Linux

Working with directories

A brief overview of the most common commands to work with directories: pwd, cd, ls, mkdir and rmdir. These commands are available on any Linux (or Unix) system.

- **pwd** : Print Working Directory (Will tell you the location you are currently working)
- **cd** : You can change your current directory with the cd command
  - **cd** : shortcut to get back into your home directory. Just typing cd without a target directory, will put you in your home directory
  - **cd ..** : To go to the parent directory (the one just above your current directory in the directory tree)
  - **cd** - : To go to the previous directory
- **ls** : You can list the contents of a directory with ls
  - **ls -a** : To show all files. Showing all files means including the hidden files. When a file name on a Linux file system starts with a dot, it is considered a hidden file and it doesn't show up in regular file listings.
  - **ls -l** : to display the contents of the directory in different formats or to display different parts of the directory.
  - **ls -lh** : shows the numbers (file sizes) in a more human readable format.
- **mkdir** : Create new directories
- **mkdir** -p: To create parent directories as needed
- **rmdir**: To remove the directory. (Directory has to be empty)

Exercise

1. Login to your VM. and Display your current directory
2. Change to /etc directory and display current directory
3. Go to root directory and list the contents
4. List a long listing of the root directory
5. Go to your home directory
6. Make directory named 'test'
7. make a directory inside test directory named **'one'** and make a hidden directory inside **'one'** directory named **'.hidden'**. Make a directory inside test directory named **'one'** and make a hidden directory inside **'one'** directory named **'unhidden'**.
8. Go to 'one' directory and list the content.
9. Then list all contents
10. Remove directory **'unhidden'**
11. Go to your home and try to remove directory **'test'**
12. Type '**pwd'** in different directories
13. Go to **'cd /etc'** → type '**pdw'** → type **'ls'** → type **'ls -l'**

File Editing

The vi editor is installed on almost every Unix. Linux will very often install vim (vi improved) which is similar. Every system administrator should know vi(m), because it is an easy tool to solve problems.

vi Commands and Tips

Command	**Result**
Open:	
vi filename	(fn=filename)
vi + filename	Place the cursor on last line of file.
vi +n filename	Place the cursor on line "n" of file.
vi +/pat filename	Place cursor on line with first occurrence of "pat"tern
Close:	
:w	Write the file to disk. Don't exit.
:w!	Write the file to disk even if read/only.
:wq	Write the file to disk and exit.
:wq!	Write the file to disk even if read/only and quit.
:q	Quit the file (only if no changes).
:q!	Quit the file even if changes.
Movement:	
A	Move to end of line, change to insert mode.
h	Move 1 space backwards (back/left arrow).
j	Move down 1 line (down arrow).
k	Move up 1 line (up arrow).
l	Move 1 space forwards (forward/right arrow)
w	Move cursor to start of next word.
W	Same as "w".
b	Move cursor to start of previous word.
B	Same as "b".
:n	Go to line number "n" in the file.
Editing:	
i	Enter in to input mode.
o	Add a line below cursor and enter in to input mode.
x	Delete character (del key in some cases).
D	Delete line from right of cursor to end of line.
dd	Delete entire line.
u	Undo last edit or restore current line.
yy	Yank current line.
p	Put yanked text before the cursor.
Searching:	
/pattern	Search for "pattern" in the file going forwards.
?pattern	Search for "pattern" in the file going backwards.
n	Find the next occurrence of pattern found forwards.
N	Find next occurrence of pattern found backwards.

Copy/ Cut and Paste	
<NUM>yyp	Copy n lines to buffer, paste below cursor
<NUM>yyP	Copy n lines to buffer, paste above cursor
<NUM>ddp	Cut n lines and copy to buffer, paste below cursor
<NUM>ddP	Cut n lines and copy to buffer, paste above cursor
Find and replace	
ESC: %s/Pattern/text/g	Find and replace words in vi editor (**pattern** – word to find & replace, **text** – word to replace with)
ESC: %s/\<pattern\>/text/g	Replace only an exact matching word in vi editor
Locate	
locate [name]	finds files in Linux using the file name

Practicing with vi

The easiest thing to do if you get confused in vi is to press the ESCape key a couple of times and start over with what you were doing. Log in to your VM and...

1. Type **'cd'**
2. Type **'vi temp.txt'**

*vi wil create the file **"temp.txt"** for you. Press the **"i"** key to switch to input mode.*

Type something like,

"VI is great! I think I'll be using vi from now on instead of Word"

Press <ENTER> to add lines.
Type some more text

Save the file that you are in. To do this do:

*Press the ESCape key for command mode Type **":wq"** then hit Enter to save and quit the file (notice the **":"** before the **"wq"**).*

Copy a large file to your home directory so that you can play around with some more vi commands. We'll copy over your /etc/sysctl.conf file for this exercise. To do this do:

$ cd
$ cp /etc/sysctl.conf sysctl.conf.bak

Edit the file, but let's start at the bottom of the file:

$ vi + sysctl.conf.bak

Go to the first line of the file. Notice the colon (":") before the "1".

:1 <ENTER>

Go to line 10, add a new line, and add in some text:

:10 <ENTER>
Press the "i" key

Add the following text:
##
## A sample comment
##

Delete the three lines you just created:

*Move to the first line of new text Press the ESCape key Press **"dd"** to delete a line, repeat until the text is gone*

Save the file, but don't exit.

:w
press <ENTER>

Practice copying and pasting text.

Go to line 12, copy 3 lines of text, go to the bottom of the file, place the text there:

ESC	**(go to command mode)**
:12 <ENTER>	**(go to line 12 of the file)**
3yy	**("yank" 3 lines of text and place in copy buffer)**
G	**(go to the end of the file)**
P	**(place the contents of the copy buffer here)**

If want to undo this you would type (in command mode):

u

Now let's exit from the file and not save the few changes we've made.

:q! <Enter>

Nano Editor

GNU nano is **a text editor for Unix-like computing systems or operating environments using a command line interface**. It emulates the Pico text editor, part of the Pine email client, and also provides additional functionality. Easy to use and all the commands are given in the editor.

To create a new file type **nano test.txt**

Find and Replace

Create a file **test1.txt**

Add the following text inside **test.txt**
Foo
Fool
Food
Foo

Replace **Foo** with **Bar**
Command - ESC: %s/Foo/Bar/g
Enter, then save and quit
Find and replace exact matching word

Create a file **test2.txt**

Add the following text inside **test.txt**
Foo
Fool
Food
Foo

Replace **Foo** with **Bar**
Command - ESC: %s/\<Foo\>/Bar/g
Enter then save and quit

Working with Files

Files on Linux (or any Unix) are case sensitive. This means that FILE1 is different from file1, and /etc/hosts is different from /etc/Hosts (the latter one does not exist on a typical Linux computer). In Linux everything is considered as a file even a directory is a special kind of file. A small overview of some basic and important file handling commands

- **file:** The file utility determines the file type. Linux does not use extensions to determine the file type. The command line does not care whether a file ends in .txt or .pdf. As a system administrator, you should use the file command to determine the file type.
- **cp:** To copy a file, use cp with a source and a target argument.
  - **cp -r:** To copy complete directories, use cp -r (the -r option forces recursive copying of all files in all subdirectories).
  - **cp -i:** To prevent cp from overwriting existing files, use the -i (for interactive) option.
- **mv:** Use to rename a file or to move the file to another directory.
- **rm:** Use to remove files ***rm -i**:To prevent yourself from accidentally removing a file. ***rm -rf:rm -r** will not remove non-empty directories. However, rm accepts several options that will allow you to remove any directory. The **rm -rf** will erase anything

Dangerous Command

When you are logged on as root, be very careful with rm -rf (the f means force and the r means recursive) since being root implies that permissions don't apply to you. You can literally erase your entire file system by accident.

- **less**:The less command is useful for displaying files that take up more than one screen
- **head/tail:**You can use head to display the first ten lines of a file. and tail to display the last ten lines of a file. you can use both commands with -n and specify the number of lines
- **cat:**The cat command is one of the most universal tools, yet all it does is copy standard input to standard output.

Exercise

Create numbers.txt (Containing ten lines of numbers one to ten) file with cat command.

```
$ cat > numbers.txt
```

*Add the lines with enter key at the end. After inserting all the lines press **Ctrl + D***

```
one
two
. .
Ten
```

*View the file you created using **less** and **cat***

Cat

```
$ cat numbers.txt
one
two
three
four
five
six
seven
eight
nine
ten
```

less

```
$ less numbers.txt
```

Press **q** to exit from less

View first four lines and last four lines using head and tail commands

```
$ head -n 4 numbers.txt
one
two
three
four
$ tail -n 4 numbers.txt
seven
eight
nine
ten
```

Create another file numbers2.txt(with lines contain eleven to fifteen)

```
$ cat > numbers2.txt
eleven
twelve
thirteen
fourteen
fifteen
```

Combine numbers.txt and numbers2.txt and create numbers3.txt. and view the file.

```
$ cat numbers.txt numbers2.txt > numbers3.txt
$ cat numbers3.txt

one
two
three
four
five
six
seven
eight
nine
ten
eleven
twevelve
thirteen
fourteen
fifteen
```

Check the file format of the newly created file

```
$ file numbers3.txt
numbers3.txt: ASCII text
```

Copy all the created files to the test directory. and verify

```
$ cp numbers.txt numbers2.txt numbers3.txt test/
$ cd test
$ ls
numbers2.txt  numbers3.txt  numbers.txt  one
```

Make a copy of test directory as newtest and rename it to numbers

```
$ cd
$ cp -r test/ newtest
$ mv newtest/ numbers
```

Delete the test directory

```
$ rm -rf test
```

File Permission

- Commands preceded with "$" imply that you should execute the command as a general user - not as root.
- Commands preceded with "#" imply that you should be working as root with "sudo"
- Commands with more specific command lines (e.g. "RTR-GW>" or "mysql>") imply that you are executing commands on remote equipment, or within another program.

chmod: Change file read write permission **chown**: Change the owner of the file **chgrp**: Change the group of the file

Reference

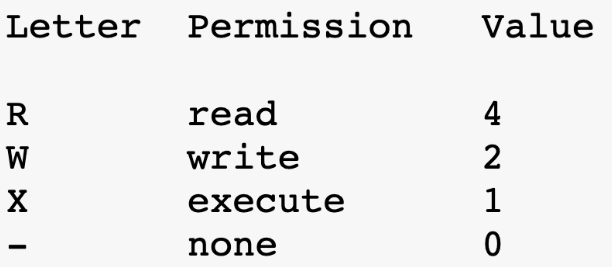
If you look at files in a directory using "ls -al" you will see the permissions for each file and directories. Here is an example:

```
drwxr-xr-x  6 test   radius 4096 Oct 16 18:18 .
drwxr-xr-x  3 root   root   4096 Sep 21 12:29 ..
```

So, the directory has r (read), w (write), x (execute) access for the User and Group. For Other it has r (read) and x (execute) access. The file has read/write/execute access for User and read only access for everyone else (Group and Other).

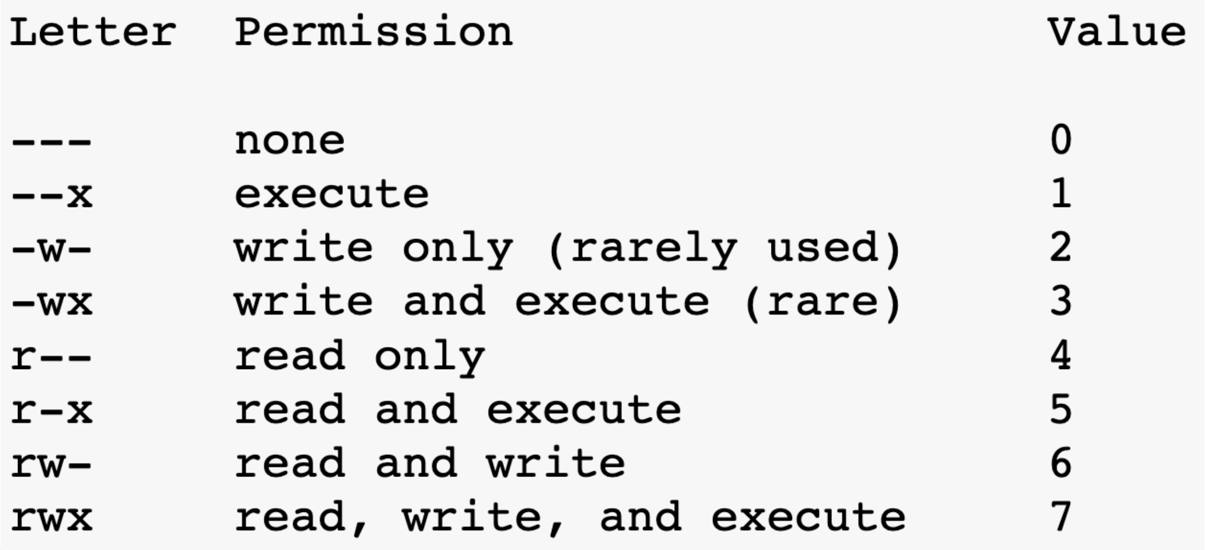
To change permissions you use the **chmod** command. chmod uses a base eight (octal) system to configure permissions. Or, you can use an alternate form to specify permissions by column (User/Group/Other) at a time.

Permissions have values like this:



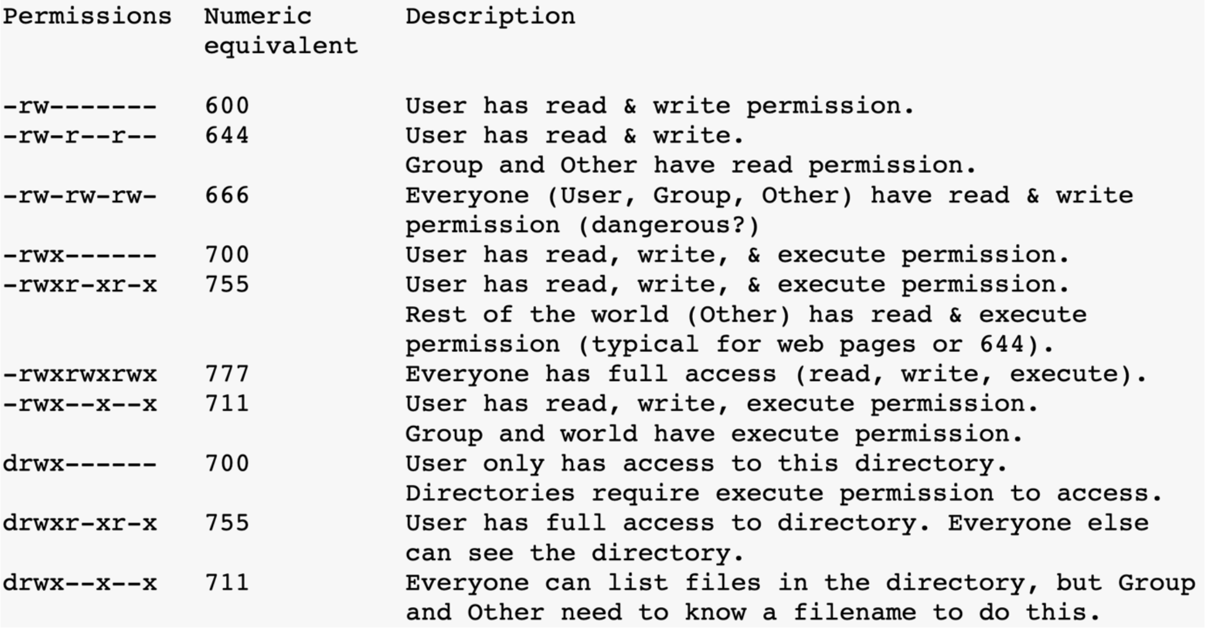
Letter	Permission	Value
R	read	4
W	write	2
X	execute	1
-	none	0

Thus you can give permissions to a file using the sum of the values for each permission you wish to give for each column. Here is an example:



Letter	Permission	Value
---	none	0
--x	execute	1
-w-	write only (rarely used)	2
-wx	write and execute (rare)	3
r--	read only	4
r-x	read and execute	5
rw-	read and write	6
rwx	read, write, and execute	7

This is just one column. Since we have three areas of permissions (User, Group, Other), it looks like this will all 3 sets:



Permissions	Numeric equivalent	Description
-rw-------	600	User has read & write permission.
-rw-r--r--	644	User has read & write. Group and Other have read permission.
-rw-rw-rw-	666	Everyone (User, Group, Other) have read & write permission (dangerous?)
-rwx------	700	User has read, write, & execute permission.
-rwxr-xr-x	755	User has read, write, & execute permission. Rest of the world (Other) has read & execute permission (typical for web pages or 644).
-rwxrwxrwx	777	Everyone has full access (read, write, execute).
-rwx--x--x	711	User has read, write, execute permission. Group and world have execute permission.
drwx------	700	User only has access to this directory. Directories require execute permission to access.
drwxr-xr-x	755	User has full access to directory. Everyone else can see the directory.
drwx--x--x	711	Everyone can list files in the directory, but Group and Other need to know a filename to do this.

Exercise

Go to the numbers directory and get a detailed list

```
cd numbers
ls -al
```

```
drwxr-xr-x    3       test test 4096 Oct 17 13:53 .
drwxrwxrwx  50       test test 4096 Oct 17 13:10 ..
-rw-r--r--      1       test test      42 Oct 17 13:08 numbers2.txt
-rw-r--r--      1       test test      91 Oct 17 13:08 numbers3.txt
-rw-r--r--      1       test test      49 Oct 17 13:08 numbers.txt
drwxr-xr-x    3       test test 4096 Oct 17 13:08 one
```

Change file permission as follows

$ chmod 044 numbers.txt

Now you have remove read privilege try view the file using cat

$ chmod 444 numbers.txt

Now you have set privilege as read only. Open the file via vi editor and try to edit the file Switch to root user

$ sudo su

- *Change the ownership and group of the numbers2.txt to root and make it read only for all the other users.*
- *Change the ownership and group of the numbers3.txt to root and remove all the privileges from all the other users.*
- *Switch back to your user and try to view numbers3.txt and try to edit numbers2.txt*

```
# chown root numbers2.txt
# chgrp root numbers2.txt
# chown root numbers3.txt
# chgrp root numbers2.txt
# chmod 700 numbers3.txt
# chmod 744 numbers2.txt
# su 'Your username'
$ cat numbers3.txt
cat: numbers3.txt: Permission denied
$ cat > numbers2.txt
bash: numbers2.txt: Permission denied
```

More Linux Commands

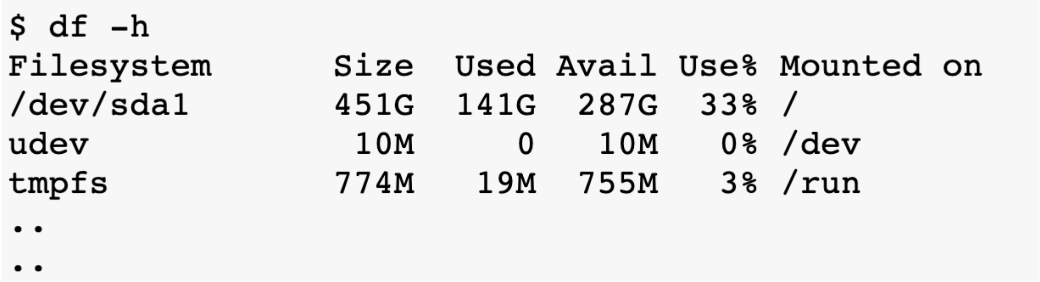
who*: who command will give you information about who is logged on the system.*

$ who

whoami *command tells you your username*

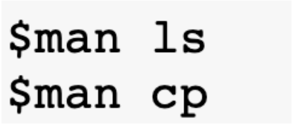
$ whoami

df -dh: report file system disk space usage



```
$ df -h
Filesystem       Size  Used Avail Use% Mounted on
/dev/sda1        451G  141G  287G  33% /
udev              10M     0   10M   0% /dev
tmpfs            774M   19M  755M   3% /run
..
..
```

man*: Will give you man pages (also called manual pages) on your Unix or Linux computer*



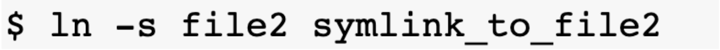
```
$man ls
$man cp
```

date*: Will give you the date and time of the system*

$ date

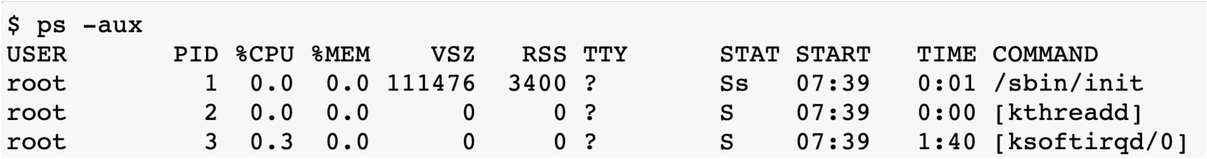
To set time zone – type ***$ timedatectl set-timezone Asia/Colombo***

ln*: Use to create links between files. Frequently use with* ***-s*** *to create symbolic links*



```
$ ln -s file2 symlink_to_file2
```

ps*: report a snapshot of the current processes. frequently use with -aux.*



```
$ ps -aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0 111476  3400 ?        Ss   07:39   0:01 /sbin/init
root         2  0.0  0.0      0     0 ?        S    07:39   0:00 [kthreadd]
root         3  0.3  0.0      0     0 ?        S    07:39   1:40 [ksoftirqd/0]
```

Networking

Get network related information

netstat -a | more : To show both listening and non-listening sockets.

netstat -at : To list all tcp ports.

netstat -au : To list all udp ports.

netstat -l : To list only the listening ports.

netstat -lt : To list only the listening tcp ports.

netstat -lu : To list only the listening udp ports.

Checking the network connectivity

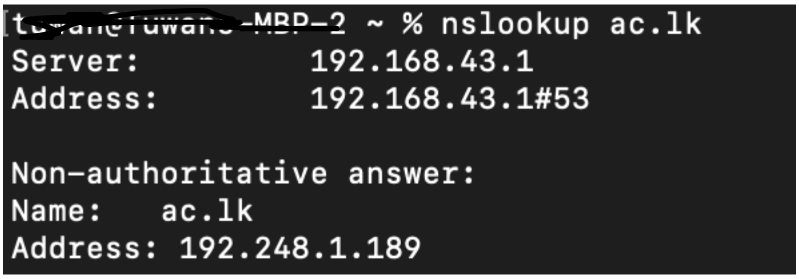
Ping 8.8.8.8
Ping google.com

The tracepath command in Linux allows to trace the path to the destination path determining MTU along this path using UDP port or any other ports that will not require any superuser permissions.

tracepath [destination]

nslookup is the name of a program that lets an Internet server administrator or any computer user enter a host name (for example, "whatis.com") and find out the corresponding IP address or domain name system (DNS) record.

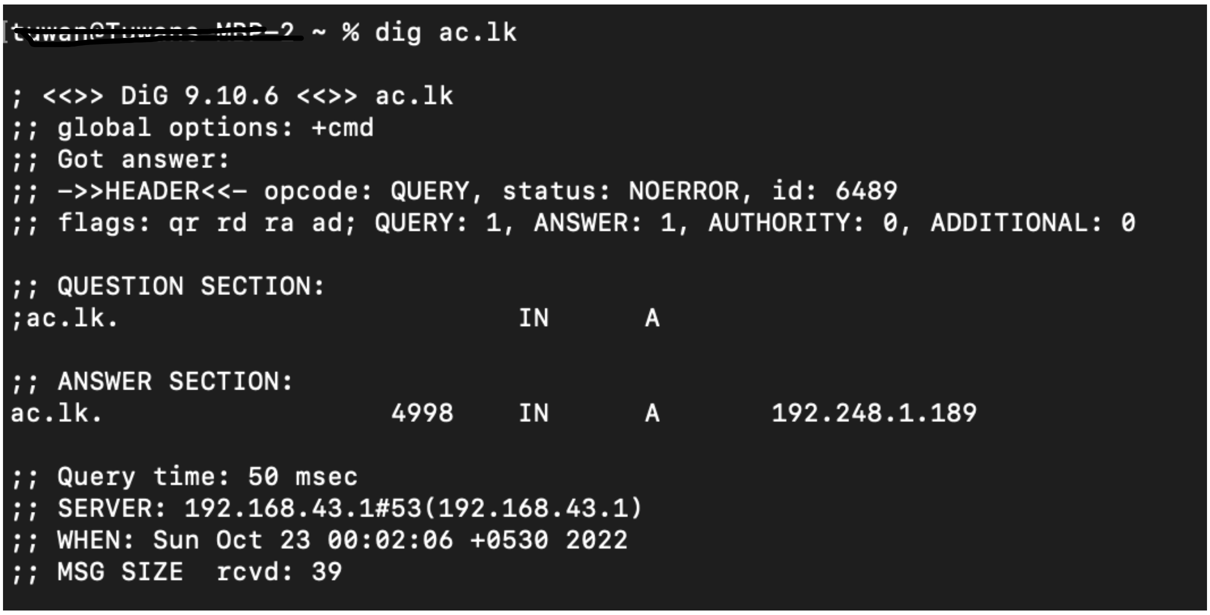
nslookup ac.lk



```
[[redacted] ~ % nslookup ac.lk
Server:         192.168.43.1
Address:        192.168.43.1#53

Non-authoritative answer:
Name:   ac.lk
Address: 192.248.1.189
```

The dig (domain information groper) command is a flexible tool for **interrogating DNS name servers**. It performs DNS lookups and displays the answers that are returned from the queried name server(s).

dig ac.lk

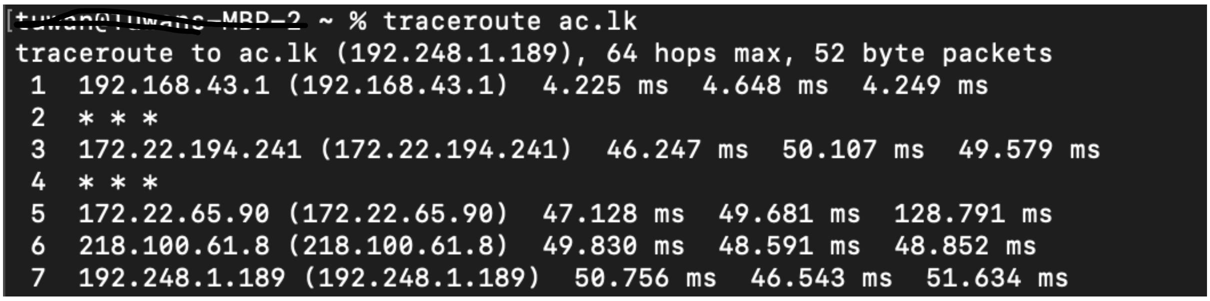


```
[[redacted] ~ % dig ac.lk

; <<>> DiG 9.10.6 <<>> ac.lk
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6489
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;ac.lk.                         IN      A

;; ANSWER SECTION:
ac.lk.                  4998    IN      A       192.248.1.189

;; Query time: 50 msec
;; SERVER: 192.168.43.1#53(192.168.43.1)
;; WHEN: Sun Oct 23 00:02:06 +0530 2022
;; MSG SIZE  rcvd: 39
```

traceroute command in Linux prints the route that a packet takes to reach the host. This command is useful when you want to know about the route and about all the hops that a packet takes.

traceroute ac.lk

*On a **Windows** machine, this command is called **tracert**; on **Linux and Mac**, it's called **traceroute**.*



```
[[redacted] ~ % traceroute ac.lk
traceroute to ac.lk (192.248.1.189), 64 hops max, 52 byte packets
 1  192.168.43.1 (192.168.43.1)  4.225 ms  4.648 ms  4.249 ms
 2  * * *
 3  172.22.194.241 (172.22.194.241)  46.247 ms  50.107 ms  49.579 ms
 4  * * *
 5  172.22.65.90 (172.22.65.90)  47.128 ms  49.681 ms  128.791 ms
 6  218.100.61.8 (218.100.61.8)  49.830 ms  48.591 ms  48.852 ms
 7  192.248.1.189 (192.248.1.189)  50.756 ms  46.543 ms  51.634 ms
```

ipconfig stands for Internet Protocol Configuration**,** while **ifconfig** stands for Interface Configuration.

ifconfig

ipconfig