



Search and Processing Text



Prabhjeet Singh

Search and Processing Text – grep command

grep searches for PATTERNS in each FILE. PATTERNS is one or more patterns separated by newline characters, and grep prints each line that matches a pattern. Typically PATTERNS should be quoted when grep is used in a shell command.

A FILE of “-” stands for standard input. If no FILE is given, recursive searches examine the working directory, and non-recursive searches read standard input.

In addition, the variant programs **egrep**, **fgrep** and **rgrep** are the same as **grep -E**, **grep -F**, and **grep -r**, respectively. These variants are deprecated but are provided for backward compatibility.

\$ grep bob wordlist.txt → it searches the bob string and substring in each line of the wordlist.txt file and print it.

\$ grep bob wordlist.txt

Bobtimber

Conbobman

Conbob

bob

If we want to search any file or line of file which don't have any particular letter or string, so use -v option

\$ grep -v e wordlist.txt -> it will print the lines of the file wordlist.txt which don't contain letter (e).

Search in multiple files.

\$ sudo grep eroor /var/log/*.log

[sudo] password for kali:

```
/var/log/auth.log:Sep 1 11:55:39 kali sudo: kali : TTY=pts/0 ; PWD=/home/kali/linux_learning ; USER=root ; COMMAND=/usr/bin/grep eroor
/var/log/alternatives.log /var/log/auth.log /var/log/boot.log /var/log/daemon.log /var/log/dpkg.log /var/log/fontconfig.log /var/log/kern.log
/var/log/macchanger.log /var/log/user.log /var/log/vmware-network.1.log /var/log/vmware-network.2.log /var/log/vmware-network.3.log
/var/log/vmware-network.4.log /var/log/vmware-network.5.log /var/log/vmware-network.6.log /var/log/vmware-network.7.log
/var/log/vmware-network.8.log /var/log/vmware-network.9.log /var/log/vmware-network.log /var/log/vmware-vmsvc-root.1.log
/var/log/vmware-vmsvc-root.2.log /var/log/vmware-vmsvc-root.3.log /var/log/vmware-vmsvc-root.log /var/log/vmware-vmtoolsd-root.log
/var/log/vmware-vmusr-root.log /var/log/Xorg.0.log /var/log/Xorg.1.log
```

Sort command

Use to sort the data of the file.

\$ sort file2.txt

another file

demo

this is

to show you the

└─\$ cat file2.txt

this is

another file

to show you the

demo

\$ uniq file2.txt -> it will remove the adjacent duplicate lines.

Count the number of lines and bytes in a file, use wc utility

\$ wc file2.txt

4 9 44 file2.txt

Only lines

\$ wc -l file2.txt

4 file2.txt

\$ grep bob wordlist.txt | wc -l

7

\$ grep -v e random-words.txt | sort | uniq | wc -l

- ➔ This command displaying the word count of the data.
- ➔ By finding sorted and unique data in a file which don't contain 'e' in the each line.

SED utility – stream editor.

Sed is a stream editor. A stream editor is used to perform basic text transformations on an input stream (a file or input from a pipeline). While in some ways similar to an editor which permits scripted edits (such as ed), sed works by making only one pass over the input(s), and is consequently more efficient. But it is sed's ability to filter text in a pipeline which particularly distinguishes it from other types of editors.

\$ sed 's/Suite/Ste/' sample.txt

This will replace Suite word with Ste in the sample.txt file.

\$ sed 's/Suite/Ste/' sample.txt

This will replace last occurrence of Suite word with Ste in the sample.txt file.

\$ sed '/Suite/d' sample.txt

➔ It'll delete the line contains Suite word.

\$ sed '/ee/' 's/Suite/Ste' sample.txt

➔ It'll replace the Suite with Ste, only in those line which contains "ee" substring or string

AWK Command

Gawk is the GNU Project's implementation of the AWK programming language.

AWK breaks each line of input into separate fields or columns using specific delimiters.

\$ echo bob john killing each | awk '{print \$2}'

John

```
$ echo linux bob sally | awk '{print $3,"likes",$1}'
```

sally likes linux

```
bob@linux101:~$ cat sample.txt
Daven Driscoll,33257 Ortiz Prairie Suite 330,Johnsshire,New Hampshire,33360-8902
Minne Whyberd,98511 Hills Pass Apt. 353,West Sheilatown,Florida,53412
Betteanne Secombe,1760 Breanna Rest Suite 823,Port Marleefurt,Georgia,86349
Marleen Dunnet,919 Cole Mill Suite 706,Patrickside,North Carolina,59979
Otis Barbosa,48189 Ankunding Landing,East Giaport,Tennessee,31975
Theresita Shapiro,3953 Graham Ferry Apt. 054,Bransonmouth,Kentucky,80455
Anitra Borrel,292 Darlene Parks,Trantowfurt,South Dakota,21278-7321
Janine Hurnell,7750 Clotilde Ville Apt. 210,Feestmouth,Indiana,84331-9527
Lula Swyne,603 Anahi Tunnel,Lake Marciamouth,North Dakota,98915
Emlyn Rendell,419 Merl Falls Suite 223,Peterborough,Mississippi,63091
```

```
bob@linux101:~$ awk -F ',' '{print $1}' sample.txt
Daven Driscoll
Minne Whyberd
Betteanne Secombe
Marleen Dunnet
Otis Barbosa
Theresita Shapiro
Anitra Borrel
Janine Hurnell
Lula Swyne
Emlyn Rendell
bob@linux101:~$
```

```
bob@linux101:~$ awk -F ',' '{print $1}' sample.txt | awk '{print $2 "," $1}'
Driscoll,Daven
Whyberd,Minne
Secombe,Betteanne
Dunnet,Marleen
Barbosa,Otis
Shapiro,Theresita
Borrel,Anitra
Hurnell,Janine
Swyne,Lula
Rendell,Emlyn
bob@linux101:~$ awk -F ',' '{print $1}' sample.txt | awk '{print $2 " ", " $1}'
Driscoll, Daven
Whyberd, Minne
Secombe, Betteanne
Dunnet, Marleen
Barbosa, Otis
Shapiro, Theresita
Borrel, Anitra
Hurnell, Janine
Swyne, Lula
Rendell, Emlyn
bob@linux101:~$
```

```
awk -F ',' '/Dakota/ {print $1}' sample.txt
→ Print the names which has Dakota as substring.
```

```
awk -F ',' '/Dakota/ {print NR,$1}' sample.txt
print with Line number , NR
```

TR utility - translate

It replaces and delete characters sent from std input and output and writes results in text.

```
cat sample.txt | tr ',' '\t'
replaces comma (,) with tab
```

```
cat sample.txt | tr 'a-z' 'A-Z'
replaces all lower-case letters to upper case letters.
```

```
cat sample.txt | tr '[:lower:]' '[:upper:]'
replaces all lower-case letters to upper case letters.
```

Manipulating Text

```
man sed
cat sample.txt
sed 's/Suite/Ste/' sample.txt
echo Suite Suite | sed 's/Suite/Ste/'
echo Suite Suite | sed 's/Suite/Ste/g'
sed '$s/Suite/Ste/g' sample.txt
sed '/Suite/d' sample.txt
grep Suite sample.txt
sed '/ee/ s/Suite/Ste/g' sample.txt
```

```
sed 's/$/\n/g' sample.txt | sed 's/,/\n/g'
sed -e 's/$/\n/g' -e 's/,/\n/g' sample.txt
```

AWK Command

```
echo linux bob sally | awk '{print $2}'
echo linux bob sally | awk '{print $3,"likes",$1}'
awk -F ',' '{print $1}' sample.txt
awk -F ',' '{print $1}' sample.txt | awk '{print $2 " ", " $1}'
cat sample.txt
awk -F ',' '/Dakota/ {print $1}' sample.txt
awk -F ',' '/Dakota/ {print NR,$1}' sample.txt
cat sample.txt | tr ',' '\t'
cat sample.txt | tr 'a-z' 'A-Z'
man tr
cat sample.txt | tr '[:lower:]' '[:upper:]'
```

Networking Commands

1. **Ping** - send ICMP ECHO_REQUEST to network hosts

Ping uses the ICMP protocol's mandatory ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams ("pings") have an IP and ICMP header, followed by a struct timeval and then an arbitrary number of "pad" bytes used to fill out the packet. ping works with both IPv4 and IPv6. Using only one of them explicitly can be enforced by specifying -4 or -6. Ping can also send IPv6 Node Information Queries (RFC4620). Intermediate hops may not be allowed, because IPv6 source routing was deprecated (RFC5095).

└─\$ ping -c 4 google.com

```
PING google.com (142.251.41.78) 56(84) bytes of data.
64 bytes from yyz10s20-in-f14.1e100.net (142.251.41.78): icmp_seq=1 ttl=128 time=18.8 ms
64 bytes from yyz10s20-in-f14.1e100.net (142.251.41.78): icmp_seq=2 ttl=128 time=18.6 ms
64 bytes from yyz10s20-in-f14.1e100.net (142.251.41.78): icmp_seq=3 ttl=128 time=17.6 ms
64 bytes from yyz10s20-in-f14.1e100.net (142.251.41.78): icmp_seq=4 ttl=128 time=19.3 ms
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 17.623/18.570/19.319/0.611 ms
```

└─\$ ping -c 4 google.com

This is for unlimited packets and above is for 4 packets.

2. **ifconfig** - configure a network interface.

Ifconfig is used to configure the kernel-resident network interfaces. It is used at boot time to set up interfaces as necessary. After that, it is usually only needed when debugging or when system tuning is needed. If no arguments are given, ifconfig displays the status of the currently active interfaces. If a single interface argument is given, it displays the status of the given interface only; if a single -a argument is given, it displays the status of all interfaces, even those that are down. Otherwise, it configures an interface.

3. **\$ ifwconfig** – configure wireless network Interface

4. **\$ ip address** – protocol address management

The address is a protocol (IPv4 or IPv6) address attached to a network device. Each device must have at least one address to use the corresponding protocol. It is possible to have several different addresses attached to one device. These addresses are not discriminated, so that the term alias is not quite appropriate for them, and we do not use it in this document.

5. **\$ ip -s link**

6. **View and modify routing tables –**

a. **\$ ip route**

b. **\$ route**

7. **DNS lookup on a domain name.**

a. **\$ nslookup google.com**

b. **\$ dig google.com**

c. **\$ dig -x 8.8.8.8 → dig can perform reverse lookup also.**

\$ nslookup google.com

Server: 192.168.222.2

Address: 192.168.222.2#53

Non-authoritative answer:

Name: google.com

Address: 142.251.41.78

Name: google.com

Address: 2607:f8b0:400b:802::200e

└─\$ dig google.com

```
; <<>> DiG 9.16.15-Debian <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 41241
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0x0005, udp: 1220
; COOKIE: e5a98919424af6b784f4902e6310ef3c1f845ad8b15f7079 (good)
;; QUESTION SECTION:
;google.com.          IN      A
;; ANSWER SECTION:
google.com.          5      IN      A      172.217.1.14
;; Query time: 16 msec
;; SERVER: 192.168.222.2#53(192.168.222.2)
;; WHEN: Thu Sep 01 13:43:24 EDT 2022
;; MSG SIZE rcvd: 83
```

─\$ dig -x 8.8.8.8

```
; <<>> DiG 9.16.15-Debian <<>> -x 8.8.8.8
;; global options: +cmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 44821
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0x0005, udp: 1220
; COOKIE: f44e6ccc76adc429159447d16310eff1196acc9a7accedd0 (good)
;; QUESTION SECTION:
;8.8.8.8.in-addr.arpa.  IN      PTR
;; ANSWER SECTION:
8.8.8.8.in-addr.arpa. 5      IN      PTR    dns.google.
;; Query time: 20 msec
;; SERVER: 192.168.222.2#53(192.168.222.2)
;; WHEN: Thu Sep 01 13:46:25 EDT 2022
```

;; MSG SIZE rcvd: 101

8. Netstat -> display network status and information

- a. To see all the open tcp connections.

\$netstat -at

- b. Listening tcp ports

\$ netsta -lt

Networking at the Command Line

```
ping google.com
ping -c 3 google.com
ifconfig
man ifconfig
ip address
ip -s link
ip help
ip address help
ip link help
ip address
sudo ip link set dev enp0s3 down
ip address
ping google.com
sudo ip link set dev enp0s3 up
ip address
ping google.com
ip route
route
sudo ip route add 10.0.3.0/24 via 10.0.2.1
ip route
sudo ip route delete 10.0.3.0/24 via 10.0.2.1
ip route
nslookup google.com
dig google.om
dig -x 8.8.8.8
netstat -at
netstat -at
netstat -lt
Type in separate terminal: python3 -m http.server
netstat -lt
```

File Transfer Utilities – scp and rsync

1. scp – openSSH secure shell copy

scp copies files between hosts on a network. It uses ssh(1) for data transfer, and uses the same authentication and provides the same security as ssh(1). scp will ask for passwords or passphrases if they are needed for authentication.

```
$ scp <source path> <destination path>
```

```
$ scp file1.txt 192.168.1.4:/home/bob/
```

```
$ scp -r sally/file1 192.168.1.4:/home/bob/ → for directory transfer
```

From remote machine to local

```
$ scp 192.168.100.4:/home/bob/file2.txt backup/
```

```
$ scp -r 192.168.100.4:/home/bob/ backup/ → for directory transfer
```

If user name is different

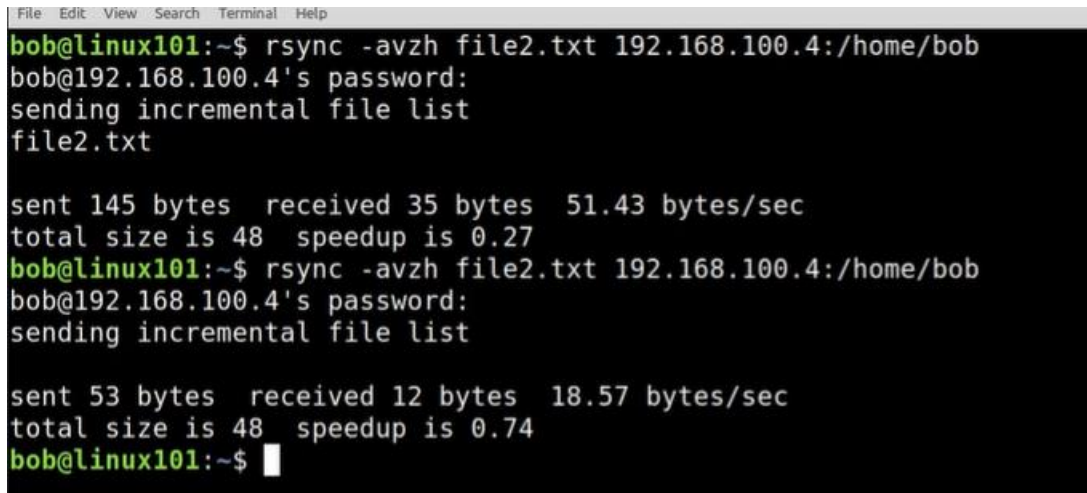
```
$ scp file1.txt john@192.168.1.4:/home/bob/
```

2. rsync – a fast, versatile, remote (and local) file-copying tool

- this command also do the same function as scp but it has one advantage which is – for example if we update backup folder daily from remote machine, so in this case it will update only that file which is changed that day not the whole folder.
- Rsync is a fast and extraordinarily versatile file copying tool. It can copy locally, to/from another host over any remote shell, or to/from a remote rsync daemon. It offers a large number of options that control every aspect of its behavior and permit very flexible specification of the set of files to be copied. It is famous for its delta-transfer algorithm, which reduces the amount of data sent over the network by sending only the differences between the source files and the existing files in the destination. Rsync is widely used for backups and mirroring and as an improved copy command for everyday use.
- Rsync finds files that need to be transferred using a "quick check" algorithm (by default) that looks for files that have changed in size or in last-modified time. Any

changes in the other preserved attributes (as requested by options) are made on the destination file directly when the quick check indicates that the file's data does not need to be updated.

```
rsync -avzh file2.txt 192.168.100.4:/home/bob/  
a - archive mode  
v - verbose mode  
z - compressing file and data  
h - display output in human readable format.
```



```
File Edit View Search Terminal Help  
bob@linux101:~$ rsync -avzh file2.txt 192.168.100.4:/home/bob  
bob@192.168.100.4's password:  
sending incremental file list  
file2.txt  
  
sent 145 bytes  received 35 bytes  51.43 bytes/sec  
total size is 48  speedup is 0.27  
bob@linux101:~$ rsync -avzh file2.txt 192.168.100.4:/home/bob  
bob@192.168.100.4's password:  
sending incremental file list  
  
sent 53 bytes  received 12 bytes  18.57 bytes/sec  
total size is 48  speedup is 0.74  
bob@linux101:~$
```

Convert Text files.

\$ file *.txt → to find the format of the file

```
all.txt:      UTF-8 Unicode text  
combine1.txt: ASCII text  
dir_list.txt: ASCII text  
errors.txt:   UTF-8 Unicode text  
file1_hard.txt: ASCII text
```

Convert unix to windows file → unix2dos

\$ unix2dos temp.txt

\$ unix2dos original.txt temp.txt

\$ unix2dos -c mac temp.txt

Convert dos to unix

```
$ dos2unix temp.txt
```

```
$ dos2unix -c mac temp.txt
```