

```

waparch:~
linux x86_64
waparch
ease: 3.19.3-1-ARCH
48

1178
B / 3909 MB
Type: Intel(R) Core(TM) i7-4790K CPU @
one
/ 79G (10%) (ext4)

```

```

swapni
File Edit View Search Terminal
waparch% archey3

      +
      #
      ###
      #####
      #####
      ; #####
      +### #####
      +#####
      #####;

4.00GHz
#####+
#####
.#####;###"
.#####;#####
#####.#####
#####'#####
#####'#####
;####;####
##'#####'##
#'#'#####'##

waparch% █

```

# Essential Linux Process Management Commands

In this presentation, we will cover several key commands in Linux for process management. By the end of this presentation, you will master the `ps`, `kill`, `top`, and `pkill` commands.



by Pranav Gaddi

```
3)  SIGQUIT      4)  SIGI
8)  SIGFPE       9)  SIGK
13) SIGPIPE      14) SIGA
18) SIGCONT      19) SIGS
23) SIGURG       24) SIGX
28) SIGWINCH     29) SIGI
35) SIGRTMIN+1   36) SIGR
40) SIGRTMIN+6   41) SIGR
45) SIGRTMIN+11  46) SIGR
50) SIGRTMAX-14  51) SIGR
55) SIGRTMAX-9   56) SIGR
60) SIGRTMAX-4   61) SIGR
```

# The `ps` Command (Process Status)

The ``ps`` command is used to see detailed information about running processes. It stands for "process status." You can quickly check what processes are running with the simple command, ``ps``. By default, it shows the processes running for the current user.

```

ra.wikipedia.org
g (208.80.152.2) 56(84) bytes of data.

ping statistics ---
ceived, 0% packet loss, time 0ms
28/540.528/540.528/0.000 ms

5 Jul 30 22:43 .
5 Sep 14 20:42 ..
5 May 14 00:15 account
5 Jul 31 22:26 cache
5 May 18 16:03 db
5 May 18 16:03 empty
5 May 18 16:03 games
5 Jun 2 18:39 gdm
5 May 18 16:03 lib
5 May 18 16:03 local
1 May 14 00:12 lock -> ../run/lock
5 Sep 14 20:42 log
9 Jul 30 22:43 mail -> spool/mail
5 May 18 16:03 nis
5 May 18 16:03 opt
5 May 18 16:03 preserve
5 Jul 1 22:11 report
5 May 14 00:12 run -> ../run
5 May 18 16:03 spool
5 Sep 12 23:50 tmp
5 May 18 16:03 yp
arch wiki
resto, refresh-packagekit, remove-with-leaves
ry_db

```

73% [=====] 62

# `ps` Options and Usage

There are many commands you can use with the `ps` command, depending on what you want to see. The most common `ps` flags are `-e`, which shows all processes, and `-f`, which shows extended information.

Other `ps` options include `-l`, `-u`, and `-x`, each of which presents detailed information about each process. To see all options and usage, type `man ps` in the command line.

```
File Edit View Terminal Tabs Help
Untitled x Unt
krd@ubuntu:~$ ps axw | grep tick
31915 ?      R       0:03 /home/krd/chanakya/src/code/mul
31916 ?      R       0:04 /home/krd/chanakya/src/code/mul
31917 ?      R       0:02 /home/krd/chanakya/src/code/mul
31918 ?      R       0:03 /home/krd/chanakya/src/code/mul
34074 pts/4    R+      0:00 grep --color=auto tick
krd@ubuntu:~$ sudo strace -f 31915
strace: Can't stat '31915': No such file or directory
krd@ubuntu:~$ sudo strace -f -p 31915
Process 31915 attached

^C

^C^C^C^C^C^C^Z
[1]+  Stopped                  sudo strace -f -p 31915
krd@ubuntu:~$ ps axw | grep strace
34169 pts/4    T       0:00 sudo strace -f -p 31915
34170 pts/4    T       0:00 strace -f -p 31915
34211 pts/4    S+      0:00 grep --color=auto strace
krd@ubuntu:~$ sudo kill -9 34169
[1]+  Killed                  sudo strace -f -p 31915
krd@ubuntu:~$
krd@ubuntu:~$
krd@ubuntu:~$ sudo strace -f -p 31916
Process 31916 attached

^C^C^C^Z
[1]+  Stopped                  sudo strace -f -p 31916
krd@ubuntu:~$ ps axw | grep strace
34170 pts/4    S       0:00 strace -f -p 31915
34238 pts/4    T       0:00 sudo strace -f -p 31916
34239 pts/4    T       0:00 strace -f -p 31916
34257 pts/4    S+      0:00 grep --color=auto strace
krd@ubuntu:~$ sudo kill -9 34170 34239
krd@ubuntu:~$ ps axw | grep strace
34238 pts/4    T       0:00 sudo strace -f -p 31916
34239 pts/4    Z       0:00 [strace] <defunct>
34282 pts/4    S+      0:00 grep --color=auto strace
krd@ubuntu:~$
```

# The **kill** Command

The `kill` command is used to send signals to specific running processes in order to stop them. With this command, you can stop running processes that are causing problems for your system, among other things.

# `kill` Signals and Usage

Each process has a unique identifier known as a Process ID (PID). To stop a process using the `kill` command, you will need its PID. The commands `kill`, `killall`, and `pkill` are all used to send a signal to a process.

The most commonly used signals with `kill` are `SIGTERM` and `SIGKILL`. `SIGTERM` tells the process to terminate gracefully, while `SIGKILL` will force the process to stop.

```
28 0 4971768 1.1g 118556 R 100.0 38.9 23:36.27 gnome-she
29 0 758740 2.4g 21472 S 0.0 0.0 0:52.94 systemd-j
30 0 0 0 0 S 0.0 0.0 0:14.38 rcu_sched
31 0 162168 2400 1500 R 0.4 0.1 0:01.10 top
32 0 1307914 4052 2500 S 0.0 0.1 0:04.55 systemd
33 0 0 0 0 S 0.0 0.0 0:00.03 kthreadd
34 -20 0 0 0 S 0.0 0.0 0:00.00 kworker/0
35 0 0 0 0 S 0.0 0.0 0:00.07 ksoftirqd
36 0 0 0 0 S 0.0 0.0 0:00.00 migration
37 0 0 0 0 S 0.0 0.0 0:00.00 rcu_bh
38 -20 0 0 0 S 0.0 0.0 0:00.00 lru-add-d
39 0 0 0 0 S 0.0 0.0 0:00.23 watchdog/
40 0 0 0 0 S 0.0 0.0 0:00.17 watchdog/
41 0 0 0 0 S 0.0 0.0 0:00.30 migration
42 0 0 0 0 S 0.0 0.0 0:00.05 ksoftirqd
43 -20 0 0 0 S 0.0 0.0 0:00.00 kworker/1
44 0 0 0 0 S 0.0 0.0 0:00.16 watchdog/
45 0 0 0 0 S 0.0 0.0 0:00.03 migration
46 0 0 0 0 S 0.0 0.0 0:00.18 ksoftirqd
47 -20 0 0 0 S 0.0 0.0 0:00.00 kworker/2
48 0 0 0 0 S 0.0 0.0 0:00.16 watchdog/
49 0 0 0 0 S 0.0 0.0 0:00.06 migration
50 0 0 0 0 S 0.0 0.0 0:00.05 ksoftirqd
51 -20 0 0 0 S 0.0 0.0 0:00.00 kworker/3
52 0 0 0 0 S 0.0 0.0 0:00.00 kdevtmpfs
53 -20 0 0 0 S 0.0 0.0 0:00.00 netns
54 0 0 0 0 S 0.0 0.0 0:00.03 khangtaskd
55 -20 0 0 0 S 0.0 0.0 0:00.00 writeback
56 -20 0 0 0 S 0.0 0.0 0:00.00 kintegrity
57 -20 0 0 0 S 0.0 0.0 0:00.00 bioset
58 -20 0 0 0 S 0.0 0.0 0:00.00 bioset
59 -20 0 0 0 S 0.0 0.0 0:00.00 bioset
60 -20 0 0 0 S 0.0 0.0 0:00.00 kblockd
61 -20 0 0 0 S 0.0 0.0 0:00.00 md
62 -20 0 0 0 S 0.0 0.0 0:00.00 edac-poller
63 -20 0 0 0 S 0.0 0.0 0:00.00 watchdogd
64 28 0 0 0 S 0.0 0.0 0:05.60 kswapd0
65 39 19 0 0 S 0.0 0.0 0:00.23 kswapd0
66 0 -20 0 0 S 0.0 0.0 0:00.00 crypto
67 0 -20 0 0 S 0.0 0.0 0:00.00 kthrotld
68 111 -s 9 2479
69 111 -KILL
70 111 -s sigspec 1 -s signal 1 -sigspec pid 1 joespec ... or kill
```

```
3: deepu@deepus-fedora: ~/workspace/XL/blueprints  + x
4: deepu@deepus-fedora: /usr  + x

~/workspace/XL/blueprints on * master 10 53% 22:38:46
> false

/usr 51% 22:38:43
> _
```

# The top Command

The `top` command is another way to check on running processes in real-time. It shows the system summary information and the processes that are currently being managed by the Linux kernel.

```
verage: 0.93, 1.65, 1.43
copped, 0 zombie
0.0%hi, 0.2%si, 0.0%st
e, 7584k buffers
e, 452396k cached
```

```
TIME+ COMMAND
0:06.66 top
24:54.54 python
0:00.14 python
0:00.86 sshd
0:00.00 sftp-server
0:07.78 sshd
0:01.94 sftp-server
0:00.84 sshd
0:00.00 sftp-server
0:03.66 sshd
0:00.00 bash
0:04.84 sshd
0:00.04 bash
0:00.62 tail
0:04.72 sshd
0:00.00 bash
0:00.48 python
0:21.36 python
0:00.96 sshd
0:00.00 sftp-server
0:00.00 java
0:00.00 java
0:00.00 java
0:04.78 sshd
0:00.00 bash
0:04.74 sshd
0:00.00 bash
0:01.32 sshd
0:00.04 sftp-server
0:00.02 tail
0:01.32 sshd
0:00.02 sftp-server
0:01.60 java
0:00.82 java
0:00.10 java
0:00.12 java
0:00.10 java
0:00.04 java
0:00.12 java
0:02.46 sshd
0:00.02 bash
0:00.44 top
0:00.34 top
0:00.10 top
0:00.22 top
0:01.56 tail
```

# Interactive top Features

The `top` command has several interactive features that allow you to sort running processes by various parameters (e.g., CPU usage), kill running processes, and refresh the display.

To sort the running processes, press **P** or **M** to sort by CPU usage or memory usage, respectively. To kill a running process, type its PID and press **K**. To refresh the display, press the spacebar.



```
#lspci -tv          =>Show PCI devices
#lsusb -tv          =>Show USB devices
#dmidecode           =>Show hardware info from the BIOS
#hdparm -i /dev/sda  =>Show info about disk sda
#hdparm -tT /dev/sda =>Do a read speed test on disk sda
#badblocks -s /dev/sda =>Test for unreadable blocks on disk sda
```

## USERS

```
#id                 =>Show the active user id with login and group
#last               =>Show last logins on the system
#who               =>Show who is logged on the system
#groupadd admin     =>Add group "admin"
#useradd -c "Sam Tomshi" =>g admin -m sam #Create user "sam"
#userdel sam        =>Delete user "sam"
#adduser sam         =>Add user "sam"
#usermod            =>Modify user "sam"
```

## FILE COMMANDS

```
#ls -al
#pwd               =>Show the path of current directory
#mkdir directory-name =>Create a directory
#rm file-name      =>Delete file
#rm -r directory-name =>Delete directory recursively
#rm -f file-name    =>Forcefully remove file
#rm -rf directory-name =>Forcefully remove directory recursively
#cp file1 file2     =>Copy file1 to file2
#cp -r dir1 dir2     =>Copy dir1 to dir2, create dir2 if it doesn't exist
#mv file1 file2      =>Rename source to dest / move source to directory
#ln -s /path/to/file-name link-name #Create symbolic link to file-name
#touch file         =>Create or update file
#cat > file         =>Place standard input into file
#more file          =>Output contents of file
#head file          =>Output first 10 lines of file
```

```
#hostname -i        =>Lookup local ip address
#wget file           =>Download file
#netstat -tupl       =>List active connections to / from system
```

## COMPRESSION / ARCHIVES

```
#tar cf home.tar home =>Create tar named home.tar containing home/
#tar xf file.tar       =>Extract the files from file.tar
#tar czf file.tar.gz files =>Create a tar with gzip compression
#gzip file             =>Compress file and renames it to file.gz
```

## INSTALL PACKAGE

```
#rpm -i pkgname.rpm  =>Install rpm based package
#rpm -e pkgname       =>Remove package
```

## INSTALL FROM SOURCE

```
#!/configure
#make
#make install
```

## SEARCH

```
#grep pattern files  =>Search for pattern in files
#grep -r pattern dir  =>Search recursively for pattern in dir
#locate file          =>Find all instances of file
#find /home/tom -name 'index*' =>Find files names that start with "index"
#find /home -size +10000k =>Find files larger than 10000k in /home
```

## LOGIN (SSH AND TELNET)

```
#ssh user@host      =>Connect to host as user
```

# The pkill Command (Process Kill)

The `pkill` command is used to find and terminate processes by name. It sends signals to processes by name rather than by PID, as with the `kill` command.



# `pkill` Options and Usage

The `pkill` command has several options that you can use to customize the process search. For example, you can specify a signal other than `SIGTERM` to terminate a process gracefully.

A widely used option for `pkill` is the `-f` flag which allows for searching a process by its full name rather than just the name of its command. To terminate a process by name, use `pkill` or `pkill -f` if you need to match by the full process command.

```
new user: y
: Marquez
:
: (again):
rld

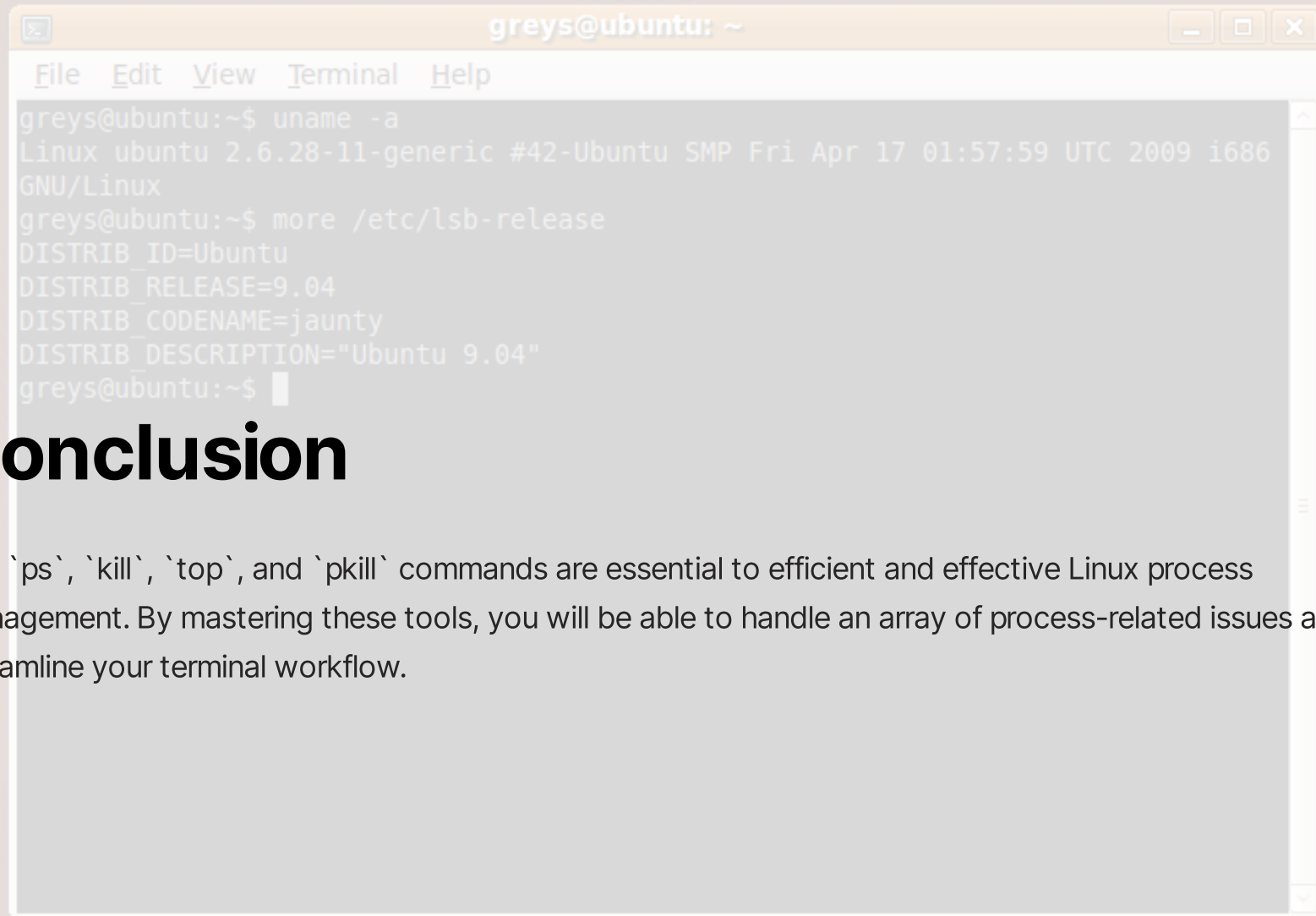
... OK, you are logged in.
> view
: user [enter = yourself]:

edit

: note 1
: my number one note!
quit
again: y

... OK, you are logged in.
> view
: user [enter = yourself]: incal

number one note!
```

A terminal window titled 'greys@ubuntu: ~' with a menu bar (File, Edit, View, Terminal, Help). The terminal shows the output of 'uname -a' and 'more /etc/lsb-release'.

```
greys@ubuntu:~$ uname -a
Linux ubuntu 2.6.28-11-generic #42-Ubuntu SMP Fri Apr 17 01:57:59 UTC 2009 i686
GNU/Linux
greys@ubuntu:~$ more /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=9.04
DISTRIB_CODENAME=jaunty
DISTRIB_DESCRIPTION="Ubuntu 9.04"
greys@ubuntu:~$
```

# Conclusion

The `ps`, `kill`, `top`, and `pkill` commands are essential to efficient and effective Linux process management. By mastering these tools, you will be able to handle an array of process-related issues and streamline your terminal workflow.