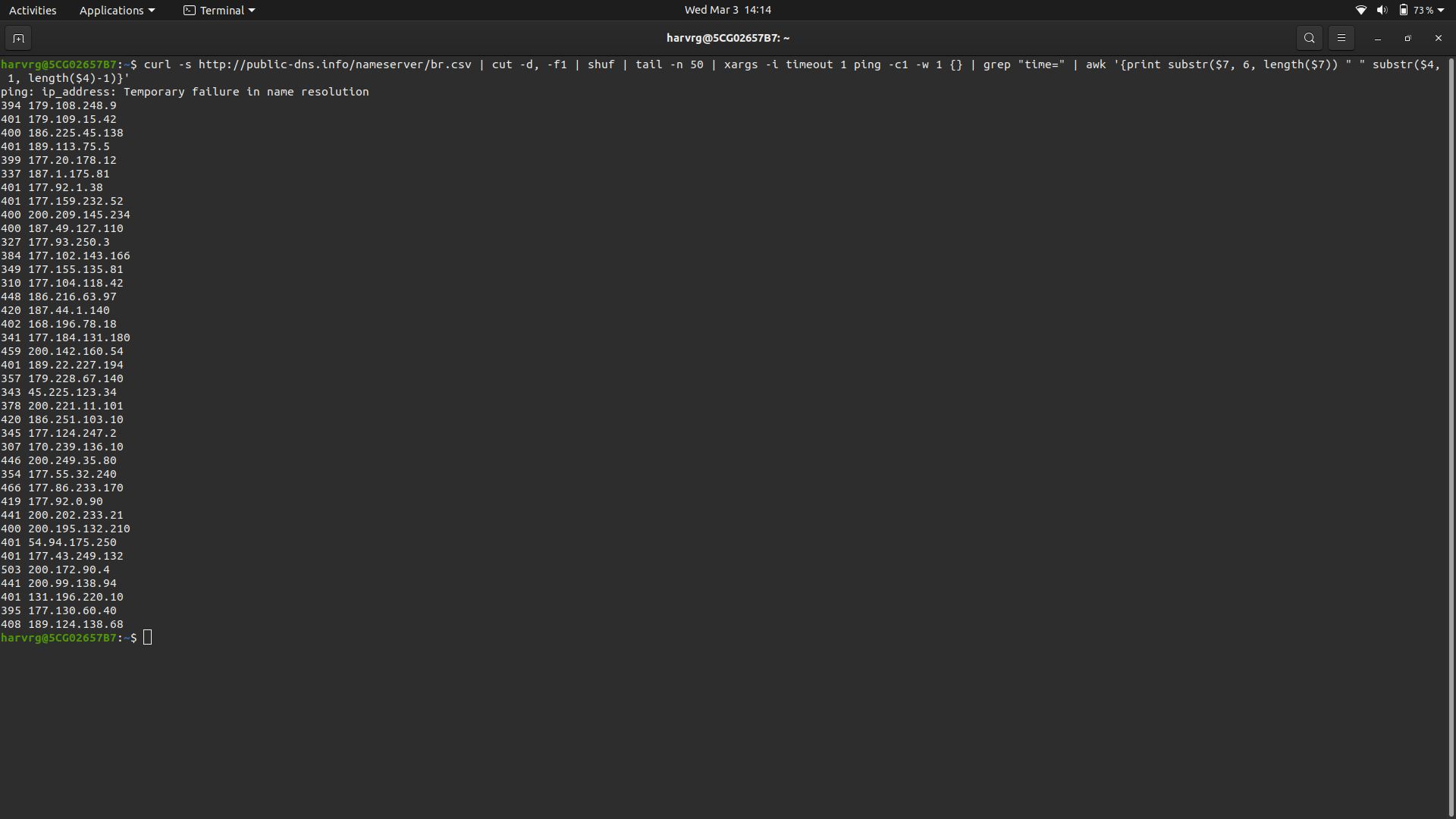
Here is the output of the grep command



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

Awk is a scripting language used for manipulating data and generating reports.The awk command programming language requires no compiling, and allows the user to use variables, numeric functions, string functions, and logical operators.

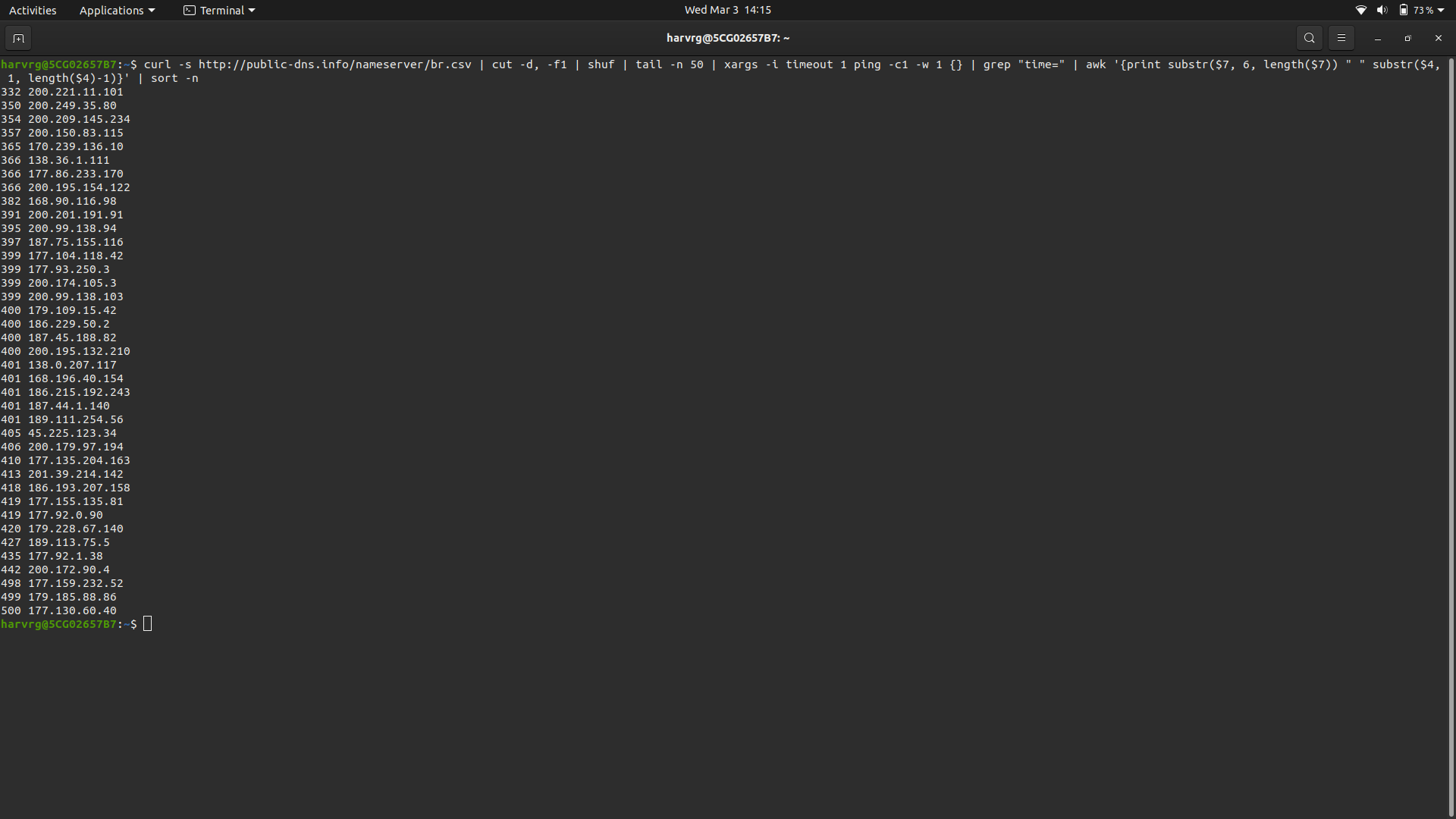
The script print substr($7, 6, length($7)) " " substr($4, 1, length($4) -1) implies that $number is the number of the word in the data, so we are slicing the input lines given from the grep so that to extract the address and time in the input lines



**8. Sort -n:**

This command is used to sort the given input lines, the option -n is to specify that the input is considered as a numerical values, by default it consider them as a string

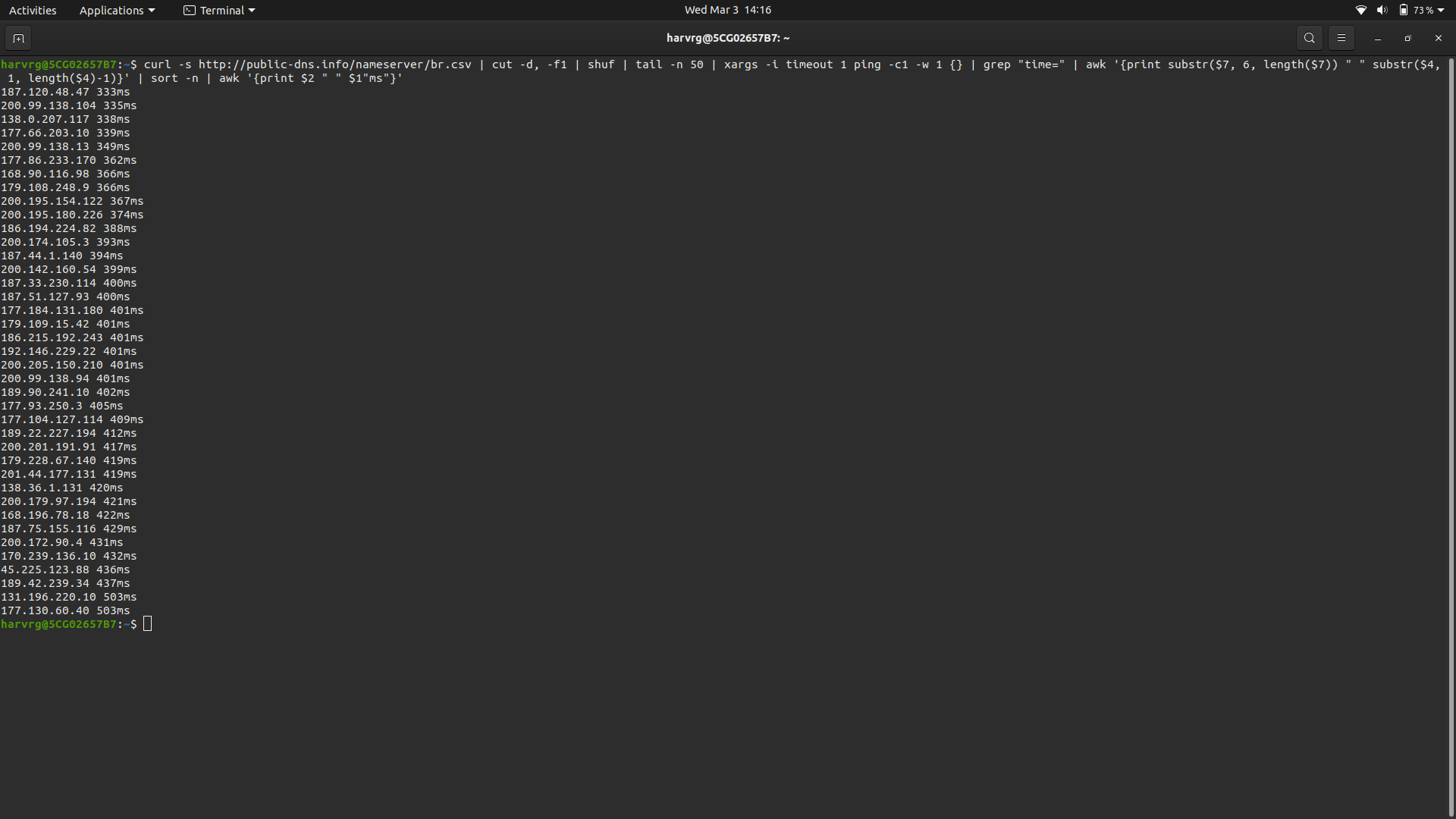
Here is the data after sorting the data of the input considering them as a numerical values



**9. Awk ‘{print $2 “ “ $1”ms”}’:**

This awk command is used to execute a scripting language which does not need any compilation, this statement is used to print the second word(address) to be first and the first word(time) to be second and followed by the string “ms”

Here is the output of the command after it executed.



**10. Head -n 10:**

This command is opposite to the tail, it will print top n number of lines of the input, the number n can be mentioned by using the option -n

Here is the output of the command for printing the top 10 lines of the standard input

