Cheat sheet

Intermediate Linux commands

This cheat sheet presents a collection of Linux commands and executables typically used by developers who want to move beyond the basics of working with the Linux operating system. For the purpose of this cheat sheet, *intermediate* use involves managing processes, users, and groups on a particular machine running under Linux, as well as monitoring disk and network usage. Commands in this cheat sheet are organized by category.

Console and output management commands

Commands in this section apply to working in a terminal window console and illustrate output from a computer or virtual machine running the Linux operating system.

history

```
history [options]
```

Displays a list of commands executed on the system. The `history` command can also be used to manipulate the history list and the way that history information is displayed.

Example:

The following example uses the `history` command to show a list of commands that have been executed on the system. The example pipes the result to the `more` command, which shows the first 15 lines of output using the `-15` option:

```
$ history | more -15
   24  diag
   25  ss
   26  uname
   27  lscpu
   28  timedatectl
   29  date
   30  chronyc
   31  lshw
   32  sosreport
   33  sos
   34  tlog
   35  fsck
   36  fsck --help
   37  fsck -A
   38  sudo fsck -A
--More--
```

more

```
more [options] </path/to/filename or stdout>
```

Allows a user to view and traverse the contents of a file or stdout. The `more` command runs within its own command-line user interface. To exit the process, press the `q` key.

Example:

This example uses the `more` command to display the first four lines of the file `/etc/passwd`. Users can then traverse the remainder of the file one line at a time by striking the `<ENTER>` key:

```
$ more -4 /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
--More--(5%)
```

top

```
top [options]
```

Displays information about the running Linux processes.

Example:

The following command displays the `top` command with the result piped to the `more` command in order to view the first portion of the output:

```
$ top | more
top - 12:02:29 up 5 days, 20:20,  2 users,  load average: 0.01, 0.02, 0.00
Tasks: 201 total,   2 running, 199 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  6.2 sy,  0.0 ni,93.8 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0
st
MiB Mem :  7770.8 total,  5409.8 free,  1240.8 used,  1120.2 buff/cache
MiB Swap:  8092.0 total,  8092.0 free,     0.0 used.  6205.6 avail Mem
    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+
COMMAND
  82399 guest  20   0   65584   5120   4212 R   5.9   0.1   0:00.02 top
      1 root      20   0  175932  14212   8924 S   0.0   0.2   0:06.21
systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.13
kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00
rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00
rcu_par_gp
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00
kworker/0:0H-events_highpri
      9 root       0 -20       0      0      0 I   0.0   0.0   0:00.00
mm_percpu_wq
     10 root      20   0       0      0      0 S   0.0   0.0   0:02.73
ksoftirqd/0
     11 root      20   0       0      0      0 R   0.0   0.0   0:01.10
rcu_sched
     12 root      rt   0       0      0      0 S   0.0   0.0   0:00.00
migration/0
     13 root      rt   0       0      0      0 S   0.0   0.0   0:00.04
watchdog/0
```

```
     14 root      20   0       0      0      0 S   0.0   0.0   0:00.00
cpuhp/0
     16 root      20   0       0      0      0 S   0.0   0.0   0:00.00
kdevtmpfs

--More--
```

Disk management commands

Commands in this section apply to working with disks, devices, and volumes on a computer running the Linux operating system.

df

```
df [options] <file name>
```

Shows the amount of disk space used and available according to the file system that represents a particular disk device mount. If no file name is given, the space available on all mounted file systems is displayed.

Example:

The following example shows the invocation and result of `df` displaying all mounted file systems. Disk space is shown in 1K blocks (note that `$` is the command-line prompt symbol):

```
$ df
Filesystem            1K-blocks    Used Available Use% Mounted on
devtmpfs                3949180       0   3949180   0% /dev
tmpfs                   3978636       0   3978636   0% /dev/shm
tmpfs                   3978636    9464   3969172   1% /run
tmpfs                   3978636       0   3978636   0% /sys/fs/cgroup
/dev/mapper/rhel-root  50065528 5588744  44476784  12% /
/dev/mapper/rhel-home  24445276  228104  24217172   1% /home
/dev/sda1               1038336  262796    775540  26% /boot
tmpfs                    795724      64    795660   1% /run/user/1000
```

du

```
du [options] <starting directory or file>
```

Reports information about disk usage on the local computer or virtual machine.

Example:

The following example uses the command `du` to report the amount of disk space used by the files in the directory `/etc/bin`:

```
$ du /usr/bin
365940      /usr/bin
```

File and directory management commands

Commands in this section apply to working with files and directories on a computer running the Linux operating system.

find

```
sudo find <starting/directory> -name <file/directory name>
```

Finds a file or directory by name.

Example:

The following command finds a file named `hostname` starting from the root (`/`) directory of the computer's file system. Note that the command starts with `sudo` in order to access files restricted to the `root` user:

```
$ sudo find / -name hostname
/proc/sys/kernel/hostname
/etc/hostname
/var/lib/selinux/targeted/active/modules/100/hostname
/usr/bin/hostname
/usr/lib64/gettext/hostname
/usr/share/licenses/hostname
/usr/share/doc/hostname
/usr/share/bash-completion/completions/hostname
/usr/share/selinux/targeted/default/active/modules/100/hostname
/usr/libexec/hostname
```

pwd

```
pwd
```

Displays the name of the present working directory.

Example:

The following example displays the invocation and result of using the command `pwd` in the `HOME` directory for a user named `guest`:

```
$ pwd
/home/guest
```

alias

```
alias [options] <shortcut=command>
```

Assigns a shortcut name to an existing command or executable.

Example:

The following example creates a temporary alias for the `clear` command. The alias is named `cls`. The clear command clears the terminal window. Once created, `cls` will also clear the terminal window:

```
$ alias cls='clear'
```

awk

```
awk <processing instruction string>
```

Finds, filters, or replaces text in a file or from stdout.

Example:

This example pipes the string "Bobby is cool" to the awk command. The `awk` command invokes the subcommand named `sub` to find any occurrence of "Bobby" and change the string to "Teddy". Then, the subcommand `print` outputs the result of the substitution:

```
$ echo "Bobby is cool" | awk '{sub("Bobby","Teddy"); print}'
Teddy is cool
```

This example uses `awk` to filter output according to field position. First, the example shows the output of the `who` command, which lists the current users logged in to the computer. The who command displays four fields (columns). The fields are username, the terminal line number, the login time, and the machine from which access originated.

The second execution of `who` pipes the result to `awk`. Then, awk uses the `print $1` subcommand set to show only the first field name. The third execution of `who` pipes the result to `awk`, which then filters input to print the values in the second field:

```
$ who
jaggermick pts/0        2022-01-19 09:14 (192.168.86.28)
guest pts/1        2022-01-19 10:07 (192.168.86.20)

$ who | awk '{print $1}'
jaggermick
guest

$ who | awk '{print $2}'
pts/0
pts/1
```

diff

```
diff [options] file1 file2
```

Displays the difference in content between two files.

Example:

The following example uses the `printf` command to create three files named `one.txt`, `two.txt`, and `three.txt`. Each file contains a list of names. The files named `one.txt` and `three.txt` have identical content. The file `two.txt` has an additional name.

The first invocation of diff compares the files `one.txt` and `two.txt`. The second invocation compares files `one.txt` to `three.txt`.

The first invocation reports that there is a difference in `two.txt` and that the fourth line from the file `two.txt` should be added (a) to the third line in `one.txt`. The value of the fourth line in `two.txt` is `Shemp`.

The second invocation uses the `-s` option to display the report that indicates the files `one.txt` and `three.txt` are identical. If the `-s` option was not used, there would be no output to the console (by default, identical files are not reported in stdout):

```
$ printf  "Moe\nLarry\nCurly\n" > one.txt
$ printf  "Moe\nLarry\nCurly\nShemp\n" > two.txt
$ printf  "Moe\nLarry\nCurly\n" > three.txt

$ diff one.txt two.txt
3a4
> Shemp

$ diff -s one.txt three.txt
Files one.txt and three.txt are identical
```

sed

```
sed [options] <manipulation instructions> <path/to/filename or stdout>
```

Manipulates the content of a file or output sent to stdout.

Example:

The following example uses the `echo` command to send the string `Bobby is cool` to the `sed` command. The command `sed` uses the s subcommand to substitute the name `Teddy` where the name `Bobby` occurs. The output is then displayed:

```
$ echo Bobby is cool | sed 's/Bobby/Teddy/'
Teddy is cool
```

Network commands

Commands in this section apply to working with networks on and from a Linux computer.

hostname

```
hostname
```

Reports the local computer's host name.

Example:

```
$ hostname
localhost.localdomain
```

nslookup

```
nslookup [options] <domain_name>
```

A program that queries for information about a particular Internet domain name.

Example:

The following example invokes `nslookup` against the domain name `developers.redhat.com`:

```
$ nslookup developers.redhat.com
Server:          192.168.86.1
Address:    192.168.86.1#53

Non-authoritative answer:
developers.redhat.com   canonical name =
developers.redhat.com2.edgekey.net.
developers.redhat.com2.edgekey.net canonical name =
developers.redhat.com2.edgekey.net.globalredir.akadns.net.
developers.redhat.com2.edgekey.net.globalredir.akadns.net  canonical name =
e40408.dsca.akamaiedge.net.
Name: e40408.dsca.akamaiedge.net
Address: 23.199.47.87
Name: e40408.dsca.akamaiedge.net
Address: 23.199.47.85
Name: e40408.dsca.akamaiedge.net
Address: 2600:1406:3400::6862:7512
Name: e40408.dsca.akamaiedge.net
Address: 2600:1406:3400::6862:7543
```

traceroute

```
traceroute [options] <target address or domain_name>
```

Reports the route that a packet takes in hops to move through the Internet to reach its destination.

The program `traceroute` is not part of Red Hat Enterprise Linux (RHEL) by default. It must be installed using `sudo dnf install traceroute`.

Example:

The following example reports `nslookup` from the local machine to `developers.redhat.com`. The `-m` option is used to limit the output to the first five hops:

```
$ traceroute -m 5 developers.redhat.com
traceroute to developers.redhat.com (23.199.47.85), 5 hops max, 60 byte
packets
 1  _gateway (192.168.86.1)  0.599 ms  0.514 ms  0.656 ms
 2  142-254-237-093.inf.spectrum.com (142.254.237.93)  11.974 ms  11.874 ms
17.793 ms
 3  agg53.lsaicaev02h.socal.rr.com (24.30.168.85)  19.294 ms  20.242 ms
19.224 ms
 4  72.129.19.22 (72.129.19.22)  18.984 ms  19.888 ms  19.969 ms
 5  agg26.tustcaft01r.socal.rr.com (72.129.17.2)  13.575 ms  19.673 ms
13.579 ms
```

RHEL management commands

The commands in this section apply to working with the Red Hat Enterprise Linux operating system.

sestatus

```
sestatus [options]
```

This program is used to report status information about a computer or virtual machine running SELinux.

Example:

The following example invokes the program `sestatus` and displays the default response:

```
$$ sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33
```

uname

```
uname [options]
```

The command `uname` reports system information about the local computer.

Example:

The following example uses the `-a` option with `uname` to report all system information about the local computer:

```
$ uname -a
Linux localhost.localdomain 4.18.0-348.el8.x86_64 #1 SMP Mon Oct 4 12:17:22
EDT 2021 x86_64 x86_64 x86_64 GNU/Linux
```

Users and groups commands

The following commands apply to working with users and groups as supported by the Linux operating system.

users

```
users [options]
```

Displays the names of users logged in to the computer.

Example:

The following example uses the command `users` to list all users logged into the system:

```
$ users
cooluser jaggermick lennonjohn
```

useradd

```
adduser [options] <username>
```

Adds a user to the computing environment. The command must be run as `sudo` for administrator access.

Example:

The following example adds a user with the login name `cooluser`. The `HOME` directory `home/cooluser` is created by default. Then, the example invokes the command `passwd` to set a password for the new user:

```
$ sudo adduser  cooluser

$ sudo passwd cooluser
Changing password for user cooluser.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

userdel

```
userdel [options] <username>
```

Deletes a user from the computer. The command must be run as `sudo` for administrator access.

Example:

The following example uses the `userdel` command to remove the user with the login name `cooluser` from the system. The `-r` option indicates that the user's `HOME` directory is also to be deleted:

```
$ sudo userdel -r cooluser
```

usermod

```
usermod [options] <username>
```

Modifies user account information and can be used to add a user to a group. The command must be run as `sudo` for administrator access.

Example:

The following example uses the command `usermod` to add a user with the login name `lennonjohn` to a group named `beatles`. Then, the command groups is used to verify that the user `lennonjohn` is indeed assigned to the group `beatles`:

```
$ sudo usermod -a -G beatles lennonjohn

$ groups lennonjohn
lennonjohn : lennonjohn beatles
```

groups

```
groups [options] <username>
```

Lists the groups to which a user belongs.

Example:

The following example uses the command `groups` to list the groups to which the user with the username `lennonjohn` belongs:

```
$ groups lennonjohn
lennonjohn : lennonjohn beatles
```

gpasswd

```
gpasswd [options] <group>
```

The command `gpasswd` is used to manage the configuration of a group under the Linux operating system. The command must be run as `sudo` for administrator access.

Example:

The following example uses `gpasswd` to remove a user from a group. The `-d` option followed by the username indicates that the user is to be deleted:

```
$ sudo gpasswd -d jaggermick beatles
Removing user jaggermick from group beatles
```

groupadd

```
groupadd [options] <groupname>
```

Adds a group to the computer. The command must be run as `sudo` for administrator access.

Example:

The following example uses the `groupadd` command to create a group named `beatles`.

```
$ sudo groupadd beatles
```

groupdel

```
groupdel [options] <groupname>
```

Deletes a group from the computer. The command must be run as `sudo` for administrator access.

Example:

The following example uses the command `groupdel` to delete the group named `beatles` from the system:

```
$ sudo groupdel beatles
```