

Linux Admin Report

10- Processes

When a system starts up, the kernel initiates a few of its own activities as processes and launches a program called init. init, in turn, runs a series of shell scripts (located in /etc) called init scripts, which start all the system services.

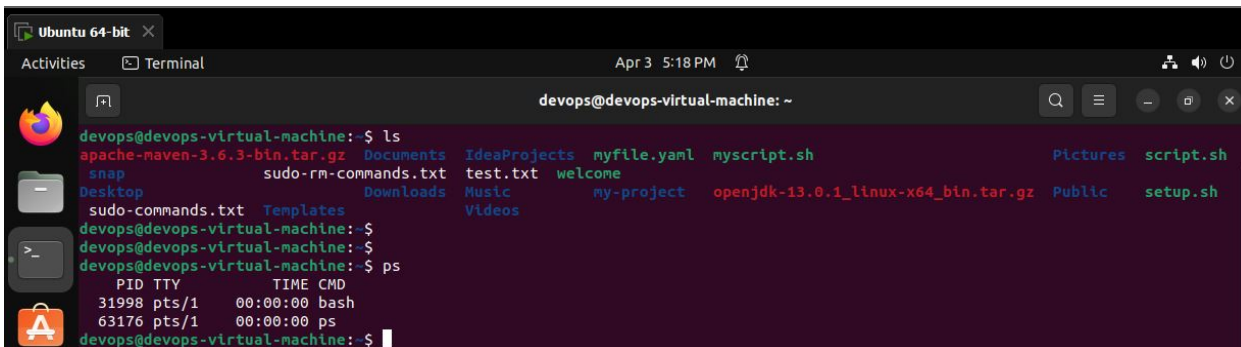
Viewing Processes

:

Let's start simple and show how can I see a snapshot of current processes:

The answer is the <ps> command that stands for process, and It only displays processes associated with the current bash.

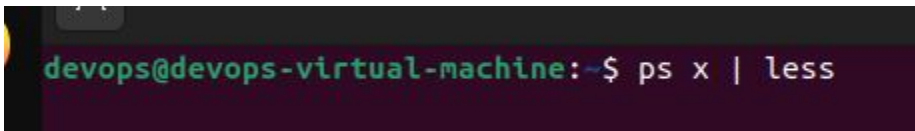
\$ ps



```
devops@devops-virtual-machine:~$ ls
apache-maven-3.6.3-bin.tar.gz  Documents  IdeaProjects  myfile.yaml  myscript.sh  Pictures  script.sh
snap                          sudo-rm-commands.txt  test.txt  welcome
Desktop                        Downloads  Music        my-project   openjdk-13.0.1_linux-x64_bin.tar.gz  Public  setup.sh
sudo-commands.txt  Templates
devops@devops-virtual-machine:~$
devops@devops-virtual-machine:~$ ps
  PID TTY          TIME CMD
 31998 pts/1    00:00:00 bash
 63176 pts/1    00:00:00 ps
devops@devops-virtual-machine:~$
```

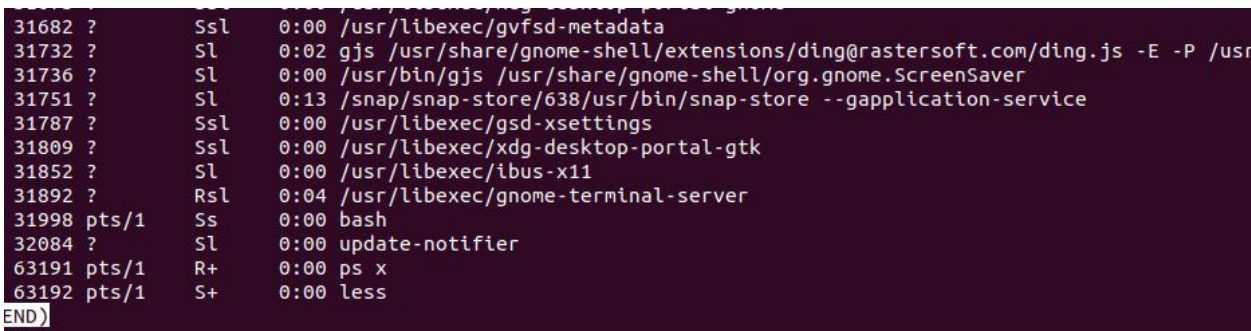
To show all processes regarding what terminal they're controlled by we can use

\$ ps x



```
devops@devops-virtual-machine:~$ ps x | less
```

I used less to handle the output in an easy way



```
31682 ?        Ssl   0:00 /usr/libexec/gvfsd-metadata
31732 ?        Sl    0:02 gjs /usr/share/gnome-shell/extensions/ding@rastersoft.com/ding.js -E -P /usr
31736 ?        Sl    0:00 /usr/bin/gjs /usr/share/gnome-shell/org.gnome.ScreenSaver
31751 ?        Sl    0:13 /snap/snap-store/638/usr/bin/snap-store --gapplication-service
31787 ?        Ssl   0:00 /usr/libexec/gsd-xsettings
31809 ?        Ssl   0:00 /usr/libexec/xdg-desktop-portal-gtk
31852 ?        Sl    0:00 /usr/libexec/ibus-x11
31892 ?        Rsl   0:04 /usr/libexec/gnome-terminal-server
31998 pts/1    Ss    0:00 bash
32084 ?        Sl    0:00 update-notifier
63191 pts/1    R+    0:00 ps x
63192 pts/1    S+    0:00 less
END)
```

This represents PID , TTL , STAT(R-S-I...)

It's important to know that STAT represents process status.

Linux Admin Report

There are different status to represent as follows:

| | |
|---|--|
| R | Running. This means that the process is running or ready to run. |
| S | Sleeping. The process is not running; rather, it is waiting for an event, such as a keystroke or network packet. |
| D | Uninterruptible Sleep. Process is waiting for I/O such as a disk drive. |
| T | Stopped. Process has been instructed to stop. More on this later. |
| Z | A defunct or “zombie” process. This is a child process that has terminated, but has not been cleaned up by its parent. |
| < | A high priority process. |
| N | A low priority process. |

If we need more information we can use:

\$ ps aux

```
devops@devops-virtual-machine:~$ ps aux | less
```

| USER | PID | %CPU | %MEM | VSZ | RSS | TTY | STAT | START | TIME | COMMAND |
|------|-----|------|------|--------|-------|-----|------|-------|------|------------------------------------|
| root | 1 | 0.7 | 0.5 | 168232 | 10520 | ? | Ss | 16:35 | 0:26 | /lib/systemd/systemd auto noprompt |
| root | 2 | 0.0 | 0.0 | 0 | 0 | ? | S | 16:35 | 0:00 | [kthreadd] |
| root | 3 | 0.0 | 0.0 | 0 | 0 | ? | I< | 16:35 | 0:00 | [rcu_gp] |
| root | 4 | 0.0 | 0.0 | 0 | 0 | ? | I< | 16:35 | 0:00 | [rcu_par_gp] |
| root | 5 | 0.0 | 0.0 | 0 | 0 | ? | I< | 16:35 | 0:00 | [netns] |
| root | 7 | 0.0 | 0.0 | 0 | 0 | ? | I< | 16:35 | 0:00 | [kworker/0:0H-events_highpri] |
| root | 9 | 0.1 | 0.0 | 0 | 0 | ? | I< | 16:35 | 0:05 | [kworker/0:1H-events_highpri] |
| root | 10 | 0.0 | 0.0 | 0 | 0 | ? | I< | 16:35 | 0:00 | [mm_percpu_wq] |
| root | 11 | 0.0 | 0.0 | 0 | 0 | ? | S | 16:35 | 0:00 | [rcu_tasks_rude_] |
| root | 12 | 0.0 | 0.0 | 0 | 0 | ? | S | 16:35 | 0:00 | [rcu_tasks_trace] |

| | |
|-------|---|
| USER | User ID. This is the owner of the process. |
| %CPU | CPU usage in percent. |
| %MEM | Memory usage in percent. |
| VSZ | Virtual memory size. |
| RSS | Resident Set Size. The amount of physical memory (RAM) the process is using in kilobytes. |
| START | Time when the process started. |

Linux Admin Report

```
root      63271  0.0  0.2 21924  4732 pts/1    S   17:29   0:00 su devops
devops    63272  0.0  0.2 20012  5476 pts/1    S   17:29   0:00 bash
root      63279  0.0  0.0      0      0 ?        I   17:29   0:00 [kworker/u256:3-events_freezable_power_]
root      63283  0.0  0.0      0      0 ?        I   17:31   0:00 [kworker/1:1-events]
devops    63308  0.0  0.1 21416  3476 pts/1    R+  17:35   0:00 ps aux
devops    63309  0.0  0.1 17500  2744 pts/1    S+  17:35   0:00 less
(END)
```

As we can see in the STAT column we have <ps aux> command running, <less> command in a sleeping state.

We can display each state used in <ps> command by using <man> command:

\$ man ps

```
devops@devops-virtual-machine:~$ man ps
```

After some navigation we get:

```
devops@devops-virtual-machine: ~
PROCESS STATE CODES
Here are the different values that the s, stat and state output specifiers (header "STAT" or "S") will display to
describe the state of a process:

D    uninterruptible sleep (usually IO)
I    Idle kernel thread
R    running or runnable (on run queue)
S    interruptible sleep (waiting for an event to complete)
T    stopped by job control signal
t    stopped by debugger during the tracing
W    paging (not valid since the 2.6.xx kernel)
X    dead (should never be seen)
Z    defunct ("zombie") process, terminated but not reaped by its parent

For BSD formats and when the stat keyword is used, additional characters may be displayed:

<    high-priority (not nice to other users)
N    low-priority (nice to other users)
L    has pages locked into memory (for real-time and custom IO)
s    is a session leader
l    is multi-threaded (using CLONE_THREAD, like NPTL pthreads do)
+    is in the foreground process group
```

It's always very useful to use <man>!

Now we displayed only a snapshot of the processes, what if we want a more dynamic way?
Fortunately we have a command for a continuously updating display of the system processes listed in order of process activity.

The top display consists of two parts: a system summary at the top of the display, followed by a table of processes sorted by CPU activity:

\$ top

```
devops@devops-virtual-machine:~$ top
```

Linux Admin Report

```
devops@devops-virtual-machine: ~
top - 17:52:59 up 1:17, 1 user, load average: 1.90, 1.50, 0.87
Tasks: 310 total, 2 running, 307 sleeping, 0 stopped, 1 zombie
%Cpu(s): 0.2 us, 0.2 sy, 0.2 ni, 50.4 id, 49.1 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 1941.6 total, 451.4 free, 985.1 used, 505.1 buff/cache
MiB Swap: 2140.0 total, 1700.1 free, 439.9 used, 764.7 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 31202 devops    20   0 4422072 144104 36652 S   0.7   7.2   1:01.81 gnome-shell
   749 root       20   0 252492  4324  3544 S   0.3   0.2   0:17.28 vmtoolsd
 31892 devops    20   0 576368  31540 22204 S   0.3   1.6   0:09.07 gnome-terminal-
 63490 root       20   0      0      0      0 I   0.3   0.0   0:00.21 kworker/0:0-events
 65223 root       20   0      0      0      0 I   0.3   0.0   0:00.16 kworker/1:0-events
 68551 devops    20   0 22164  4024  3164 R   0.3   0.2   0:03.17 top
     1 root       20   0 168232  9948  5300 S   0.0   0.5   0:28.57 systemd
     2 root       20   0      0      0      0 S   0.0   0.0   0:00.03 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
     5 root       0 -20      0      0      0 I   0.0   0.0   0:00.00 netns
     7 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   0:07.77 kworker/0:1H-events_highpri
    10 root       0 -20      0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
    11 root       20   0      0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_rude_
    12 root       20   0      0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_trace
    13 root       20   0      0      0      0 S   0.0   0.0   0:02.66 ksoftirqd/0
    14 root       20   0      0      0      0 R   0.0   0.0   0:08.47 rcu_sched
    15 root       rt    0      0      0      0 S   0.0   0.0   0:00.11 migration/0
    16 root      -51   0      0      0      0 S   0.0   0.0   0:00.00 idle_inject/0
    18 root       20   0      0      0      0 S   0.0   0.0   0:00.10 cpuhp/0
    19 root       20   0      0      0      0 S   0.0   0.0   0:00.05 cpuhp/1
    20 root      -51   0      0      0      0 S   0.0   0.0   0:00.00 idle_inject/1
```

The system summary contains a lot of good stuff. Here's a rundown:

| | | |
|-------|--------------|--|
| Row 1 | top | Name of the program. |
| | 17:52:59 | Current time of day. |
| | Up 1:17 | The amount of time since the machine was last booted |
| | 1 user | There is one user logged in. |
| | load average | Refers to the number of processes that are waiting to run. |
| Row 2 | Tasks | This summarizes the number of processes and their various process states. |
| Row 3 | Cpu(s): | This row describes the character of the activities that the CPU is performing. |
| Row 4 | Mem: | Shows how physical RAM is being used. |
| Row 5 | Swap: | Shows how swap space (virtual memory) is being used. |

As we can see, the <top> command is very useful as it refreshes itself every some fixed time, and the system summary can help us a lot to get more information about our resources usage.

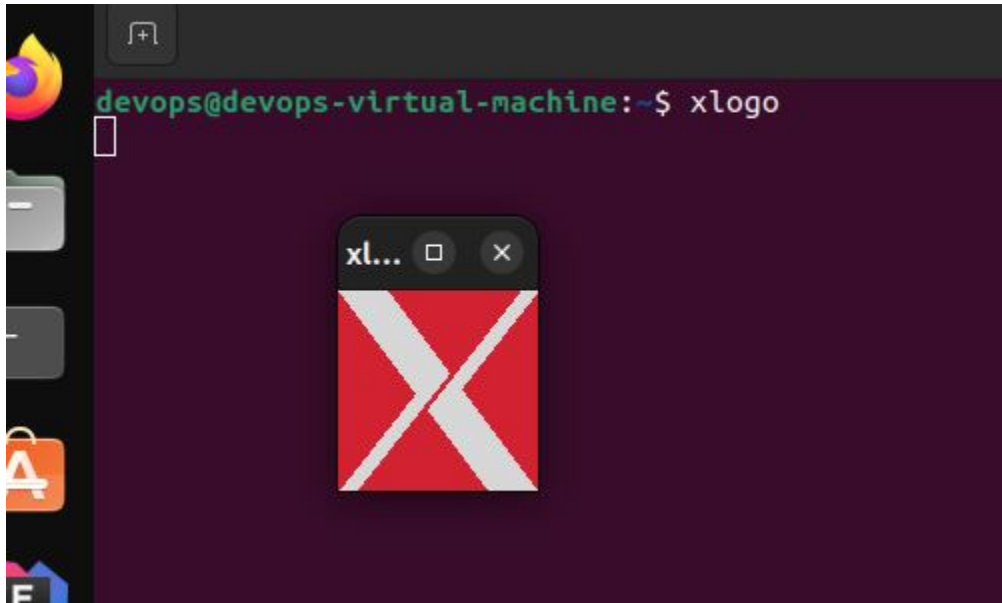
This is how we can view processes in different ways statically and dynamically using different commands, the next part will be about how to control it!

Linux Admin Report

Controlling Processes

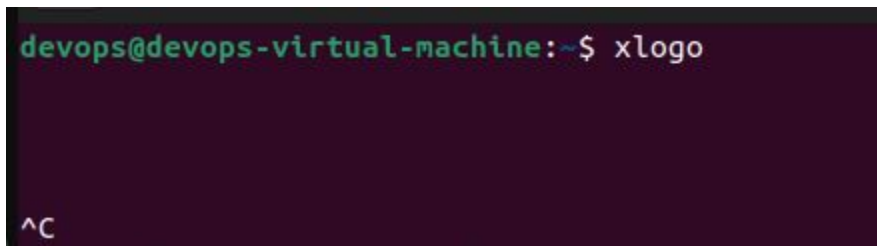
Running a process to test on (the command we'll be using in this section will be <xlogo>)

\$ xlogo



After running this we can't use the same terminal as it's running in foreground. What we should do to resolve this is to interrupt this process

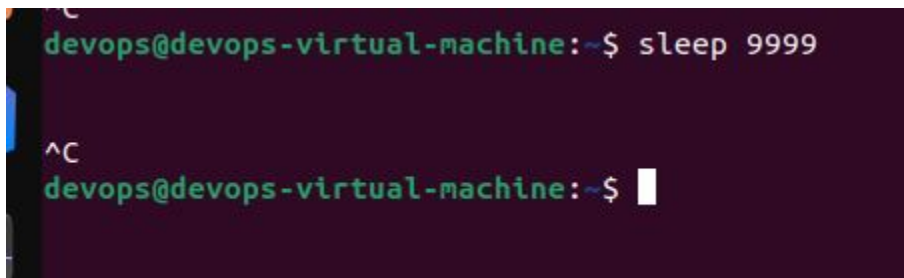
Ctrl + c



Now we terminated the process and got back to terminal use successfully!

Another example is to interrupt a sleeping process

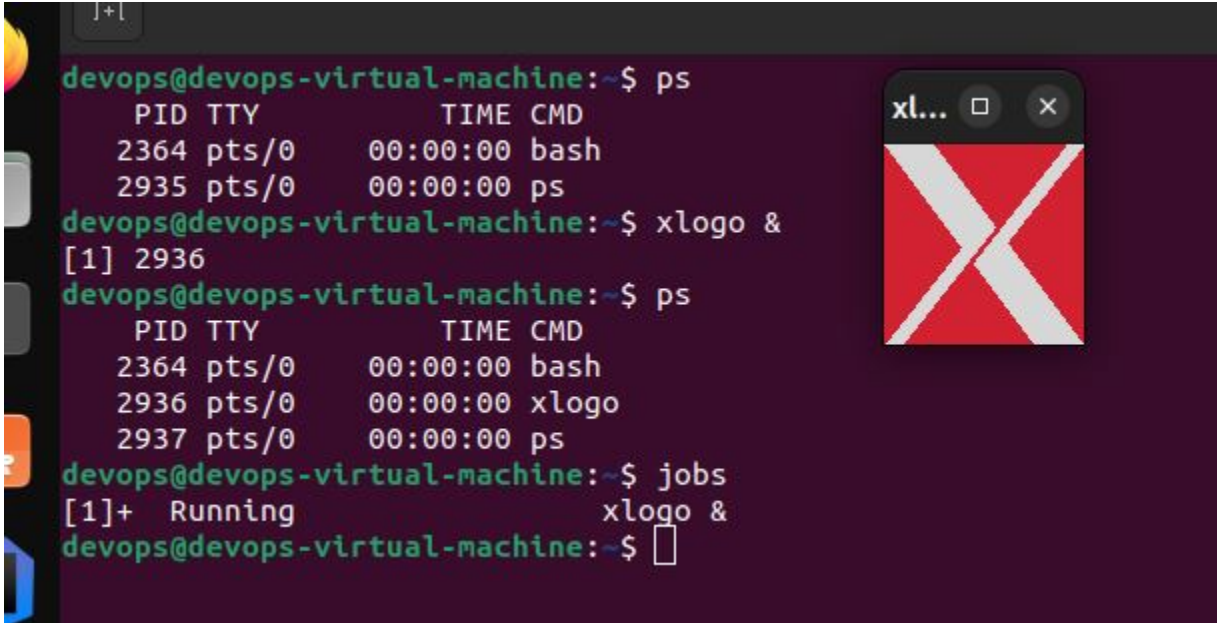
\$ sleep 9999



Linux Admin Report

Let's say we wanted to get the shell prompt back without terminating the <xlogo> program. We'll do this by placing the program in the background.

\$ xlogo &

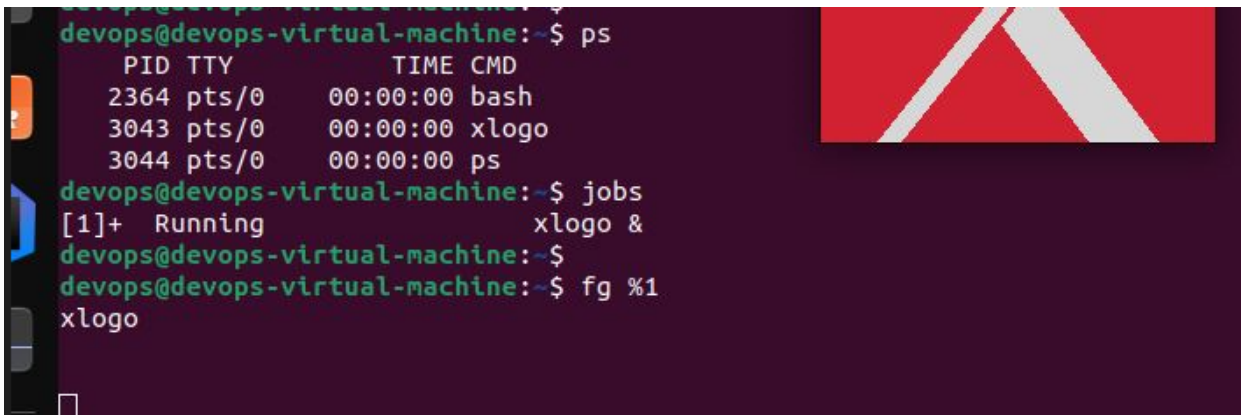
A terminal window with a dark purple background. The prompt is 'devops@devops-virtual-machine:~\$'. The user runs 'ps', showing a table with columns PID, TTY, TIME, and CMD. The first two rows are PID 2364 (bash) and PID 2935 (ps). Then the user runs 'xlogo &', which returns '[1] 2936'. The user runs 'ps' again, and the table now includes PID 2936 (xlogo) and PID 2937 (ps). Finally, the user runs 'jobs', which shows '[1]+ Running xlogo &'.

| PID | TTY | TIME | CMD |
|------|-------|----------|-------|
| 2364 | pts/0 | 00:00:00 | bash |
| 2935 | pts/0 | 00:00:00 | ps |
| 2936 | pts/0 | 00:00:00 | xlogo |
| 2937 | pts/0 | 00:00:00 | ps |

As we can see the process is running but in the background.

A process in the background is immune from keyboard input, including any attempt to interrupt it with a Ctrl-c. To return a process to the foreground use:

\$ fg %1 # 1 stand for the job number

A terminal window showing the continuation of the previous session. The prompt is 'devops@devops-virtual-machine:~\$'. The user runs 'ps', and the table now shows PID 3043 (xlogo) and PID 3044 (ps). The user runs 'jobs', which shows '[1]+ Running xlogo &'. Finally, the user runs 'fg %1', and the prompt changes to 'xlogo', indicating the process is now in the foreground.

| PID | TTY | TIME | CMD |
|------|-------|----------|-------|
| 2364 | pts/0 | 00:00:00 | bash |
| 3043 | pts/0 | 00:00:00 | xlogo |
| 3044 | pts/0 | 00:00:00 | ps |

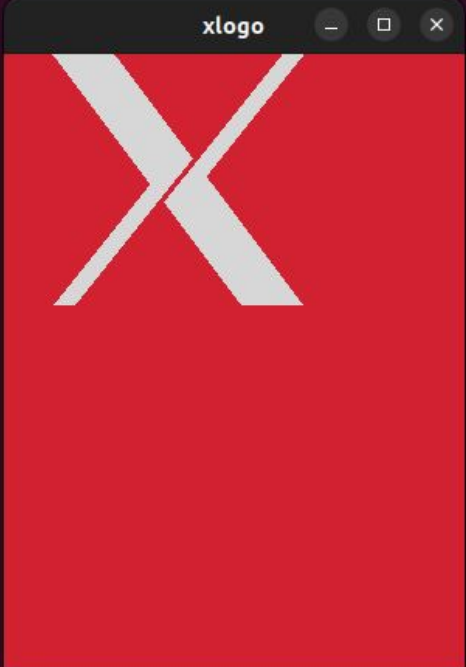
Now we can use another shortcut Ctrl-z to only stop the process.

Linux Admin Report

Ctrl + z

```
devops@devops-virtual-machine:~$ ps
  PID TTY          TIME CMD
 2364 pts/0    00:00:00 bash
 3042 pts/0    00:00:00 ps
devops@devops-virtual-machine:~$
devops@devops-virtual-machine:~$ xlogo &
[1] 3043
devops@devops-virtual-machine:~$
devops@devops-virtual-machine:~$ ps
  PID TTY          TIME CMD
 2364 pts/0    00:00:00 bash
 3043 pts/0    00:00:00 xlogo
 3044 pts/0    00:00:00 ps
devops@devops-virtual-machine:~$ jobs
[1]+  Running                  xlogo &
devops@devops-virtual-machine:~$
devops@devops-virtual-machine:~$ fg %1
xlogo

^Z
[1]+  Stopped                  xlogo
devops@devops-virtual-machine:~$
devops@devops-virtual-machine:~$ jobs
[1]+  Stopped                  xlogo
devops@devops-virtual-machine:~$
```




Notice that when we maximize the logo id does not respond in a meaningful way as the process is not running.

But we can get it back to ward in background as follows:

\$ bg %1

```
devops@devops-virtual-machine:~$ ps
  PID TTY          TIME CMD
 2364 pts/0    00:00:00 bash
 3042 pts/0    00:00:00 ps
devops@devops-virtual-machine:~$
devops@devops-virtual-machine:~$ xlogo &
[1] 3043
devops@devops-virtual-machine:~$
devops@devops-virtual-machine:~$ ps
  PID TTY          TIME CMD
 2364 pts/0    00:00:00 bash
 3043 pts/0    00:00:00 xlogo
 3044 pts/0    00:00:00 ps
devops@devops-virtual-machine:~$ jobs
[1]+  Running                  xlogo &
devops@devops-virtual-machine:~$
devops@devops-virtual-machine:~$ fg %1
xlogo

^Z
[1]+  Stopped                  xlogo
devops@devops-virtual-machine:~$
devops@devops-virtual-machine:~$ jobs
[1]+  Stopped                  xlogo
devops@devops-virtual-machine:~$
devops@devops-virtual-machine:~$ bg %1
[1]+ xlogo &
devops@devops-virtual-machine:~$
devops@devops-virtual-machine:~$
```



Now we can see the difference!

Linux Admin Report

Sending Signals To Processes With kill

\$ kill 3043

3043 is the PID

```
devops@devops-virtual-machine:~$ bg %1
[1]+ xlogo &
devops@devops-virtual-machine:~$
devops@devops-virtual-machine:~$ jobs
[1]+  Running                  xlogo &
devops@devops-virtual-machine:~$ ps
  PID TTY          TIME CMD
 2364 pts/0        00:00:00 bash
 3043 pts/0        00:00:00 xlogo
 3061 pts/0        00:00:00 ps
devops@devops-virtual-machine:~$ kill 3043
devops@devops-virtual-machine:~$ ps
  PID TTY          TIME CMD
 2364 pts/0        00:00:00 bash
 3062 pts/0        00:00:00 ps
[1]+  Terminated              xlogo
devops@devops-virtual-machine:~$ jobs
devops@devops-virtual-machine:~$
```

Kill by default uses a terminate signal, but we can also use different signals:

\$kill [-signal] PID

```
devops@devops-virtual-machine:~$ ps
  PID TTY          TIME CMD
 2364 pts/0        00:00:00 bash
 3066 pts/0        00:00:00 ps
devops@devops-virtual-machine:~$ xlogo &
[1] 3068
devops@devops-virtual-machine:~$ ps
  PID TTY          TIME CMD
 2364 pts/0        00:00:00 bash
 3068 pts/0        00:00:00 xlogo
 3069 pts/0        00:00:00 ps
devops@devops-virtual-machine:~$ jobs
[1]+  Running                  xlogo &
devops@devops-virtual-machine:~$ kill -STOP 3068
[1]+  Stopped                  xlogo
devops@devops-virtual-machine:~$ kill -CONT 3068
devops@devops-virtual-machine:~$ jobs
[1]+  Running                  xlogo &
devops@devops-virtual-machine:~$ kill -INT 3068
[1]+  Interrupt                 xlogo
devops@devops-virtual-machine:~$ ps
  PID TTY          TIME CMD
 2364 pts/0        00:00:00 bash
 3072 pts/0        00:00:00 ps
devops@devops-virtual-machine:~$ jobs
devops@devops-virtual-machine:~$
```


Linux Admin Report

We can see all the signals we can use using:

\$ kill -l

```
devops@devops-virtual-machine:~$ kill -l
1) SIGHUP      2) SIGINT      3) SIGQUIT     4) SIGILL      5) SIGTRAP
6) SIGABRT     7) SIGBUS     8) SIGFPE     9) SIGKILL     10) SIGUSR1
11) SIGSEGV    12) SIGUSR2    13) SIGPIPE    14) SIGALRM     15) SIGTERM
16) SIGSTKFLT  17) SIGCHLD    18) SIGCONT     19) SIGSTOP     20) SIGTSTP
21) SIGTTIN    22) SIGTTOU    23) SIGURG     24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM  27) SIGPROF    28) SIGWINCH    29) SIGIO       30) SIGPWR
31) SIGSYS     34) SIGRTMIN   35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
63) SIGRTMAX-1 64) SIGRTMAX
devops@devops-virtual-machine:~$
```

Sending Signals To Multiple Processes With killall

It's also possible to send signals to multiple processes matching a specified program or username by using the killall command.

\$ killall [-u user] [-signal] name

```
devops@devops-virtual-machine:~$ ps
  PID TTY          TIME CMD
 2364 pts/0      00:00:00 bash
 3104 pts/0      00:00:00 ps
devops@devops-virtual-machine:~$ jobs
devops@devops-virtual-machine:~$ xlogo &
[1] 3105
devops@devops-virtual-machine:~$ xlogo &
[2] 3106
devops@devops-virtual-machine:~$ killall -STOP xlogo
[1]-  Stopped                  xlogo
[2]+  Stopped                  xlogo
devops@devops-virtual-machine:~$ jobs
[1]-  Stopped                  xlogo
[2]+  Stopped                  xlogo
devops@devops-virtual-machine:~$ killall xlogo
devops@devops-virtual-machine:~$ ps
  PID TTY          TIME CMD
 2364 pts/0      00:00:00 bash
 3105 pts/0      00:00:00 xlogo
 3106 pts/0      00:00:00 xlogo
 3113 pts/0      00:00:00 ps
devops@devops-virtual-machine:~$ jobs
[1]-  Stopped                  xlogo
[2]+  Stopped                  xlogo
devops@devops-virtual-machine:~$ killall -CONT xlogo
[1] Terminated                xlogo
[2]+ Terminated                xlogo
devops@devops-virtual-machine:~$ ps
  PID TTY          TIME CMD
 2364 pts/0      00:00:00 bash
 3115 pts/0      00:00:00 ps
devops@devops-virtual-machine:~$ jobs
devops@devops-virtual-machine:~$
```

Linux Admin Report

As Shown above, Processes can't be terminated when they're in a stop status, but once we run them again the terminated signal gets activated

\$ vmstat

```
devops@devops-virtual-machine:~$ vmstat 3
procs -----memory----- --swap-- ----io---- -system-- -----cpu-----
r  b   swpd   free   buff  cache   si   so    bi    bo    in   cs   us   sy   id   wa   st
0  0   416280  86588   6828 175900   42  116   406   137   104  175   1   3  95   2   0
1  0   416280  86588   6828 175940    0    0    0    0   106  200   0  0 100   0   0
0  0   416280  86588   6836 175932    0    0    0    4   106  183   0  0 100   0   0
0  0   416280  86588   6836 175940    0    0    0    0    97  175   0  0 100   0   0
0  0   416280  86588   6836 175940    0    0    0    1    99  178   0  0 100   0   0
0  0   416280  86588   6836 175940    0    0    0    0   103  188   0  0 100   0   0
1  0   416280  86588   6836 175940    0    0    0    0    98  179   0  0 100   0   0
0  0   416280  86588   6836 175940    0    0    0    0   100  182   0  0 100   0   0
0  0   416280  86588   6836 175940    0    0    0    0   100  178   0  0 100   0   0
^C
devops@devops-virtual-machine:~$
```

Outputs a snapshot of system resource usage including, memory, swap and disk I/O. To see a continuous display, follow the command with a time delay (in seconds) for updates. For example: <vmstat 5>. Terminate the output with Ctrl-c.

Linux Admin Report

11- The Environment

The shell maintains a body of information during our shell session called the environment. Data stored in the environment is used by programs to determine facts about our configuration.

Examining The Environment

To see what is stored in the environment, we can use either the set builtin in bash or the <printenv> program. The set command will show both the shell and environment variables, while <printenv> will only display the latter.

\$ printenv | less

```
devops@devops-virtual-machine:~$ printenv | less
SHELL=/bin/bash
SESSION_MANAGER=local/devops-virtual-machine:@/tmp/.ICE-unix/1733,unix/devops-virtual-machine:/tmp/.ICE-unix/1733
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
SSH_AGENT_LAUNCHER=gnome-keyring
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
LANGUAGE=en_US:en
LC_ADDRESS=en_GB.UTF-8
GNOME_SHELL_SESSION_MODE=ubuntu
LC_NAME=en_GB.UTF-8
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
LC_MONETARY=en_GB.UTF-8
GTK_MODULES=gail:atk-bridge
PWD=/home/devops
LOGNAME=devops
```

We can see clearly different known environmental variables just as:
(SHELL - DESKTOP_SESSION - PWD - LOGNAME)

The <printenv> command can also list the value of a specific variable:

\$ printenv SHELL LOGNAME

```
devops@devops-virtual-machine:~$ printenv SHELL LOGNAME
/bin/bash
devops
devops@devops-virtual-machine:~$
```

Linux Admin Report

The set command, when used without options or arguments, will display both the shell and environment variables:

```
$ set | less
```

```
devops@devops-virtual-machine:~$ set | less
BASH=/bin/bash
BASHOPTS=checkwinsize:cmdhist:complete_fullquote:expand_aliases:extglob:extquote:force
BASH_ALIASES=()
BASH_ARGC=([0]="0")
BASH_ARGV=()
BASH_CMDS=()
BASH_COMPLETION_VERSION=([0]="2" [1]="11")
BASH_LINENO=()
BASH_SOURCE=()
BASH_VERSION=([0]="5" [1]="1" [2]="16" [3]="1" [4]="release" [5]="x86_64-pc-linux-gnu")
BASH_VERSION='5.1.16(1)-release'
COLORTERM=truecolor
COLUMNS=203
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
DESKTOP_SESSION=ubuntu
DIRSTACK=()
```

As we can see it's sorted and more detailed and contains the **DESKTOP_SESSION** as shown before.

One element of the environment that neither <set> nor <printenv> displays is aliases:

```
$ alias
```

```
alias ls='ls --color=auto'
devops@devops-virtual-machine:~$ alias foo='pwd'
devops@devops-virtual-machine:~$ foo
/home/devops
devops@devops-virtual-machine:~$ alias
alias alert='notify-send --urgency=low -i "${[ $? = 0 ]} && echo terminal || echo error)" "$'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias foo='pwd'
alias grep='grep --color=auto'
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -alF'
alias ls='ls --color=auto'
devops@devops-virtual-machine:~$
```

As shown we can make our own alias!

Linux Admin Report

How Is The Environment Established?

When we log on to the system, the bash program starts, and reads a series of configuration scripts called startup files, which define the default environment shared by all users. This is followed by more startup files in our home directory that define our personal environment.

A startup file is an executable script to run mostly by the system during starting of the OS.

Modifying The Environment

Since we know where the startup files are and what they contain, we can modify them to customize our environment.

As a general rule, to add directories to your PATH, or define additional environment variables, place those changes in .profile (for Ubuntu). For everything else, place the changes in .bashrc.

\$ vim ~/.bashrc

```
devops@devops-virtual-machine:~$ pwd
/home/devops
devops@devops-virtual-machine:~$ ls -a
.                .bash_history   .config         .gitcon
..               .bash_logout    Desktop         .gnupg
.android         .bashrc         Documents       .gradle
apache-maven-3.6.3-bin.tar.gz .cache          Downloads      IdeaPro
devops@devops-virtual-machine:~$
devops@devops-virtual-machine:~$ vim .bashrc
```

```
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

export GIT_EDITOR=vim

JAVA_HOME='/opt/jdk-13.0.1'
PATH="$JAVA_HOME/bin:$PATH"
export PATH

M2_HOME='/opt/apache-maven-3.6.3'
PATH="$M2_HOME/bin:$PATH"
export PATH
```

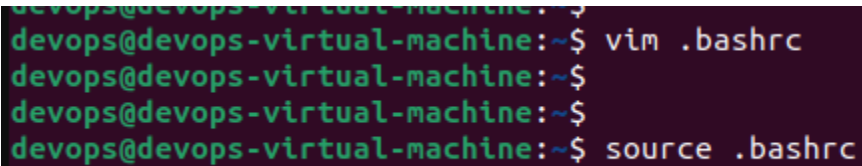
Linux Admin Report

As shown above I created JAVA_HOME and added it to the path.

Activating Our Changes

The changes we have made to our .bashrc will not take effect until we close our terminal session and start a new one, since the .bashrc file is only read at the beginning of a session. However, we can force bash to re-read the modified .bashrc file with the following command:

\$ source .bashrc

A terminal window screenshot with a dark background and green text. It shows a series of commands and prompts. The first line is partially cut off. The second line shows the prompt 'devops@devops-virtual-machine:~\$' followed by the command 'vim .bashrc'. The third and fourth lines show the prompt 'devops@devops-virtual-machine:~\$' followed by a blank line. The fifth line shows the prompt 'devops@devops-virtual-machine:~\$' followed by the command 'source .bashrc'.

```
devops@devops-virtual-machine:~$ vim .bashrc
devops@devops-virtual-machine:~$
devops@devops-virtual-machine:~$
devops@devops-virtual-machine:~$ source .bashrc
```

Linux Admin Report

14- Package Management

The most important determinant of distribution quality is the packaging system and the vitality of the distribution's support community. As we spend more time with Linux, we see that its software landscape is extremely dynamic. Things are constantly changing. Most of the top-tier Linux distributions release new versions every six months and many individual program updates every day. To keep up with this blizzard of software, we need good tools for package management.

Package management is a method of installing and maintaining software on the system.

Packaging Systems

Different distributions use different packaging systems and as a general rule, a package intended for one distribution is not compatible with another distribution. Most distributions fall into one of two camps of packaging technologies: the **Debian “.deb”** camp and the **Red Hat “.rpm”** camp.

| | |
|----------------------|---|
| Debian Style (.deb) | Debian, Ubuntu, Xandros, Linspire |
| Red Hat Style (.rpm) | Fedora, CentOS, Red Hat Enterprise Linux, OpenSUSE, Mandriva, PCLinuxOS |

Virtually all software for a Linux system will be found on the Internet. Most of it will be provided by the distribution vendor in the form of package files and the rest will be available in source code form that can be installed manually.

Package Files

The basic unit of software in a packaging system is the package file. A package file is a compressed collection of files that comprise the software package. A package may consist of numerous programs and data files that support the programs. In addition to the files to be installed, the package file also includes metadata about the package.

Repositories

Most packages today are created by the distribution vendors and interested third parties. Packages are made available to the users of a distribution in central repositories that may contain many thousands of packages, each specially built and maintained for the distribution.

Linux Admin Report

A distribution may maintain several different repositories for different stages of the software development life cycle.

Dependencies

Programs seldom “standalone”; rather they rely on the presence of other software components to get their work done. Common activities, such as input/output for example, are handled by routines shared by many programs. These routines are stored in what are called shared libraries, which provide essential services to more than one program. If a package requires a shared resource such as a shared library, it is said to have a dependency. Modern package management systems all provide some method of dependency resolution to ensure that when a package is installed, all of its dependencies are installed, too.

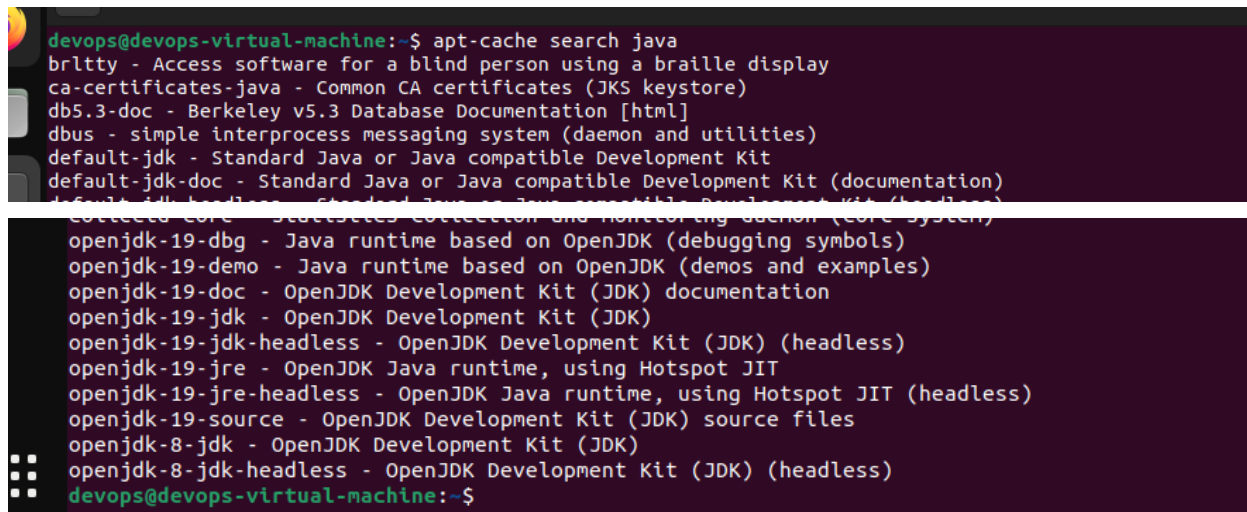
High And Low-level Package Tools

Package management systems usually consist of two types of tools: low-level tools which handle tasks such as installing and removing package files, and high-level tools that perform metadata searching and dependency resolution.

Finding A Package In A Repository

Notice that I'm using Ubuntu

\$ apt-cache search java

A terminal window with a dark purple background and light green text. The prompt is 'devops@devops-virtual-machine:~\$'. The command entered is 'apt-cache search java'. The output lists various Java-related packages and their descriptions, including 'brltty', 'ca-certificates-java', 'db5.3-doc', 'dbus', 'default-jdk', 'default-jdk-doc', 'default-jdk-headless', 'default-jdk-headless-doc', 'openjdk-19-dbg', 'openjdk-19-demo', 'openjdk-19-doc', 'openjdk-19-jdk', 'openjdk-19-jdk-headless', 'openjdk-19-jre', 'openjdk-19-jre-headless', 'openjdk-19-source', 'openjdk-8-jdk', and 'openjdk-8-jdk-headless'.

```
devops@devops-virtual-machine:~$ apt-cache search java
brltty - Access software for a blind person using a braille display
ca-certificates-java - Common CA certificates (JKS keystore)
db5.3-doc - Berkeley v5.3 Database Documentation [html]
dbus - simple interprocess messaging system (daemon and utilities)
default-jdk - Standard Java or Java compatible Development Kit
default-jdk-doc - Standard Java or Java compatible Development Kit (documentation)
default-jdk-headless - Standard Java or Java compatible Development Kit (headless)
default-jdk-headless-doc - Standard Java or Java compatible Development Kit (headless) (documentation)
openjdk-19-dbg - Java runtime based on OpenJDK (debugging symbols)
openjdk-19-demo - Java runtime based on OpenJDK (demos and examples)
openjdk-19-doc - OpenJDK Development Kit (JDK) documentation
openjdk-19-jdk - OpenJDK Development Kit (JDK)
openjdk-19-jdk-headless - OpenJDK Development Kit (JDK) (headless)
openjdk-19-jre - OpenJDK Java runtime, using Hotspot JIT
openjdk-19-jre-headless - OpenJDK Java runtime, using Hotspot JIT (headless)
openjdk-19-source - OpenJDK Development Kit (JDK) source files
openjdk-8-jdk - OpenJDK Development Kit (JDK)
openjdk-8-jdk-headless - OpenJDK Development Kit (JDK) (headless)
devops@devops-virtual-machine:~$
```

Installing A Package From A Repository

High-level tools permit a package to be downloaded from a repository and installed with full dependency resolution.

Linux Admin Report

\$ apt-get update

```
devops@devops-virtual-machine:~$ apt-get update
Reading package lists... Done
E: Could not open lock file /var/lib/apt/lists/lock - open (13: Permission denied)
E: Unable to lock directory /var/lib/apt/lists/
W: Problem unlinking the file /var/cache/apt/pkgcache.bin - RemoveCaches (13: Permission denied)
W: Problem unlinking the file /var/cache/apt/srcpkgcache.bin - RemoveCaches (13: Permission denied)
devops@devops-virtual-machine:~$
devops@devops-virtual-machine:~$ sudo apt-get update
[sudo] password for devops:
Hit:1 http://eg.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 http://eg.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:4 http://eg.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Get:5 http://eg.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [990 kB]
Get:6 http://eg.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [467 kB]
Fetched 1,794 kB in 4s (501 kB/s)
Reading package lists... Done
devops@devops-virtual-machine:~$
```

\$ sudo apt-get install emacs

```
devops@devops-virtual-machine:~$ sudo apt-get install emacs
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2 libllvm13
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  emacs-bin-common emacs-common emacs-el emacs-gtk libm17n-0 libotf1 m17n-db
Suggested packages:
  mailutils emacs-common-non-dfsg ncurses-term m17n-docs gawk
Setting up emacs-gtk (1:27.1+1-3ubuntu5) ...
update-alternatives: using /usr/bin/emacs-gtk to provide /usr/bin/emacs (emacs) in auto mode
Install emacs-common for emacs
emacs-common: Handling install of emacs flavor emacs
Install dictionaries-common for emacs
install/dictionaries-common: Byte-compiling for emacs flavor emacs
Setting up emacs (1:27.1+1-3ubuntu5) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu3) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for install-info (6.8-4build1) ...
Processing triggers for mailcap (3.70+nmu1ubuntu1) ...
Processing triggers for desktop-file-utils (0.26-1ubuntu3) ...
devops@devops-virtual-machine:~$
```

Removing A Package

Packages can be uninstalled using either the high-level or low-level tools. The high-level tools are shown below

Linux Admin Report

\$ sudo apt-get remove emacs

```
devops@devops-virtual-machine:~$ sudo apt-get remove emacs
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  emacs-bin-common emacs-common emacs-el emacs-gtk libflashrom1 libftdi1-2 lib
Use 'sudo apt autoremove' to remove them.
The following packages will be REMOVED:
  emacs
0 upgraded, 0 newly installed, 1 to remove and 12 not upgraded.
After this operation, 78.8 kB disk space will be freed.
Do you want to continue? [Y/n] Y
(Reading database ... 212725 files and directories currently installed.)
Removing emacs (1:27.1+1-3ubuntu5) ...
```

Updating Packages From A Repository

The most common package management task is keeping the system up-to-date with the latest packages. The high-level tools can perform this vital task in one single step:

\$ sudo apt-get update; sudo apt-get upgrade

```
devops@devops-virtual-machine:~$ sudo apt-get update; sudo apt-get upgrade
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:2 http://eg.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://eg.archive.ubuntu.com/ubuntu jammy-updates InRelease
Get:4 http://eg.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Fetched 218 kB in 2s (123 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  emacs-bin-common emacs-common emacs-el emacs-gtk libflashrom1 libftdi1-2 libllvm13 libm17n-0 libotf1
Use 'sudo apt autoremove' to remove them.
The following packages have been kept back:
  gnome-remote-desktop gnome-settings-daemon gnome-settings-daemon-common grub-efi-amd64-bin grub-efi-a
  pulseaudio-utils shin-signed
0 upgraded, 0 newly installed, 0 to remove and 12 not upgraded.
devops@devops-virtual-machine:~$
devops@devops-virtual-machine:~$
```

Listing Installed Packages

This command can be used to display a list of all the packages installed on the system:

\$ dpkg --list

```
devops@devops-virtual-machine:~$ dpkg --list
```

Linux Admin Report

```
devops@devops-virtual-machine:~$ dpkg-query -f='${Package} ${Version} ${Architecture} ${Description}\n'
```

| Name | Version | Architecture | Description |
|--------------------|----------------------|--------------|---|
| accountsservice | 22.07.5-2ubuntu1.3 | amd64 | query and manipulate user account information |
| acl | 2.3.1-1 | amd64 | access control list - utilities |
| acpi-support | 0.144 | amd64 | scripts for handling many ACPI events |
| acpid | 1:2.0.33-1ubuntu1 | amd64 | Advanced Configuration and Power Interface event daemon |
| adduser | 3.118ubuntu5 | all | add and remove users and groups |
| adwaita-icon-theme | 41.0-1ubuntu1 | all | default icon theme of GNOME (small subset) |
| aisleriot | 1:3.22.22-1 | amd64 | GNOME solitaire card game collection |
| alsa-base | 1.0.25+dfsg-0ubuntu7 | all | ALSA driver configuration files |
| alsa-topology-conf | 1.2.5.1-2 | all | ALSA topology configuration files |
| alsa-ucm-conf | 1.2.6.3-1ubuntu1.4 | all | ALSA Use Case Manager configuration files |
| alsa-utils | 1.2.6-1ubuntu1 | amd64 | Utilities for configuring and using ALSA |
| alsa-plugins-usb | 1.2.6-1ubuntu1 | amd64 | ALSA plugins for USB devices |

Determining If A Package Is Installed

This low-level tools can be used to display whether a specified package is installed:

\$ dpkg --status emacs

```
devops@devops-virtual-machine:~$ dpkg --status emacs
dpkg-query: package 'emacs' is not installed and no information is available
Use dpkg --info (= dpkg-deb --info) to examine archive files.
devops@devops-virtual-machine:~$
```

Displaying Info About An Installed Package

If the name of an installed package is known, the following command can be used to display a description of the package:

\$ apt-cache show adduser

```
devops@devops-virtual-machine:~$ apt-cache show adduser
Package: adduser
Architecture: all
Version: 3.118ubuntu5
Multi-Arch: foreign
Priority: important
Section: admin
Origin: Ubuntu
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Debian Adduser Developers <adduser@packages.debian.org>
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Installed-Size: 608
Depends: passwd, debconf (>= 0.5) | debconf-2.0
Suggests: liblocale-gettext-perl, perl, ecryptfs-utils (>= 67-1)
Filename: pool/main/a/adduser/adduser_3.118ubuntu5_all.deb
Size: 155528
MD5sum: d1971ca864f2f9062e81ea59e94fc5fd
SHA1: 0258792290ed3a8d00fab5c5c2ee00afac90a204
SHA256: 53c30389227d4d16f1108a1df7ae1540c890378318335d1dc7ee95b7257f1f17
SHA512: e262a604903de1daa9fe24c52f2fe5ac04274a5c9a73ab0d974adcb541087188b554e515e0c85dad8cc47c1cf32207
Description-en: add and remove users and groups
This package includes the 'adduser' and 'deluser' commands for creating
and removing users.
.
- 'adduser' creates new users and groups and adds existing users to
existing groups;
- 'deluser' removes users and groups and removes users from a given
group.
```

Linux Admin Report

15 - Storage Media

Linux has amazing capabilities for handling storage devices, whether physical storage, such as hard disks, or network storage, or virtual storage devices like RAID (Redundant Array of Independent Disks) and LVM (Logical Volume Manager).

Viewing A List Of Mounted File Systems

The mount command is used to mount file systems. Entering the command without arguments will display a list of the file systems currently mounted:

\$ mount

```
devops@devops-virtual-machine:~$ mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relatime,size=951668k,nr_inodes=237917,mode=755,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,noexec,relatime,size=198720k,mode=755,inode64)
/dev/sda3 on / type ext4 (rw,relatime,errors=remount-ro)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,inode64)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k,inode64)
cgrou2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursiveprot)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=29,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_in
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,pagesize=2M)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
ramfs on /run/credentials/systemd-sysusers.service type ramfs (ro,nosuid,nodev,noexec,relatime,mode=700)
vmware-vmblock on /run/vmblock-fuse type fuse.vmware-vmblock (rw,relatime,user_id=0,group_id=0,default_permissions,allow_o
/var/lib/napd/snaps/bare_5.snap on /snap/bare/5 type squashfs (ro,nodev,relatime,errors=continue,x-gdu.hide)
```

We can unmount any minted device using the <unmount> command

Determining Device Names

\$ ls /dev

```
devops@devops-virtual-machine:~$ ls /dev/
autofs      cpu_dma_latency  full      loop0      loop18      loop7      null      rtc0      snd      tty11      tty20      tty3
block       cuse            fuse      loop1      loop19      loop8      nvram     sda      sr0      tty12      tty21      tty3
bsg         disk           hidraw0   loop10     loop2       loop9      port      sda1     sr1      tty13      tty22      tty3
btrfs-control dma_heap        hpet      loop11     loop20      loop-control  ppp       sda2     stderr   tty14      tty23      tty3
bus         dmideid        hugepages loop12     loop21      mapper      psaux     sda3     stdin    tty15      tty24      tty3
cdrom       dri            hwrng     loop13     loop22      mclog       ptmx      sg0      stdout   tty16      tty25      tty3
char        ecryptfs       initctl   loop14     loop3       mem         pts       sg1      tty      tty17      tty26      tty3
console     fb0            input     loop15     loop4       midi        random     sg2      tty0     tty18      tty27      tty3
core        fd              kmsg      loop16     loop5       mqueue      rfkill     shm      tty1     tty19      tty28      tty3
cpu         fd0            log       loop17     loop6       net         rtc        snapshot  tty10    tty2      tty29      tty3
devops@devops-virtual-machine:~$
```


Linux Admin Report

Manipulating Partitions With fdisk

The fdisk program allows us to interact directly with disk-like devices (such as hard disk drives and flash drives) at a very low level. With this tool we can edit, delete, and create partitions on the device. To work with our flash drive, we must first unmount it (if needed) and then invoke the fdisk program.

fsck This command scans the file system and checks for errors. If it finds any errors, it will attempt to repair them.

\$ fsck

```
devops@devops-virtual-machine:~$ fsck
fsck from util-linux 2.37.2
e2fsck 1.46.5 (30-Dec-2021)
/dev/sda3 is mounted.

WARNING!!! The filesystem is mounted.  If you continue you ***WILL***
cause ***SEVERE*** filesystem damage.

Do you really want to continue<n>? cancelled!
check aborted.
```

Fdisk is a partition table manipulator

\$ fdisk

```
devops@devops-virtual-machine:~$ fdisk
fdisk: bad usage
Try 'fdisk --help' for more information.
devops@devops-virtual-machine:~$
```

\$ fdisk --help

```
devops@devops-virtual-machine:~$ fdisk --help
Usage:
  fdisk [options] <disk>          change partition table
  fdisk [options] -l [<disk>...]  list partition table(s)

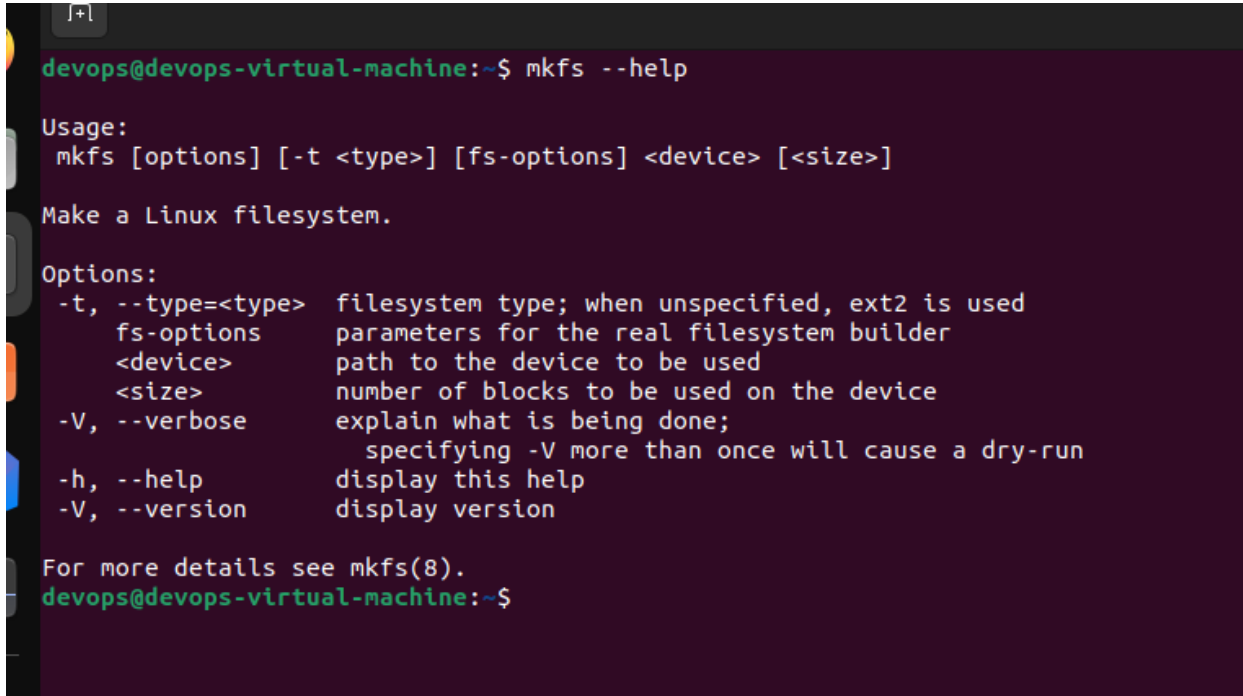
Display or manipulate a disk partition table.

Options:
  -b, --sector-size <size>      physical and logical sector size
  -B, --protect-boot            don't erase bootbits when creating a new label
  -c, --compatibility[=<mode>]  mode is 'dos' or 'nondos' (default)
  -L, --color[=<when>]         colorize output (auto, always or never)
                                colors are enabled by default
  -l, --list                    display partitions and exit
  -x, --list-details            like --list but with more details
  -n, --noauto-pt              don't create default partition table on empty devices
  -o, --output <list>          output columns
  -t, --type <type>            recognize specified partition table type only
  -u, --units[=<unit>]         display units: 'cylinders' or 'sectors' (default)
                                display device size in 512-byte sectors [DEPRECATED]
  -s, --getsz                  set 512 to 4096 bytes rather than the machine-readable format
```

Linux Admin Report

mkfs command typically requires the name of the device or partition that the file system will be created on, as well as the type of file system to create.

\$ mkfs --help



```
devops@devops-virtual-machine:~$ mkfs --help

Usage:
mkfs [options] [-t <type>] [fs-options] <device> [<size>]

Make a Linux filesystem.

Options:
-t, --type=<type>  filesystem type; when unspecified, ext2 is used
fs-options         parameters for the real filesystem builder
<device>          path to the device to be used
<size>            number of blocks to be used on the device
-V, --verbose      explain what is being done;
                   specifying -V more than once will cause a dry-run
-h, --help         display this help
-V, --version      display version

For more details see mkfs(8).
devops@devops-virtual-machine:~$
```

This part is very hard to work with, It's very complicated and I'm only using a virtual machine not a complete system. I'm sorry for any inconvenience I tried to do my best, I hope you understand.

I really spent a good effort in the first three parts so I hope you put that into consideration.

Best Regards.