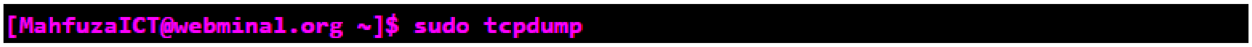
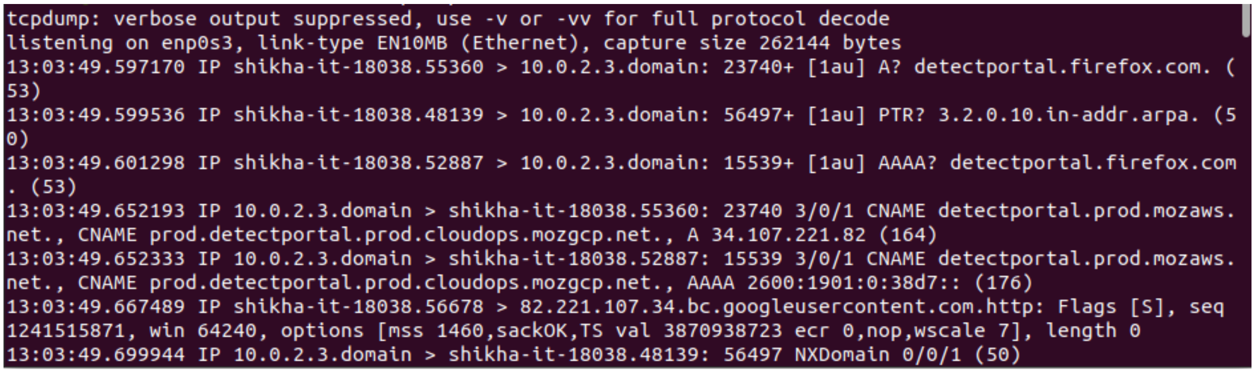
Linux Tool :

1.Sudo tcpdump :

Tcpdump is a packet sniffing and packet analyzing tool for a System Administrator to troubleshoot connectivity issues in Linux. It is used to capture, filter, and analyze network traffic such as TCP/IP packets going through your system. It is many times used as a security tool as well. It saves the captured information in a pcap file, these pcap files can then be opened through a wireshark or through the command tool itself.



```
[MahfuzaICT@webminal.org ~]$ sudo tcpdump
```

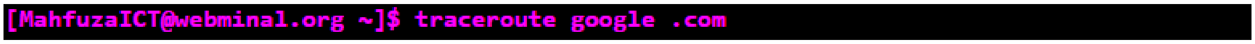


```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
13:03:49.597170 IP shikha-it-18038.55360 > 10.0.2.3.domain: 23740+ [1au] A? detectportal.firefox.com. (
53)
13:03:49.599536 IP shikha-it-18038.48139 > 10.0.2.3.domain: 56497+ [1au] PTR? 3.2.0.10.in-addr.arpa. (5
0)
13:03:49.601298 IP shikha-it-18038.52887 > 10.0.2.3.domain: 15539+ [1au] AAAA? detectportal.firefox.com
. (53)
13:03:49.652193 IP 10.0.2.3.domain > shikha-it-18038.55360: 23740 3/0/1 CNAME detectportal.prod.mozaws.
net., CNAME prod.detectportal.prod.cloudops.mozgcp.net., A 34.107.221.82 (164)
13:03:49.652333 IP 10.0.2.3.domain > shikha-it-18038.52887: 15539 3/0/1 CNAME detectportal.prod.mozaws.
net., CNAME prod.detectportal.prod.cloudops.mozgcp.net., AAAA 2600:1901:0:38d7:: (176)
13:03:49.667489 IP shikha-it-18038.56678 > 82.221.107.34.bc.googleusercontent.com.http: Flags [S], seq
1241515871, win 64240, options [mss 1460,sackOK,TS val 3870938723 ecr 0,nop,wscale 7], length 0
13:03:49.699944 IP 10.0.2.3.domain > shikha-it-18038.48139: 56497 NXDomain 0/0/1 (50)
```

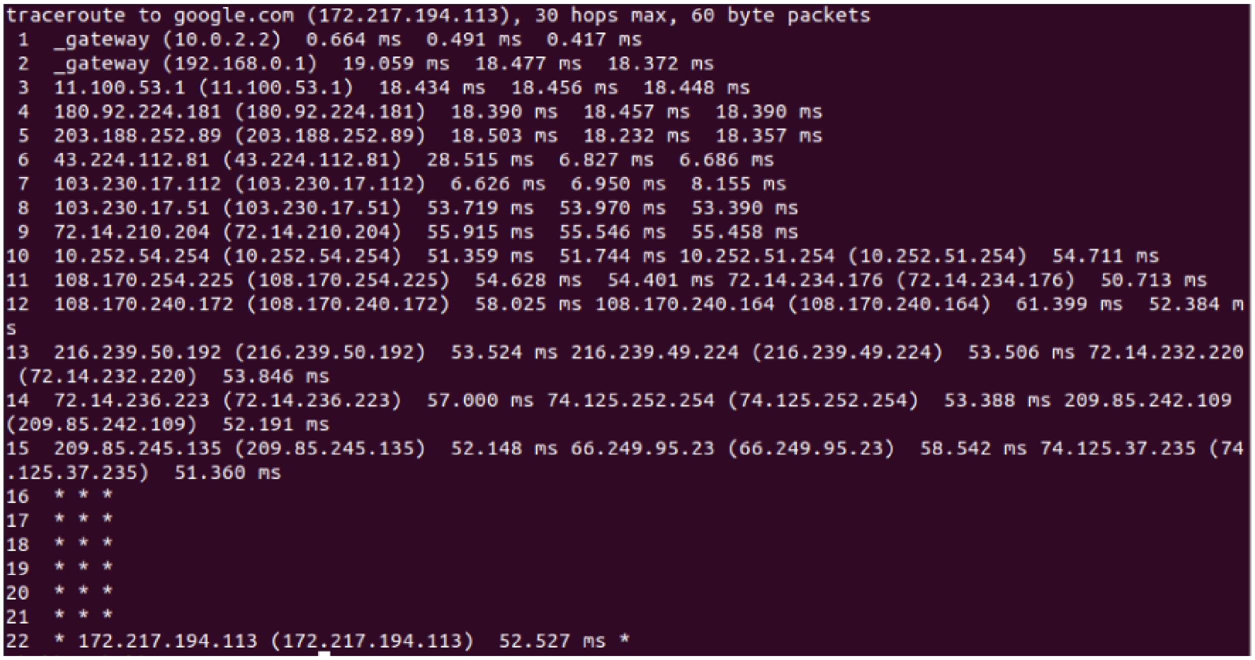
2.Traceroute :

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. Below image depicts how traceroute command is used to reach the Google(172.217.26.206) host from the local machine and it also prints detail about all the hops that it visits in between.

The first column corresponds to the hop count. The second column represents the address of that hop and after that, you see three space-separated time in milliseconds. traceroute command sends three packets to the hop and each of the time refers to the time taken by the packet to reach the hop.



```
[MahfuzaICT@webminal.org ~]$ traceroute google .com
```

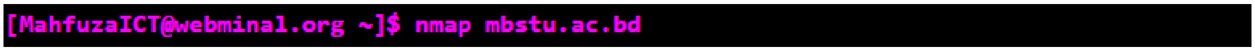


```
traceroute to google.com (172.217.194.113), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.664 ms  0.491 ms  0.417 ms
 2  _gateway (192.168.0.1)  19.059 ms  18.477 ms  18.372 ms
 3  11.100.53.1 (11.100.53.1)  18.434 ms  18.456 ms  18.448 ms
 4  180.92.224.181 (180.92.224.181)  18.390 ms  18.457 ms  18.390 ms
 5  203.188.252.89 (203.188.252.89)  18.503 ms  18.232 ms  18.357 ms
 6  43.224.112.81 (43.224.112.81)  28.515 ms  6.827 ms  6.686 ms
 7  103.230.17.112 (103.230.17.112)  6.626 ms  6.950 ms  8.155 ms
 8  103.230.17.51 (103.230.17.51)  53.719 ms  53.970 ms  53.390 ms
 9  72.14.210.204 (72.14.210.204)  55.915 ms  55.546 ms  55.458 ms
10  10.252.54.254 (10.252.54.254)  51.359 ms  51.744 ms 10.252.51.254 (10.252.51.254)  54.711 ms
11  108.170.254.225 (108.170.254.225)  54.628 ms  54.401 ms 72.14.234.176 (72.14.234.176)  50.713 ms
12  108.170.240.172 (108.170.240.172)  58.025 ms 108.170.240.164 (108.170.240.164)  61.399 ms  52.384 m
s
13  216.239.50.192 (216.239.50.192)  53.524 ms 216.239.49.224 (216.239.49.224)  53.506 ms 72.14.232.220
 (72.14.232.220)  53.846 ms
14  72.14.236.223 (72.14.236.223)  57.000 ms 74.125.252.254 (74.125.252.254)  53.388 ms 209.85.242.109
(209.85.242.109)  52.191 ms
15  209.85.245.135 (209.85.245.135)  52.148 ms 66.249.95.23 (66.249.95.23)  58.542 ms 74.125.37.235 (74
.125.37.235)  51.360 ms
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * 172.217.194.113 (172.217.194.113)  52.527 ms *
```

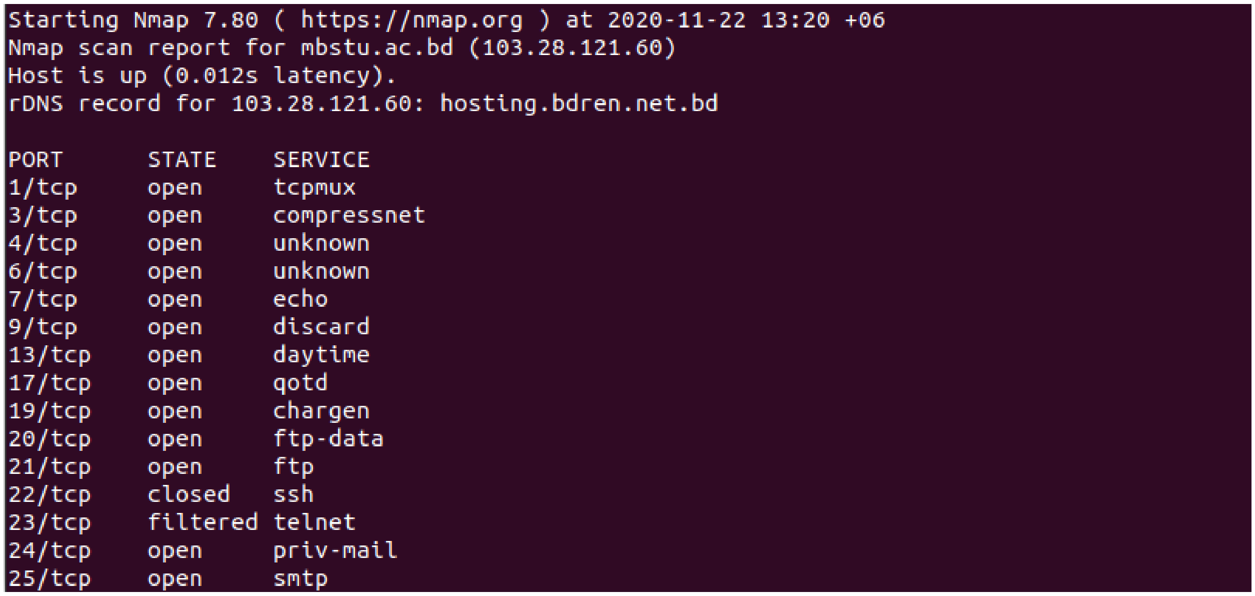
3.nmap :

Nmap ("Network Mapper") is a free and open source (license) utility for network discovery and security auditing. Many systems and network administrators also find it useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. It was designed to rapidly scan large networks, but works fine against single hosts. Nmap runs on all major computer operating systems, and official binary packages are available for Linux, Windows, and Mac OS X. In addition to the classic command-line Nmap executable, the Nmap suite includes an advanced GUI and results viewer (Zenmap), a flexible data transfer, redirection, and debugging tool (Ncat), a utility for comparing scan results (Ndiff), and a packet generation and response analysis tool (Nping).

Nmap was named "Security Product of the Year" by Linux Journal, Info World, LinuxQuestions.Org, and Codetalker Digest. It was even featured in twelve movies, including The Matrix Reloaded, Die Hard 4, Girl With the Dragon Tattoo, and The Bourne Ultimatum.



```
[MahfuzaICT@webminal.org ~]$ nmap mbstu.ac.bd
```



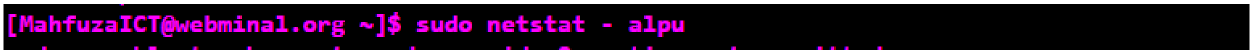
```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-22 13:20 +06
Nmap scan report for mbstu.ac.bd (103.28.121.60)
Host is up (0.012s latency).
rDNS record for 103.28.121.60: hosting.bdren.net.bd

PORT     STATE    SERVICE
1/tcp    open     tcpmux
3/tcp    open     compressnet
4/tcp    open     unknown
6/tcp    open     unknown
7/tcp    open     echo
9/tcp    open     discard
13/tcp   open     daytime
17/tcp   open     qotd
19/tcp   open     chargen
20/tcp   open     ftp-data
21/tcp   open     ftp
22/tcp   closed   ssh
23/tcp   filtered telnet
24/tcp   open     priv-mail
25/tcp   open     smtp
```

4.sudo netstat :

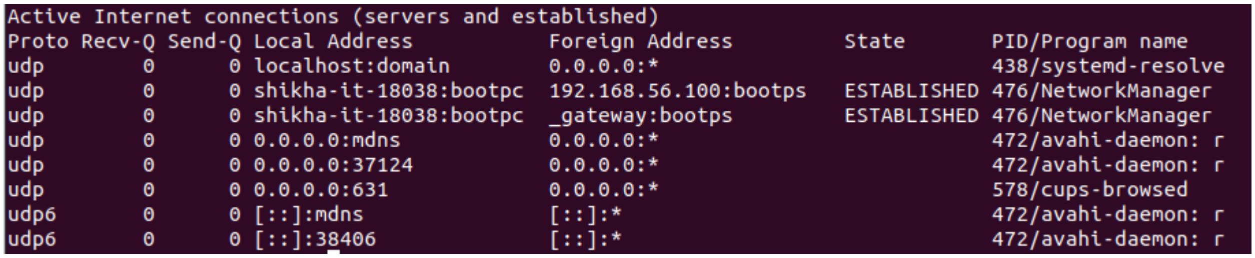
netstat (network statistics) is a command line tool for monitoring network connections both incoming and outgoing as well as viewing routing tables, interface statistics etc.

netstat is available on all Unix-like Operating Systems and also available on Windows OS as well. It is very useful in terms of network troubleshooting and performance measurement. netstat is one of the most basic network service debugging tools, telling you what ports are open and whether any programs are listening on ports.

This tool is very important and much useful for Linux network administrators as well as system administrators to monitor and troubleshoot their network-related problems and determine network traffic performance. This article shows usages of netstat command with their examples which may be useful in daily operation.



```
[MahfuzaICT@webminal.org ~]$ sudo netstat - alpu
```



```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
udp        0      0 localhost:domain        0.0.0.0:*                           438/systemd-resolve
udp        0      0 shikha-it-18038:bootpc  192.168.56.100:bootps   ESTABLISHED 476/NetworkManager
udp        0      0 shikha-it-18038:bootpc  _gateway:bootps         ESTABLISHED 476/NetworkManager
udp        0      0 0.0.0.0:mdns            0.0.0.0:*                           472/avahi-daemon: r
udp        0      0 0.0.0.0:37124           0.0.0.0:*                           472/avahi-daemon: r
udp        0      0 0.0.0.0:631             0.0.0.0:*                           578/cups-browsed
udp6       0      0 [::]:mdns               [::]:*                              472/avahi-daemon: r
udp6       0      0 [::]:38406              [::]:*                              472/avahi-daemon: r
```

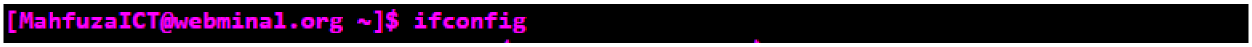
5.ifconfig :

ifconfig(interface configuration) command is used to configure the kernel-resident network interfaces. It is used at the boot time to set up the interfaces as necessary. After that, it is usually used when needed during debugging or when you need system tuning. Also, this command is used to assign the IP address and netmask to an interface or to enable or disable a given interface.

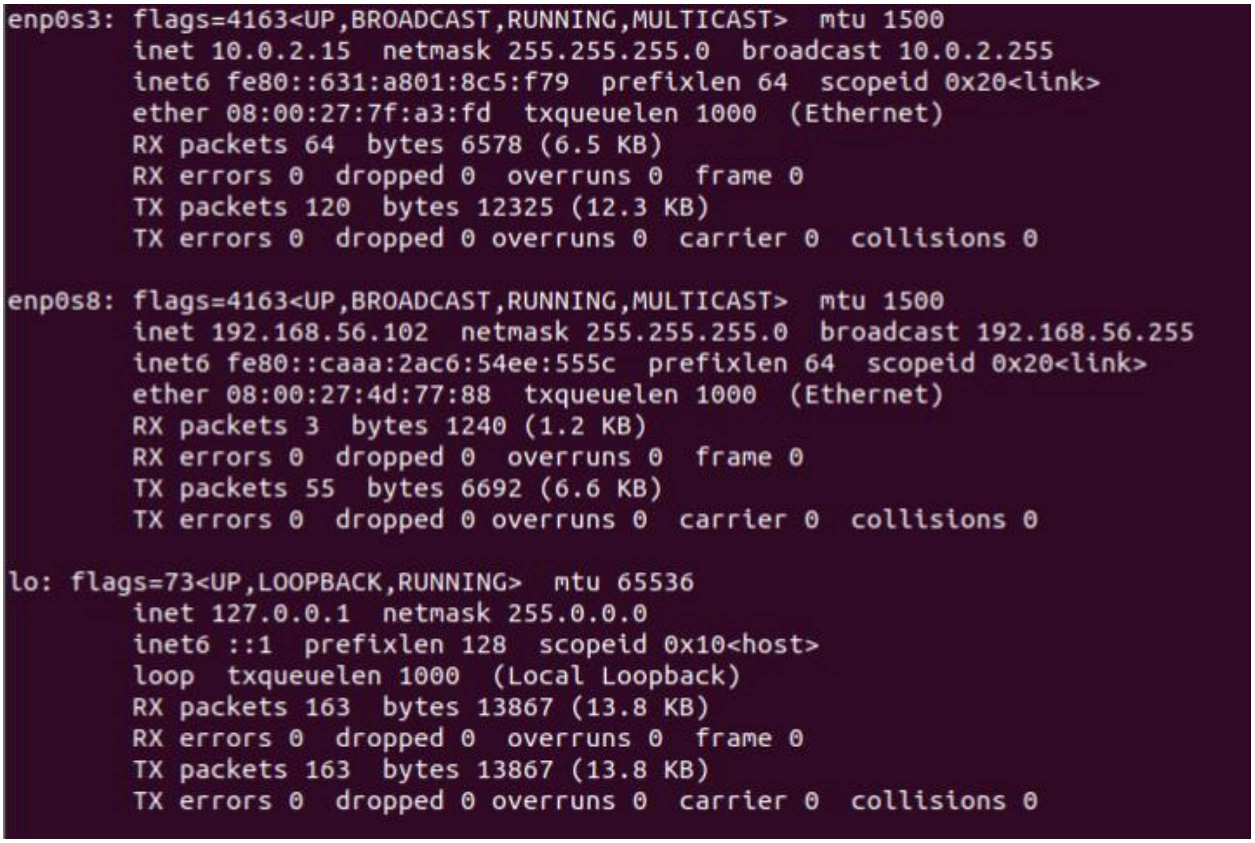
Syntax:

ifconfig [...OPTIONS] [INTERFACE]

Newer versions of some Linux distributions don’t have ifconfig command pre-installed. So, in case, there is an error **“**ifconfig: command not found**”**, Then execute the following command to install ifconfig.



```
[MahfuzaICT@webminal.org ~]$ ifconfig
```



```
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::631:a801:8c5:f79  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:7f:a3:fd  txqueuelen 1000  (Ethernet)
        RX packets 64  bytes 6578 (6.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 120  bytes 12325 (12.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.102  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::caaa:2ac6:54ee:555c  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:4d:77:88  txqueuelen 1000  (Ethernet)
        RX packets 3  bytes 1240 (1.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 55  bytes 6692 (6.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 163  bytes 13867 (13.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 163  bytes 13867 (13.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

<u>**6.dig**</u> :

People use the Linux dig command to query Domain Name System (DNS) servers. dig is an acronym for Domain Information Groper. With dig, you can query DNS servers for information regarding various DNS records, including host addresses, mail exchanges, name servers, and related information. It was intended to be a tool for diagnosing DNS issues. However, you can use it to poke around and learn more about DNS, which is one of the central systems that keep the internet routing traffic.

The internet uses internet protocol (IP) addresses to identify “locations” around the web, but people use domain names. When you type a domain name into an application, like a web browser or SSH client, something has to translate from the domain name to the actual IP address. This is where the Domain Name System comes in.

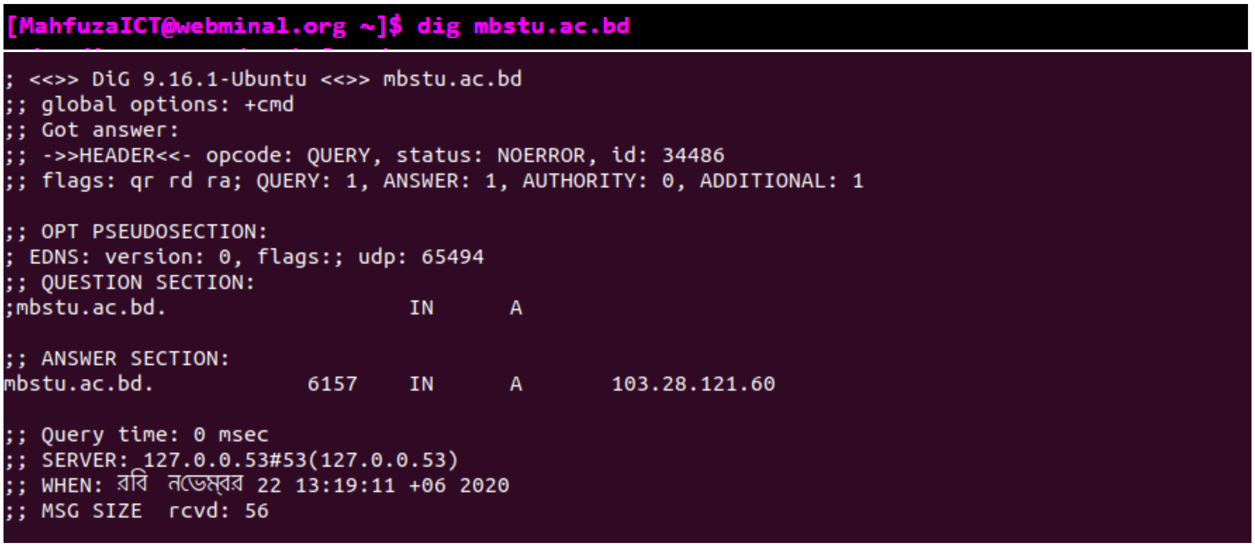
When you use a domain name with any internet-connected program, your local router can’t resolve it (unless it’s cached from a previous request). So, your router queries either your Internet Service Provider’s (ISP) DNS server, or any other you’ve configured your system to use. These are called DNS precursor servers.

If the DNS server recently received the same request from someone else on the same computer, the answer might be in *its* cache. If that’s the case, it simply sends that same information back to your program.

If the DNS precursor server can’t locate the domain in its cache, it contacts a DNS root name server. A root server won’t hold the information required to resolve domain names to IP addresses, but it will hold lists of servers that can help with your request.

The root server looks at the top-level domain to which your domain name belongs, such as .COM, .ORG, .CO.UK, and so on. It then sends a list of the top-level domain servers that handle those types of domains back to the DNS precursor server. The DNS precursor server can then make its request once more, to a top-level domain server.

The top-level domain server sends the details of the authoritative name server (where the details of the domain are stored) back to the DNS precursor server. The DNS server then queries the authoritative name server that’s hosting the zone of the domain you originally entered into your program. The authoritative name server sends the IP address back to the DNS server, which, in turn, sends it back to you.



```
[MahfuzaICT@webminal.org ~]$ dig mbstu.ac.bd
```

```
; <<>> DiG 9.16.1-Ubuntu <<>> mbstu.ac.bd
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34486
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;mbstu.ac.bd.                   IN      A

;; ANSWER SECTION:
mbstu.ac.bd.            6157    IN      A       103.28.121.60

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: রবি  নভেম্বর 22 13:19:11 +06 2020
;; MSG SIZE  rcvd: 56
```

7.curl :

curl is a tool to transfer data from or to a server, using one of the supported protocols (DICT, FILE, FTP, FTPS, GOPHER, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, MQTT, POP3, POP3S, RTMP, RTMPS, RTSP, SCP, SFTP, SMB, SMBS, SMTP, SMTPS, TELNET and TFTP). The command is designed to work without user interaction.

curl offers a busload of useful tricks like proxy support, user authentication, FTP upload, HTTP post, SSL connections, cookies, file transfer resume, Metalink, and more. As you will see below, the number of features will make your head spin!

curl is powered by libcurl for all transfer-related features. See *libcurl(3)* for details.

The URL syntax is protocol-dependent. You'll find a detailed description in RFC 3986.

You can specify multiple URLs or parts of URLs by writing part sets within braces and quoting the URL as in:

"http://site.{one,two,three}.com"

or you can get sequences of alphanumeric series by using [] as in:

"ftp://ftp.example.com/file[1-100].txt"

"ftp://ftp.example.com/file[001-100].txt"(with leading zeros)

"ftp://ftp.example.com/file[a-z].txt"

Nested sequences are not supported, but you can use several ones next to each other:

"http://example.com/archive[1996-1999]/vol[1-4]/part{a,b,c}.html"

You can specify any amount of URLs on the command line. They will be fetched in a sequential manner in the specified order. You can specify command line options and URLs mixed and in any order on the command line.

You can specify a step counter for the ranges to get every Nth number or letter:

"http://example.com/file[1-100:10].txt"

"http://example.com/file[a-z:2].txt"

When using [] or {} sequences when invoked from a command line prompt, you probably have to put the full URL within double quotes to avoid the shell from interfering with it. This also goes for other characters treated special, like for example '&', '?' and '*'.

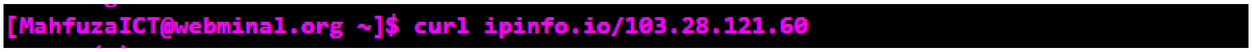
Provide the IPv6 zone index in the URL with an escaped percentage sign and the interface name. Like in

"http://[fe80::3%25eth0]/"

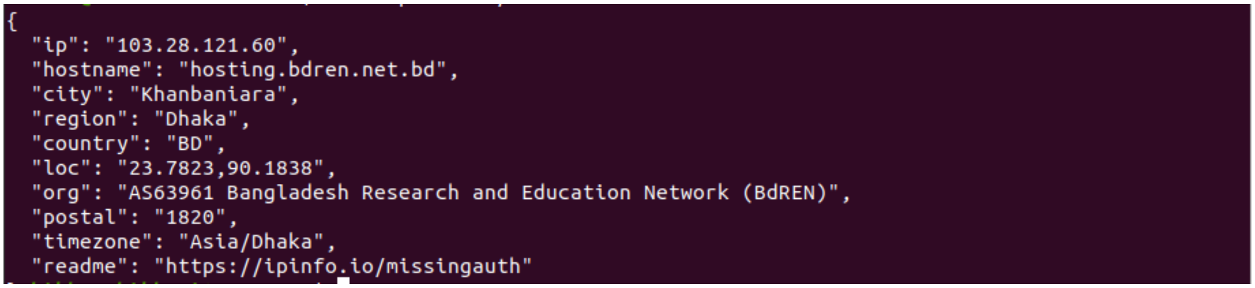
If you specify URL without protocol:// prefix, curl will attempt to guess what protocol you might want. It will then default to HTTP but try other protocols based on often-used host name prefixes. For example, for host names starting with "ftp." curl will assume you want to speak FTP.

curl will do its best to use what you pass to it as a URL. It is not trying to validate it as a syntactically correct URL by any means but is instead very liberal with what it accepts.

curl will attempt to re-use connections for multiple file transfers, so that getting many files from the same server will not do multiple connects / handshakes. This improves speed. Of course this is only done on files specified on a single command line and cannot be used between separate curl invokes.



```
[MahfuzaICT@webminal.org ~]$ curl ipinfo.io/103.28.121.60
```



```
{
  "ip": "103.28.121.60",
  "hostname": "hosting.bdren.net.bd",
  "city": "Khanbaniara",
  "region": "Dhaka",
  "country": "BD",
  "loc": "23.7823,90.1838",
  "org": "AS63961 Bangladesh Research and Education Network (BdREN)",
  "postal": "1820",
  "timezone": "Asia/Dhaka",
  "readme": "https://ipinfo.io/missingauth"
```

8. whois :

The whois system is a listing of records that contains details about both the ownership of domains and the owners. The Internet Corporation for Assigned Names and Number (ICANN) regulates domain name registration and ownership, but the list of records is held by many companies, known as registries.

Anyone can query the list of records. When you do, one of the registries will handle your request and send you details from the appropriate whois record.

Before we go any further, it’s important that you’re familiar with the following terms:

Registry: A company that manages a list containing a set of domain names (there are many of these).

Registrant: The legal owner of the domain; it’s registered to this person.

Registrar: A registrant uses a registrar to make his or her registration.

A whois record contains all the contact information associated with the person, company, or other entity that registered the domain name. Some registrations contain more information than others, and some registries return differing amounts of information.

A typical whois record will contain the following information:

The name and contact information of the registrant: The owner of the domain.

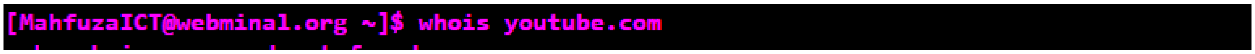
The name and contact information of the registrar: The organization that registered the domain name.

The registration date.

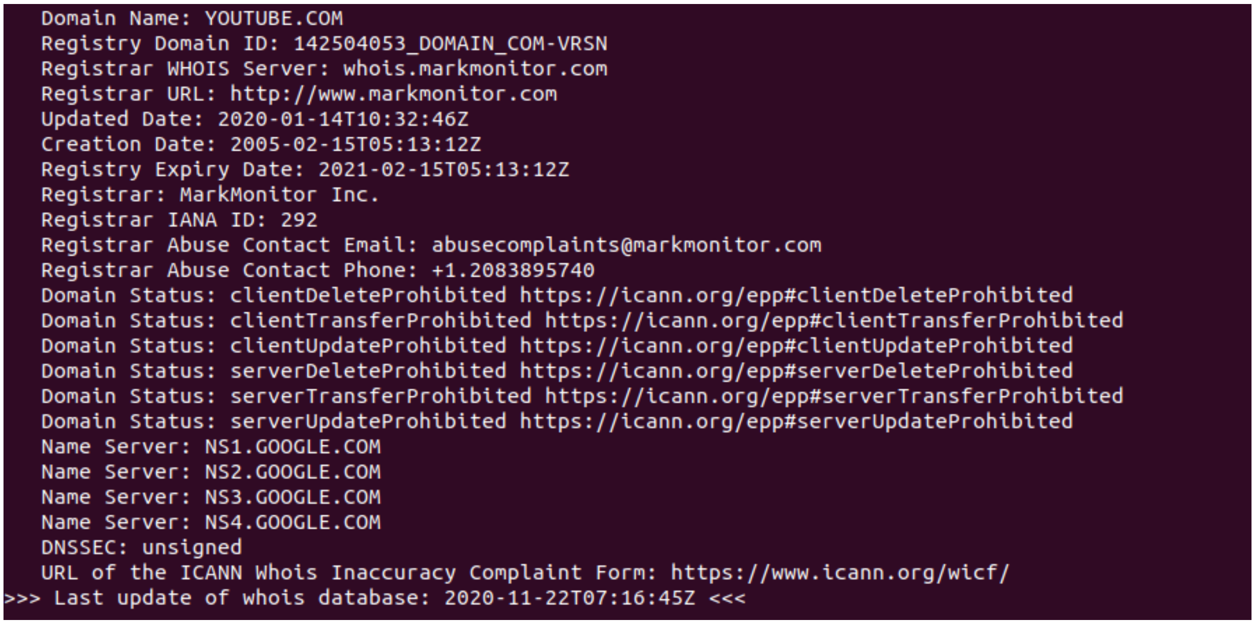
When the information was last updated.

The expiration date.

You can make whois requests on the web, but, with the Linux whois command, you can perform lookups right from the command line. This is useful if you need to perform a lookup from a computer without a graphical user interface, or if you want to do so from a shell script.



```
[MahfuzaICT@webminal.org ~]$ whois youtube.com
```



```
   Domain Name: YOUTUBE.COM
   Registry Domain ID: 142504053_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.markmonitor.com
   Registrar URL: http://www.markmonitor.com
   Updated Date: 2020-01-14T10:32:46Z
   Creation Date: 2005-02-15T05:13:12Z
   Registry Expiry Date: 2021-02-15T05:13:12Z
   Registrar: MarkMonitor Inc.
   Registrar IANA ID: 292
   Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
   Registrar Abuse Contact Phone: +1.2083895740
   Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
   Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
   Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
   Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
   Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
   Name Server: NS1.GOOGLE.COM
   Name Server: NS2.GOOGLE.COM
   Name Server: NS3.GOOGLE.COM
   Name Server: NS4.GOOGLE.COM
   DNSSEC: unsigned
   URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2020-11-22T07:16:45Z <<<
```

9. Ip :

The ip command is a powerful tool for configuring network interfaces that any Linux system administrator should know. It is used to bring interfaces up or down, assign and remove addresses and routes, manage ARP cache, and much more.

This article explains how to use the ip command through practical examples and detailed explanations of the most common options.

ip command in Linux is present in the net-tools which is used for performing several network administration tasks. IP stands for Internet Protocol. This command is used to show or manipulate routing, devices, and tunnels. It is similar to ifconfig command but it is much more powerful with more functions and facilities attached to it. ifconfig is one of the deprecated commands in the net-tools of Linux that has not been maintained for many years. ip command is used to perform several tasks like assigning an address to a network interface or configuring

network interface parameters.
It can perform several other tasks like configuring and modifying the default and static routing, setting up tunnel over IP, listing IP addresses and property information, modifying the status of the interface, assigning, deleting and setting up IP addresses and routes.

Syntax:

ip [OPTIONS] OBJECT { COMMAND | help }

With the ip command, you can adjust the way a Linux computer handles IP addresses, network interfaces controllers (NICs), and routing rules. The changes also take immediate effect—you don't have to reboot. The ip command can do a lot more than this, but we'll focus on the most common uses in this article.

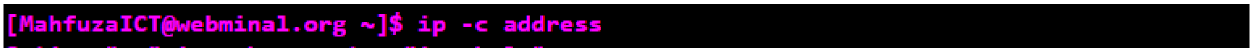
The ip command has many subcommands, each of which works on a type of object, such as IP addresses and routes. There are, in turn, many options for each of these objects. It's this richness of functionality that gives the ip command the granularity you need to perform what can be delicate tasks. This isn't ax work—it calls for a set of scalpels.

We'll look at the following objects:

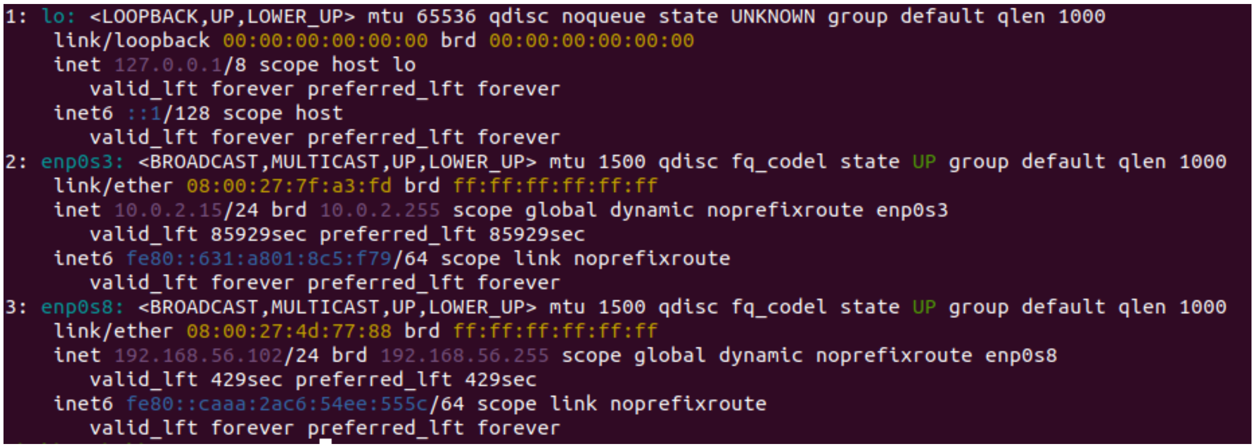
Address: IP addresses and ranges.

Link: Network interfaces, such as wired connections and Wi-Fi adapters.

Route: The rules that manage the routing of traffic sent to addresses via interfaces (links).



```
[MahfuzaICT@webminal.org ~]$ ip -c address
```



```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:7f:a3:fd brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 85929sec preferred_lft 85929sec
    inet6 fe80::631:a801:8c5:f79/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:4d:77:88 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.102/24 brd 192.168.56.255 scope global dynamic noprefixroute enp0s8
       valid_lft 429sec preferred_lft 429sec
    inet6 fe80::caaa:2ac6:54ee:555c/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

10. rsync :

rsync or remote synchronization is a software utility for Unix-Like systems that efficiently sync files and directories between two hosts or machines. One of them being the source or the local-host from which the files will be synced, the other one being the remote-host, on which synchronization will take place. There are basically two ways in which rsync can copy/sync data:

Copying/syncing to/from another host over any remote shell like ssh, rsh.

Copying/Syncing through rsync daemon using TCP.

Rsync is famous for its delta-transfer algorithm, in which it copies only the differences between the source files present in the local-host and the existing files in the destination or the remote host.

Example:

```
rsync local-file user@remote-host:remote-file
```

What Happens here: Rsync will first use SSH to connect as user to remote-host and will ask for user's password. Once connected, it will invoke the remote host’s rsync and then the two programs will determine what parts of the local-file needs to be copied so that the remote file matches the local one. Please note the following behavior of rsync:

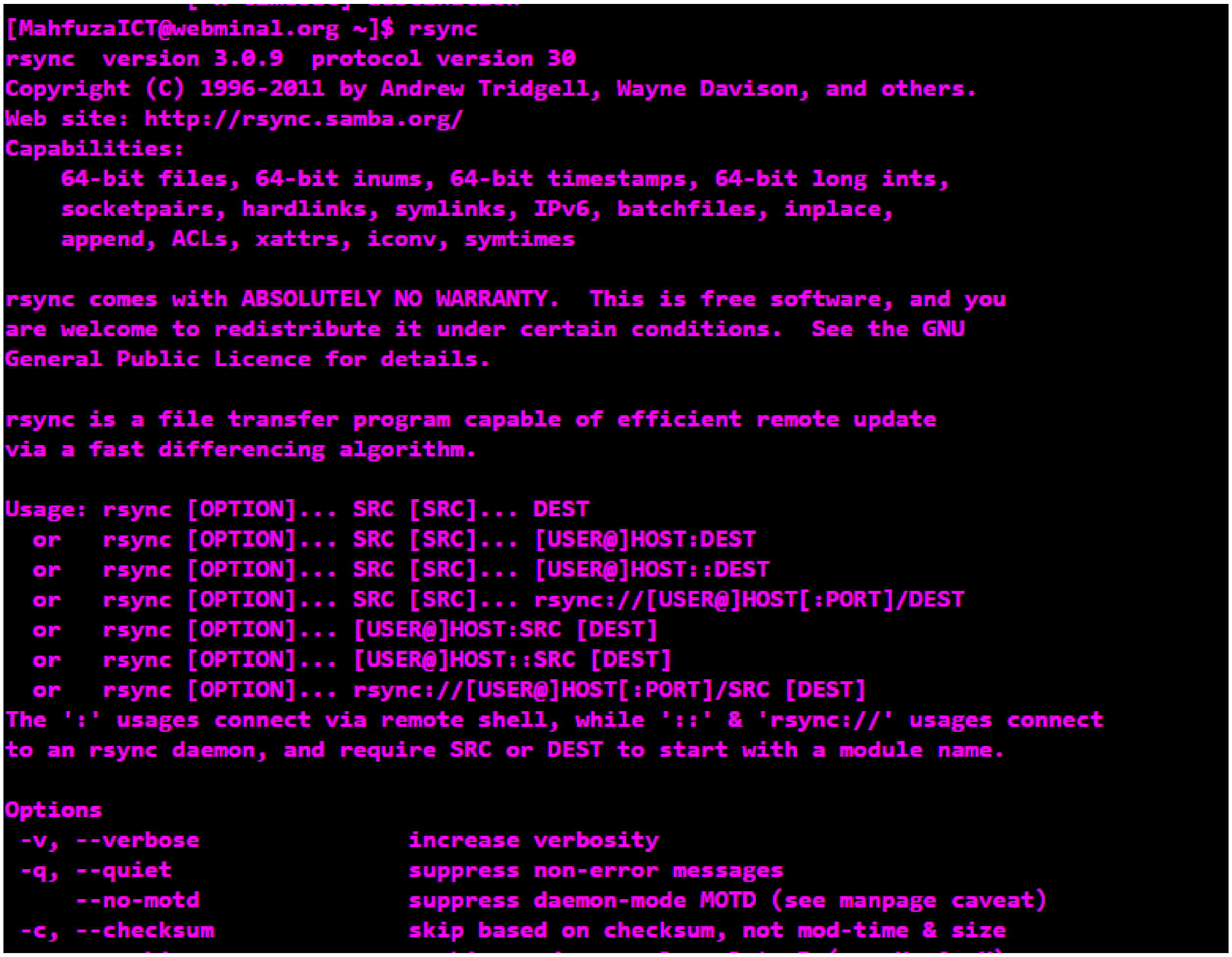
Files that do not exist on the remote-host are copied.

Files that have been updated will be synced, rsync will copy only the changed parts of files to the remote host.

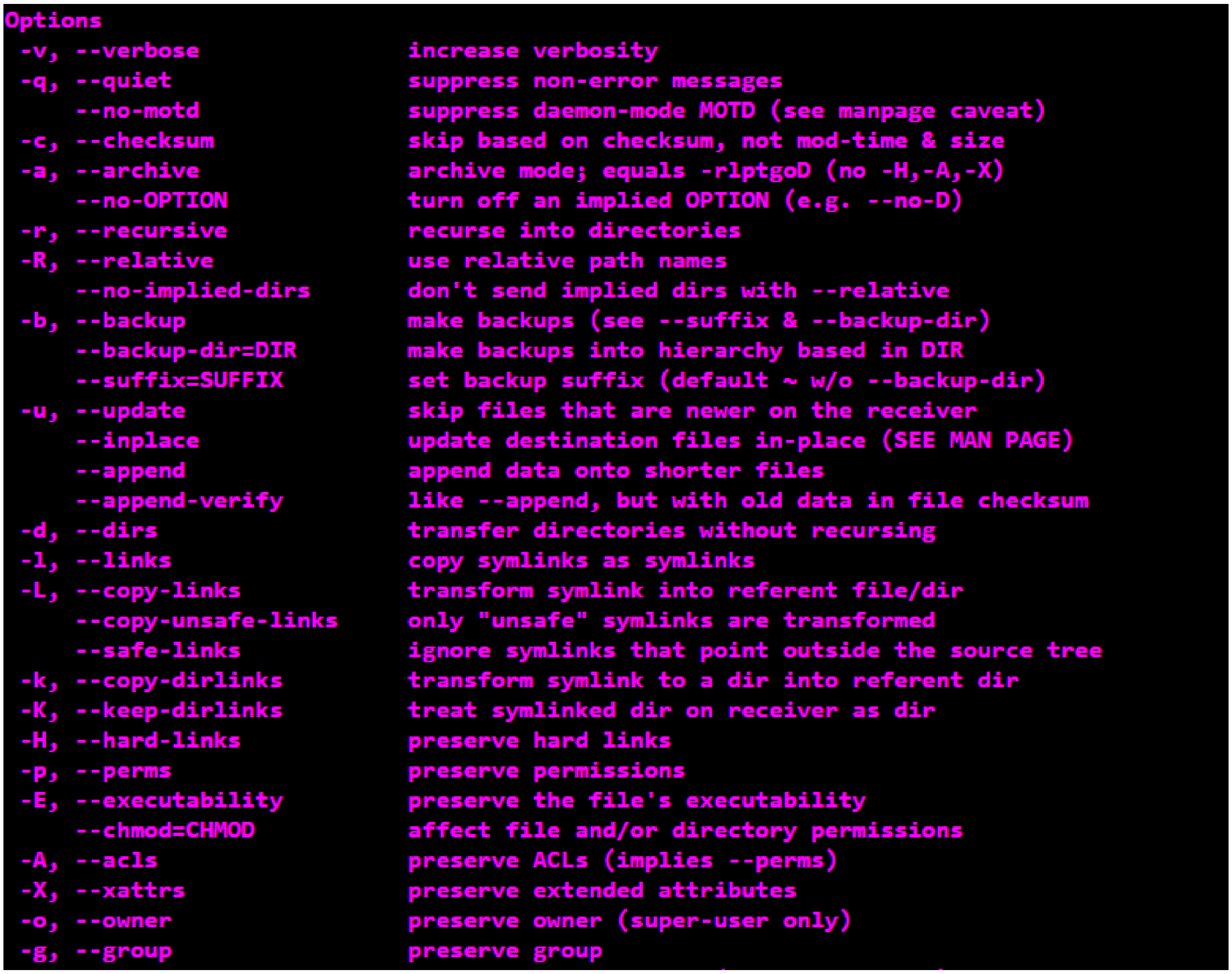
File that is exactly the same are not copied to the remote host at all.

Syntax of rsync:

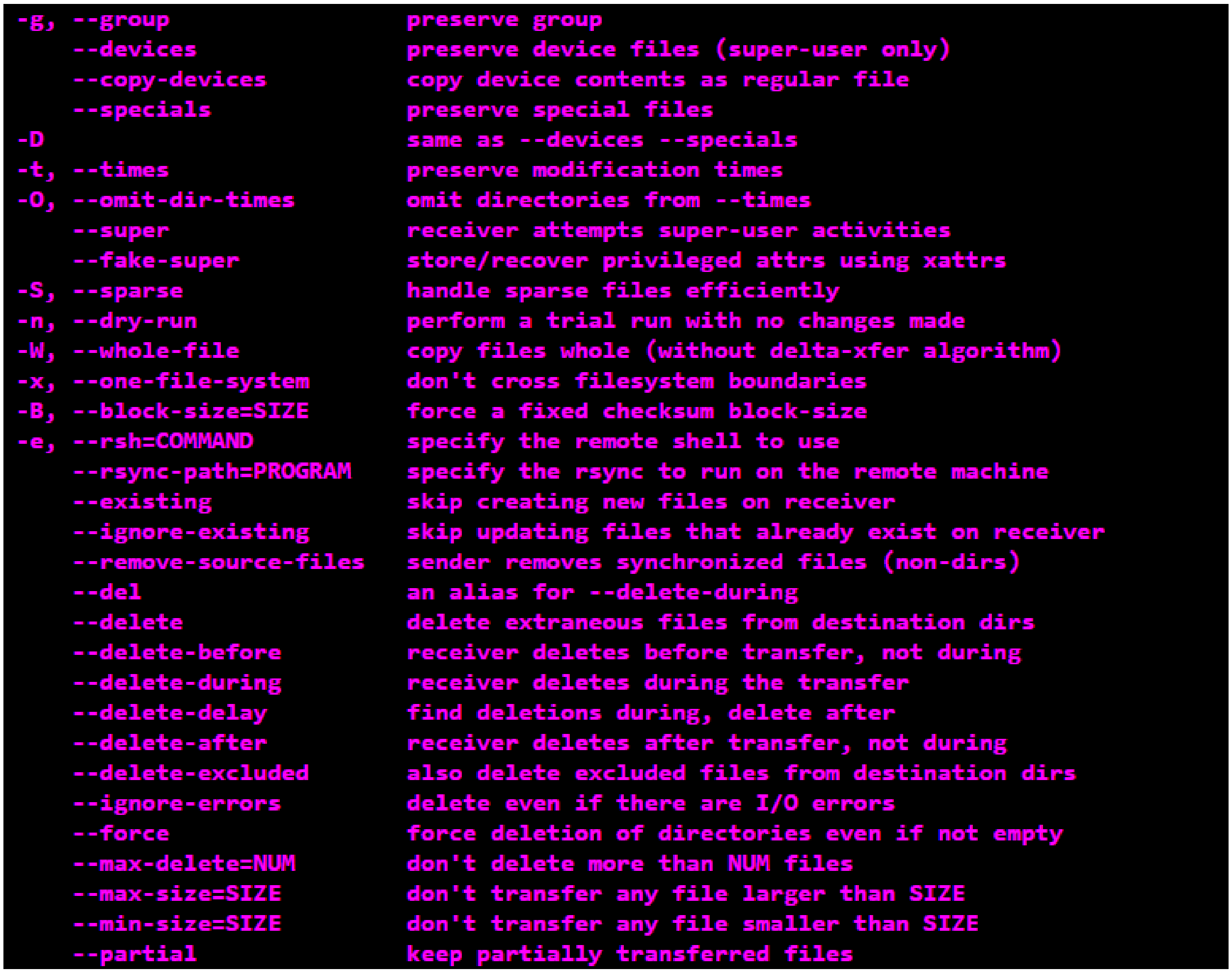
```
rsync [options] source [destination]
```



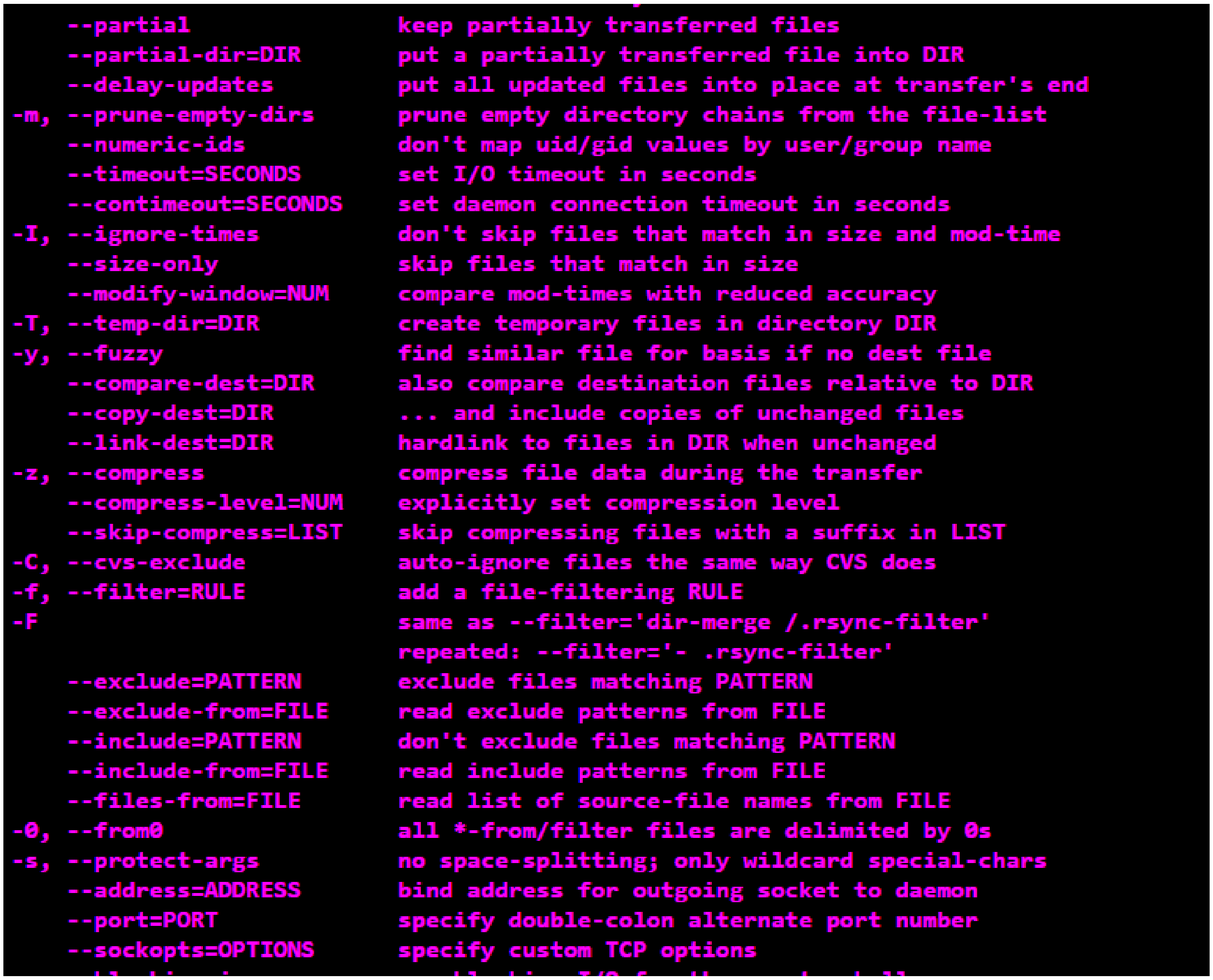
```
[MahfuzaICT@webminal.org ~]$ rsync
rsync  version 3.0.9  protocol version 30
Copyright (C) 1996-2011 by Andrew Tridgell, Wayne Davison, and others.
Web site: http://rsync.samba.org/
Capabilities:
    64-bit files, 64-bit inums, 64-bit timestamps, 64-bit long ints,
    socketpairs, hardlinks, symlinks, IPv6, batchfiles, inplace,
    append, ACLs, xattrs, iconv, symtimes

rsync comes with ABSOLUTELY NO WARRANTY.  This is free software, and you
are welcome to redistribute it under certain conditions.  See the GNU
General Public Licence for details.

rsync is a file transfer program capable of efficient remote update
via a fast differencing algorithm.

Usage: rsync [OPTION]... SRC [SRC]... DEST
  or   rsync [OPTION]... SRC [SRC]... [USER@]HOST:DEST
  or   rsync [OPTION]... SRC [SRC]... [USER@]HOST::DEST
  or   rsync [OPTION]... SRC [SRC]... rsync://[USER@]HOST[:PORT]/DEST
  or   rsync [OPTION]... [USER@]HOST:SRC [DEST]
  or   rsync [OPTION]... [USER@]HOST::SRC [DEST]
  or   rsync [OPTION]... rsync://[USER@]HOST[:PORT]/SRC [DEST]
The ':' usages connect via remote shell, while '::' & 'rsync://' usages connect
to an rsync daemon, and require SRC or DEST to start with a module name.

Options
 -v, --verbose               increase verbosity
 -q, --quiet                 suppress non-error messages
     --no-motd               suppress daemon-mode MOTD (see manpage caveat)
 -c, --checksum              skip based on checksum, not mod-time & size
```



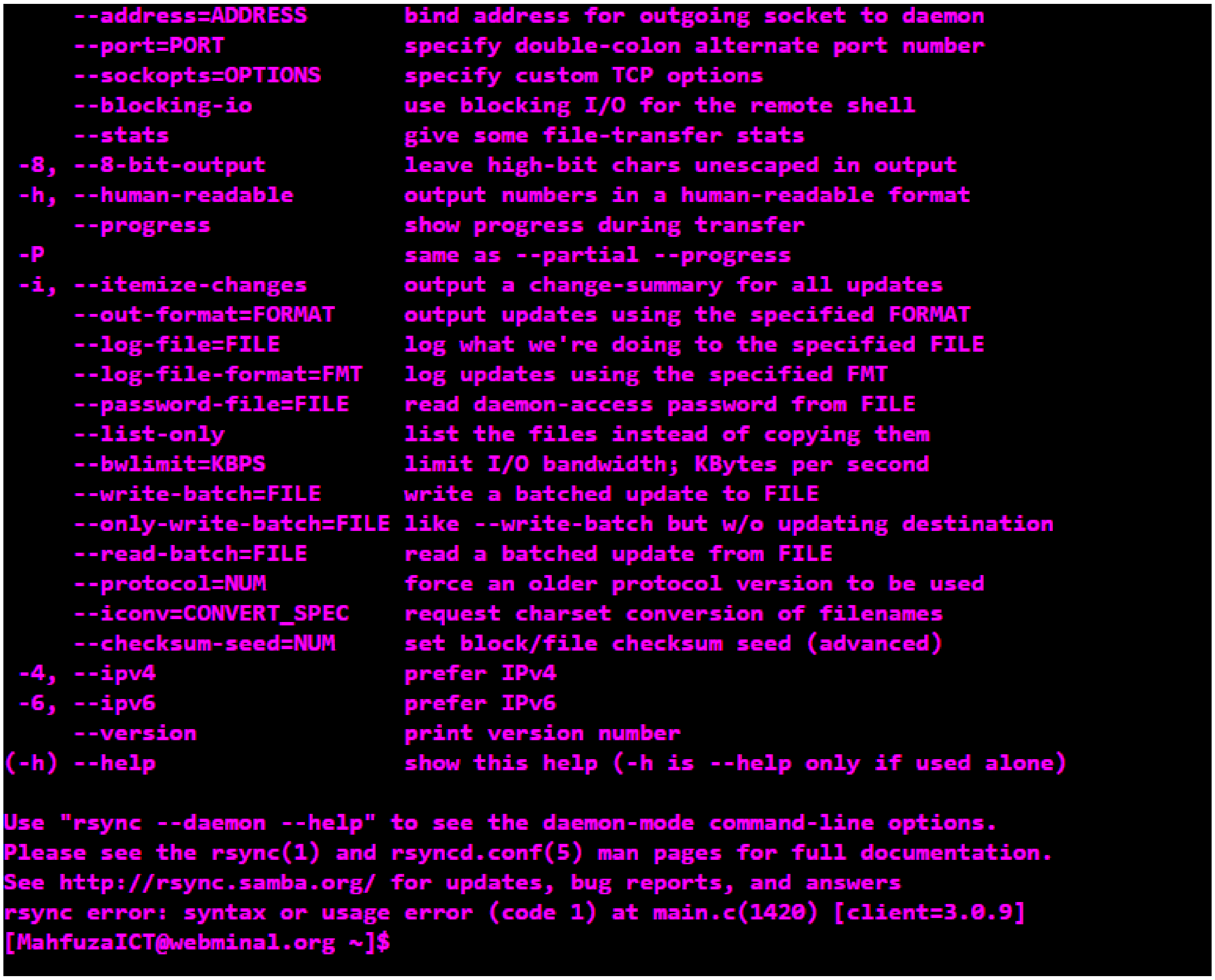
```
Options
 -v, --verbose               increase verbosity
 -q, --quiet                 suppress non-error messages
     --no-motd               suppress daemon-mode MOTD (see manpage caveat)
 -c, --checksum              skip based on checksum, not mod-time & size
 -a, --archive               archive mode; equals -rlptgoD (no -H,-A,-X)
     --no-OPTION             turn off an implied OPTION (e.g. --no-D)
 -r, --recursive             recurse into directories
 -R, --relative              use relative path names
     --no-implied-dirs       don't send implied dirs with --relative
 -b, --backup                make backups (see --suffix & --backup-dir)
     --backup-dir=DIR        make backups into hierarchy based in DIR
     --suffix=SUFFIX         set backup suffix (default ~ w/o --backup-dir)
 -u, --update                skip files that are newer on the receiver
     --inplace               update destination files in-place (SEE MAN PAGE)
     --append                append data onto shorter files
     --append-verify         like --append, but with old data in file checksum
 -d, --dirs                  transfer directories without recursing
 -l, --links                 copy symlinks as symlinks
 -L, --copy-links            transform symlink into referent file/dir
     --copy-unsafe-links     only "unsafe" symlinks are transformed
     --safe-links            ignore symlinks that point outside the source tree
 -k, --copy-dirlinks         transform symlink to a dir into referent dir
 -K, --keep-dirlinks         treat symlinked dir on receiver as dir
 -H, --hard-links            preserve hard links
 -p, --perms                 preserve permissions
 -E, --executability         preserve the file's executability
     --chmod=CHMOD           affect file and/or directory permissions
 -A, --acls                  preserve ACLs (implies --perms)
 -X, --xattrs                preserve extended attributes
 -o, --owner                 preserve owner (super-user only)
 -g, --group                 preserve group
```



```
-g, --group                 preserve group
    --devices               preserve device files (super-user only)
    --copy-devices          copy device contents as regular file
    --specials              preserve special files
-D                          same as --devices --specials
-t, --times                 preserve modification times
-O, --omit-dir-times        omit directories from --times
    --super                 receiver attempts super-user activities
    --fake-super            store/recover privileged attrs using xattrs
-S, --sparse                handle sparse files efficiently
-n, --dry-run               perform a trial run with no changes made
-W, --whole-file            copy files whole (without delta-xfer algorithm)
-x, --one-file-system       don't cross filesystem boundaries
-B, --block-size=SIZE       force a fixed checksum block-size
-e, --rsh=COMMAND           specify the remote shell to use
    --rsync-path=PROGRAM    specify the rsync to run on the remote machine
    --existing              skip creating new files on receiver
    --ignore-existing       skip updating files that already exist on receiver
    --remove-source-files   sender removes synchronized files (non-dirs)
    --del                   an alias for --delete-during
    --delete                delete extraneous files from destination dirs
    --delete-before         receiver deletes before transfer, not during
    --delete-during         receiver deletes during the transfer
    --delete-delay          find deletions during, delete after
    --delete-after          receiver deletes after transfer, not during
    --delete-excluded       also delete excluded files from destination dirs
    --ignore-errors         delete even if there are I/O errors
    --force                 force deletion of directories even if not empty
    --max-delete=NUM        don't delete more than NUM files
    --max-size=SIZE         don't transfer any file larger than SIZE
    --min-size=SIZE         don't transfer any file smaller than SIZE
    --partial               keep partially transferred files
```



```
    --partial               keep partially transferred files
    --partial-dir=DIR       put a partially transferred file into DIR
    --delay-updates         put all updated files into place at transfer's end
-m, --prune-empty-dirs      prune empty directory chains from the file-list
    --numeric-ids           don't map uid/gid values by user/group name
    --timeout=SECONDS       set I/O timeout in seconds
    --contimeout=SECONDS    set daemon connection timeout in seconds
-I, --ignore-times          don't skip files that match in size and mod-time
    --size-only             skip files that match in size
    --modify-window=NUM     compare mod-times with reduced accuracy
-T, --temp-dir=DIR          create temporary files in directory DIR
-y, --fuzzy                 find similar file for basis if no dest file
    --compare-dest=DIR      also compare destination files relative to DIR
    --copy-dest=DIR         ... and include copies of unchanged files
    --link-dest=DIR         hardlink to files in DIR when unchanged
-z, --compress              compress file data during the transfer
    --compress-level=NUM    explicitly set compression level
    --skip-compress=LIST    skip compressing files with a suffix in LIST
-C, --cvs-exclude           auto-ignore files the same way CVS does
-f, --filter=RULE           add a file-filtering RULE
-F                          same as --filter='dir-merge /.rsync-filter'
                            repeated: --filter='- .rsync-filter'
    --exclude=PATTERN       exclude files matching PATTERN
    --exclude-from=FILE     read exclude patterns from FILE
    --include=PATTERN       don't exclude files matching PATTERN
    --include-from=FILE     read include patterns from FILE
    --files-from=FILE       read list of source-file names from FILE
-0, --from0                 all *-from/filter files are delimited by 0s
-s, --protect-args          no space-splitting; only wildcard special-chars
    --address=ADDRESS       bind address for outgoing socket to daemon
    --port=PORT             specify double-colon alternate port number
    --sockopts=OPTIONS      specify custom TCP options
```



```
     --address=ADDRESS       bind address for outgoing socket to daemon
     --port=PORT             specify double-colon alternate port number
     --sockopts=OPTIONS      specify custom TCP options
     --blocking-io           use blocking I/O for the remote shell
     --stats                 give some file-transfer stats
 -8, --8-bit-output          leave high-bit chars unescaped in output
 -h, --human-readable        output numbers in a human-readable format
     --progress              show progress during transfer
 -P                          same as --partial --progress
 -i, --itemize-changes       output a change-summary for all updates
     --out-format=FORMAT     output updates using the specified FORMAT
     --log-file=FILE         log what we're doing to the specified FILE
     --log-file-format=FMT   log updates using the specified FMT
     --password-file=FILE    read daemon-access password from FILE
     --list-only             list the files instead of copying them
     --bwlimit=KBPS          limit I/O bandwidth; KBytes per second
     --write-batch=FILE      write a batched update to FILE
     --only-write-batch=FILE like --write-batch but w/o updating destination
     --read-batch=FILE       read a batched update from FILE
     --protocol=NUM          force an older protocol version to be used
     --iconv=CONVERT_SPEC    request charset conversion of filenames
     --checksum-seed=NUM     set block/file checksum seed (advanced)
 -4, --ipv4                  prefer IPv4
 -6, --ipv6                  prefer IPv6
     --version               print version number
(-h) --help                  show this help (-h is --help only if used alone)

Use "rsync --daemon --help" to see the daemon-mode command-line options.
Please see the rsync(1) and rsyncd.conf(5) man pages for full documentation.
See http://rsync.samba.org/ for updates, bug reports, and answers
rsync error: syntax or usage error (code 1) at main.c(1420) [client=3.0.9]
[MahfuzaICT@webminal.org ~]$
```

11.ping :

The ping command is one of the most used tools for troubleshooting, testing, and diagnosing network connectivity issues.

Ping works by sending one or more ICMP (Internet Control Message Protocol) Echo Request packages to a specified destination IP on the network and waits for a reply. When the destination receives the package, it responds with an ICMP echo reply.

With the ping command, you can determine whether a remote destination IP is active or inactive. You can also find the round-trip delay in communicating with the destination and check whether there is a packet loss.

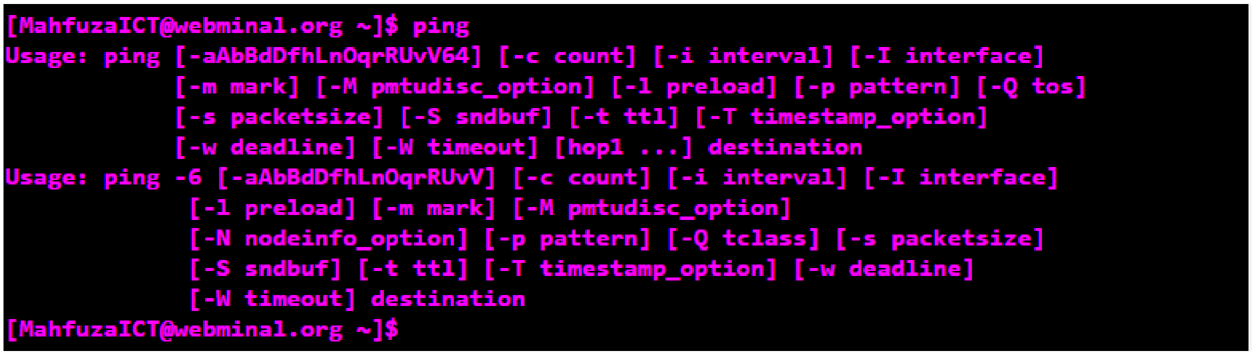
ping is part of the iputils (or iputils-ping) package, which is pre-installed on nearly all Linux distributions. It is also available on Windows, macOS, and FreeBSD.

PING (Packet Internet Groper) command is used to check the network connectivity between host and server/host. This command takes as input the IP address or the URL and sends a data packet

to the specified address with the message "PING" and get a response from the server/host this time is recorded which is called latency. Fast ping low latency means faster connection. Ping uses ICMP(Internet Control Message Protocol) to send an ICMP echo message to the specified host if that host is available then it sends ICMP reply message. Ping is generally measured in millisecond every modern operating system has this ping pre-installed.
Now let see the PING command :

PING Version:
To get ping version installed on your system.

sudo ping –v



```
[MahfuzaICT@webminal.org ~]$ ping
Usage: ping [-aAbBdDfhLnOqrRUvV64] [-c count] [-i interval] [-I interface]
            [-m mark] [-M pmtudisc_option] [-l preload] [-p pattern] [-Q tos]
            [-s packetsize] [-S sndbuf] [-t ttl] [-T timestamp_option]
            [-w deadline] [-W timeout] [hop1 ...] destination
Usage: ping -6 [-aAbBdDfhLnOqrRUvV] [-c count] [-i interval] [-I interface]
             [-l preload] [-m mark] [-M pmtudisc_option]
             [-N nodeinfo_option] [-p pattern] [-Q tclass] [-s packetsize]
             [-S sndbuf] [-t ttl] [-T timestamp_option] [-w deadline]
             [-W timeout] destination
[MahfuzaICT@webminal.org ~]$
```

<u>12.sudo apt install wireshark</u> :

Wireshark is an open-source network protocol analyzer tool indispensable for system administration and security. It drills down and displays data travelling on the network. Wireshark allows you to either capture live network packets or to save it for offline analysis.

One of the features of Wireshark that you will love to learn is the display filter which lets you inspect only that traffic you are really interested in. Wireshark is available for various platforms including Windows, Linux, MacOS, FreeBSD, and some others.

Some of the tasks one can perform with Wireshark are

Capturing and finding traffic passing through your network

Inspection of hundreds of different protocols

Live capture of traffic/offline analysis

Troubleshooting dropped packets and latency problems

Looking at attempts of attacks or malicious activities

In this article, we will explain how to install Wireshark on the Ubuntu system. The installation procedures have been tested on Ubuntu 20.04 LTS.

Note:

We have used the command line Terminal for the installation procedure. You can launch the Terminal via Ctrl+Alt+T keyboard shortcut.

You must be a root user or have sudo privileges in order to install and use Wireshark to capture data on your system.

Installing Wireshark

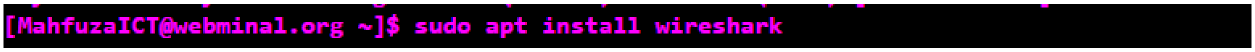
For installing Wireshark, you will need to add the “Universe” repository. Issue the following command in Terminal to do so:

$ sudo add-apt-repository universe

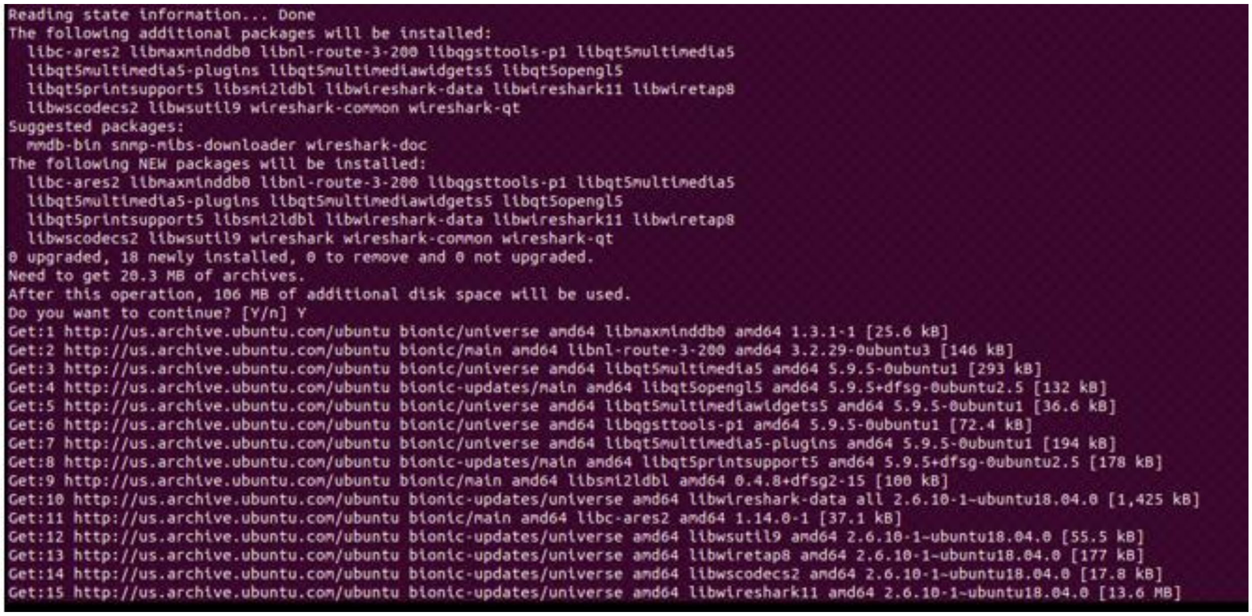
Now issue the following command in Terminal to install Wireshark on your system:

$ sudo apt install Wireshark

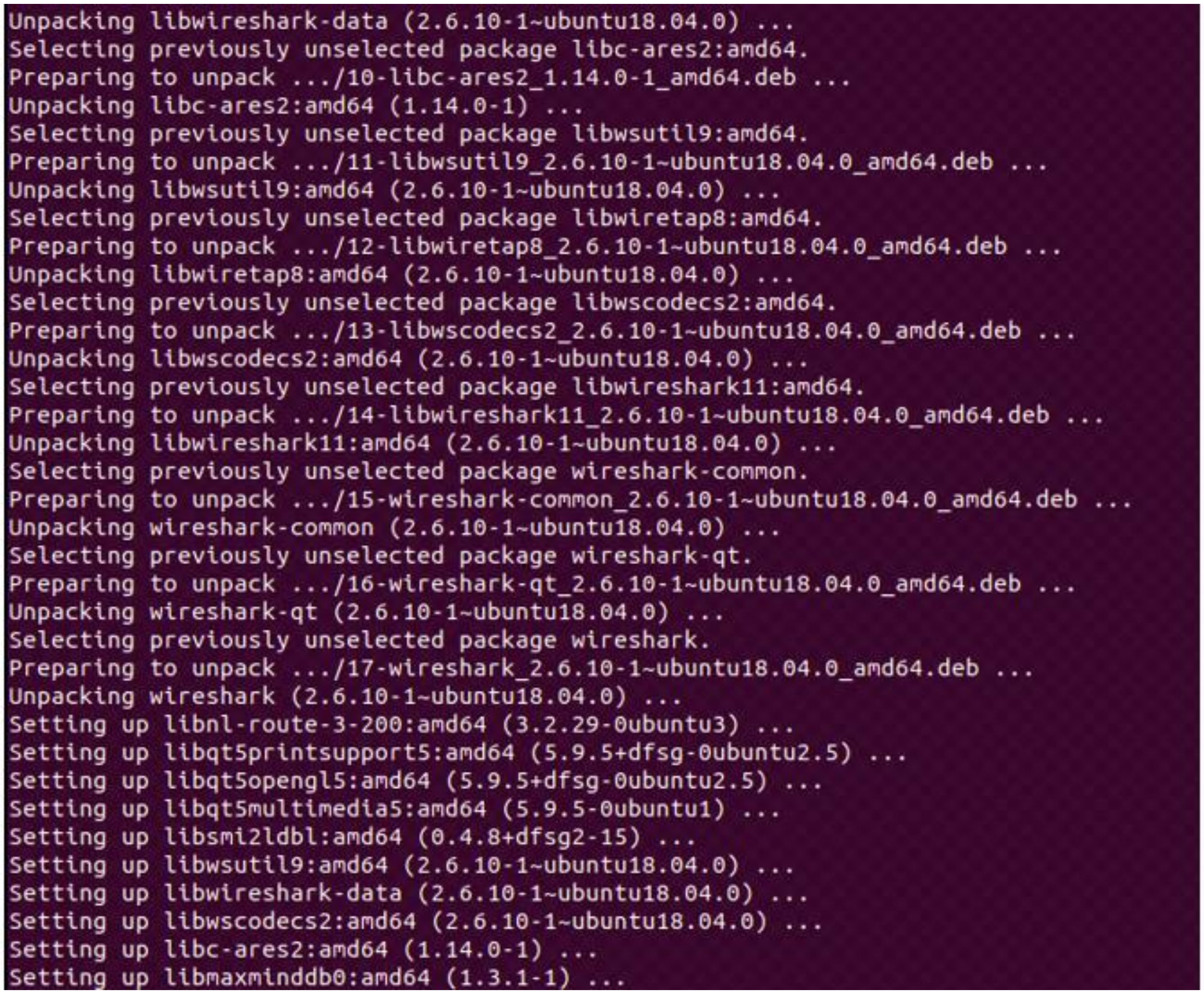
When prompted for a password, type sudo password.



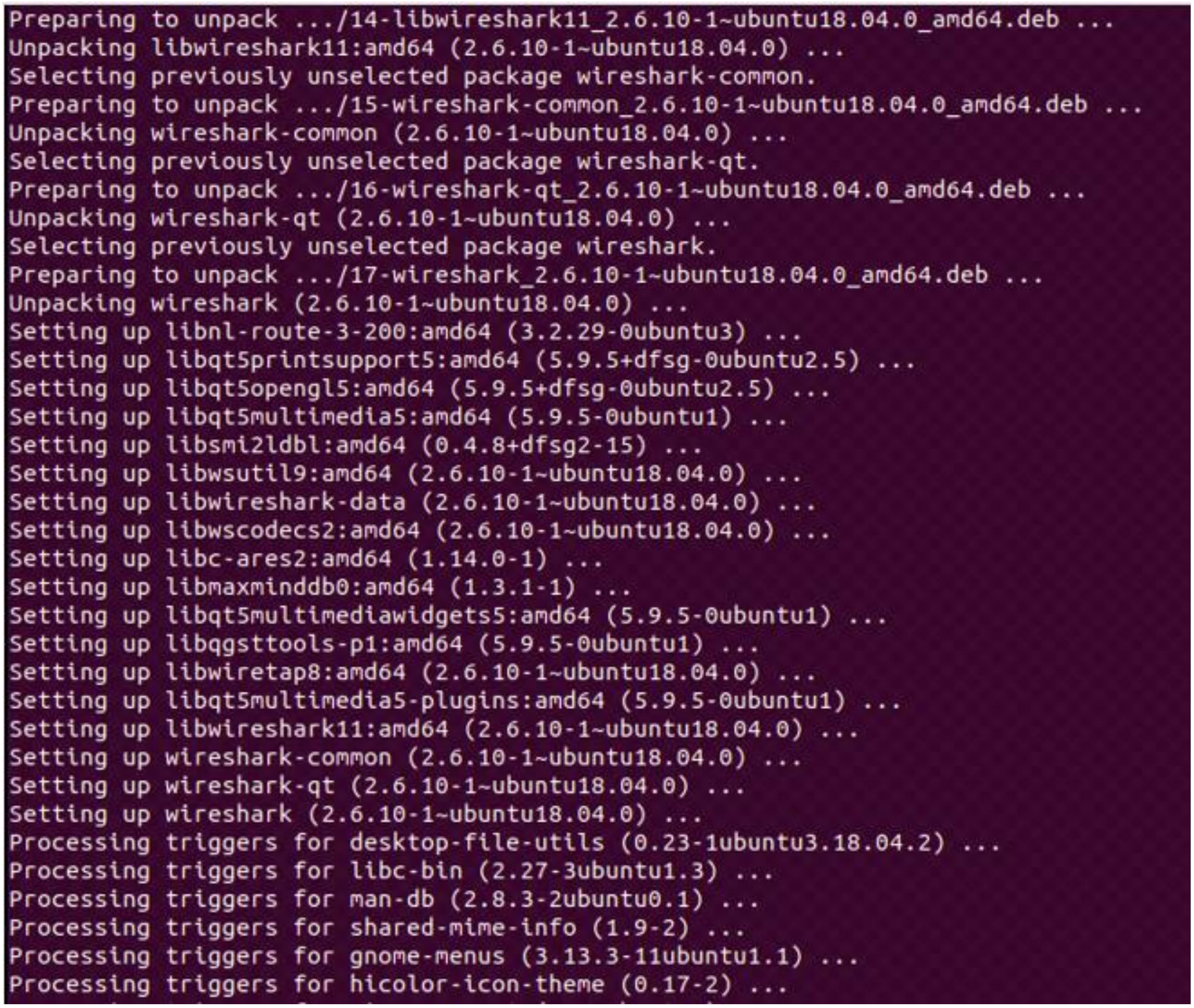
```
[MahfuzaICT@webminal.org ~]$ sudo apt install wireshark
```



```
Reading state information... Done
The following additional packages will be installed:
  libc-ares2 libmaxminddb0 libnl-route-3-200 libqgsttools-p1 libqt5multimedia5
  libqt5multimedia5-plugins libqt5multimediawidgets5 libqt5opengl5
  libqt5printsupport5 libsmi2ldbl libwireshark-data libwireshark11 libwiretap8
  libwscodecs2 libwsutil9 wireshark-common wireshark-qt
Suggested packages:
  mmdb-bin snmp-mibs-downloader wireshark-doc
The following NEW packages will be installed:
  libc-ares2 libmaxminddb0 libnl-route-3-200 libqgsttools-p1 libqt5multimedia5
  libqt5multimedia5-plugins libqt5multimediawidgets5 libqt5opengl5
  libqt5printsupport5 libsmi2ldbl libwireshark-data libwireshark11 libwiretap8
  libwscodecs2 libwsutil9 wireshark wireshark-common wireshark-qt
0 upgraded, 18 newly installed, 0 to remove and 0 not upgraded.
Need to get 20.3 MB of archives.
After this operation, 106 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu bionic/universe amd64 libmaxminddb0 amd64 1.3.1-1 [25.6 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 libnl-route-3-200 amd64 3.2.29-0ubuntu3 [146 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu bionic/universe amd64 libqt5multimedia5 amd64 5.9.5-0ubuntu1 [293 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libqt5opengl5 amd64 5.9.5+dfsg-0ubuntu2.5 [132 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu bionic/universe amd64 libqt5multimediawidgets5 amd64 5.9.5-0ubuntu1 [36.6 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu bionic/universe amd64 libqgsttools-p1 amd64 5.9.5-0ubuntu1 [72.4 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu bionic/universe amd64 libqt5multimedia5-plugins amd64 5.9.5-0ubuntu1 [194 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libqt5printsupport5 amd64 5.9.5+dfsg-0ubuntu2.5 [178 kB]
Get:9 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 libsmi2ldbl amd64 0.4.8+dfsg2-15 [100 kB]
Get:10 http://us.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 libwireshark-data all 2.6.10-1~ubuntu18.04.0 [1,425 kB]
Get:11 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 libc-ares2 amd64 1.14.0-1 [37.1 kB]
Get:12 http://us.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 libwsutil9 amd64 2.6.10-1~ubuntu18.04.0 [55.5 kB]
Get:13 http://us.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 libwiretap8 amd64 2.6.10-1~ubuntu18.04.0 [177 kB]
Get:14 http://us.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 libwscodecs2 amd64 2.6.10-1~ubuntu18.04.0 [17.8 kB]
Get:15 http://us.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 libwireshark11 amd64 2.6.10-1~ubuntu18.04.0 [13.6 MB]
```



```
Unpacking libwireshark-data (2.6.10-1~ubuntu18.04.0) ...
Selecting previously unselected package libc-ares2:amd64.
Preparing to unpack .../10-libc-ares2_1.14.0-1_amd64.deb ...
Unpacking libc-ares2:amd64 (1.14.0-1) ...
Selecting previously unselected package libwsutil9:amd64.
Preparing to unpack .../11-libwsutil9_2.6.10-1~ubuntu18.04.0_amd64.deb ...
Unpacking libwsutil9:amd64 (2.6.10-1~ubuntu18.04.0) ...
Selecting previously unselected package libwiretap8:amd64.
Preparing to unpack .../12-libwiretap8_2.6.10-1~ubuntu18.04.0_amd64.deb ...
Unpacking libwiretap8:amd64 (2.6.10-1~ubuntu18.04.0) ...
Selecting previously unselected package libwscodecs2:amd64.
Preparing to unpack .../13-libwscodecs2_2.6.10-1~ubuntu18.04.0_amd64.deb ...
Unpacking libwscodecs2:amd64 (2.6.10-1~ubuntu18.04.0) ...
Selecting previously unselected package libwireshark11:amd64.
Preparing to unpack .../14-libwireshark11_2.6.10-1~ubuntu18.04.0_amd64.deb ...
Unpacking libwireshark11:amd64 (2.6.10-1~ubuntu18.04.0) ...
Selecting previously unselected package wireshark-common.
Preparing to unpack .../15-wireshark-common_2.6.10-1~ubuntu18.04.0_amd64.deb ...
Unpacking wireshark-common (2.6.10-1~ubuntu18.04.0) ...
Selecting previously unselected package wireshark-qt.
Preparing to unpack .../16-wireshark-qt_2.6.10-1~ubuntu18.04.0_amd64.deb ...
Unpacking wireshark-qt (2.6.10-1~ubuntu18.04.0) ...
Selecting previously unselected package wireshark.
Preparing to unpack .../17-wireshark_2.6.10-1~ubuntu18.04.0_amd64.deb ...
Unpacking wireshark (2.6.10-1~ubuntu18.04.0) ...
Setting up libnl-route-3-200:amd64 (3.2.29-0ubuntu3) ...
Setting up libqt5printsupport5:amd64 (5.9.5+dfsg-0ubuntu2.5) ...
Setting up libqt5opengl5:amd64 (5.9.5+dfsg-0ubuntu2.5) ...
Setting up libqt5multimedia5:amd64 (5.9.5-0ubuntu1) ...
Setting up libsmi2ldbl:amd64 (0.4.8+dfsg2-15) ...
Setting up libwsutil9:amd64 (2.6.10-1~ubuntu18.04.0) ...
Setting up libwireshark-data (2.6.10-1~ubuntu18.04.0) ...
Setting up libwscodecs2:amd64 (2.6.10-1~ubuntu18.04.0) ...
Setting up libc-ares2:amd64 (1.14.0-1) ...
Setting up libmaxminddb0:amd64 (1.3.1-1) ...
```



```
Preparing to unpack .../14-libwireshark11_2.6.10-1~ubuntu18.04.0_amd64.deb ...
Unpacking libwireshark11:amd64 (2.6.10-1~ubuntu18.04.0) ...
Selecting previously unselected package wireshark-common.
Preparing to unpack .../15-wireshark-common_2.6.10-1~ubuntu18.04.0_amd64.deb ...
Unpacking wireshark-common (2.6.10-1~ubuntu18.04.0) ...
Selecting previously unselected package wireshark-qt.
Preparing to unpack .../16-wireshark-qt_2.6.10-1~ubuntu18.04.0_amd64.deb ...
Unpacking wireshark-qt (2.6.10-1~ubuntu18.04.0) ...
Selecting previously unselected package wireshark.
Preparing to unpack .../17-wireshark_2.6.10-1~ubuntu18.04.0_amd64.deb ...
Unpacking wireshark (2.6.10-1~ubuntu18.04.0) ...
Setting up libnl-route-3-200:amd64 (3.2.29-0ubuntu3) ...
Setting up libqt5printsupport5:amd64 (5.9.5+dfsg-0ubuntu2.5) ...
Setting up libqt5opengl5:amd64 (5.9.5+dfsg-0ubuntu2.5) ...
Setting up libqt5multimedia5:amd64 (5.9.5-0ubuntu1) ...
Setting up libsmi2ldbl:amd64 (0.4.8+dfsg2-15) ...
Setting up libwsutil9:amd64 (2.6.10-1~ubuntu18.04.0) ...
Setting up libwireshark-data (2.6.10-1~ubuntu18.04.0) ...
Setting up libwscodecs2:amd64 (2.6.10-1~ubuntu18.04.0) ...
Setting up libc-ares2:amd64 (1.14.0-1) ...
Setting up libmaxminddb0:amd64 (1.3.1-1) ...
Setting up libqt5multimediawidgets5:amd64 (5.9.5-0ubuntu1) ...
Setting up libqgsttools-p1:amd64 (5.9.5-0ubuntu1) ...
Setting up libwiretap8:amd64 (2.6.10-1~ubuntu18.04.0) ...
Setting up libqt5multimedia5-plugins:amd64 (5.9.5-0ubuntu1) ...
Setting up libwireshark11:amd64 (2.6.10-1~ubuntu18.04.0) ...
Setting up wireshark-common (2.6.10-1~ubuntu18.04.0) ...
Setting up wireshark-qt (2.6.10-1~ubuntu18.04.0) ...
Setting up wireshark (2.6.10-1~ubuntu18.04.0) ...
Processing triggers for desktop-file-utils (0.23-1ubuntu3.18.04.2) ...
Processing triggers for libc-bin (2.27-3ubuntu1.3) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for shared-mime-info (1.9-2) ...
Processing triggers for gnome-menus (3.13.3-11ubuntu1.1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
```