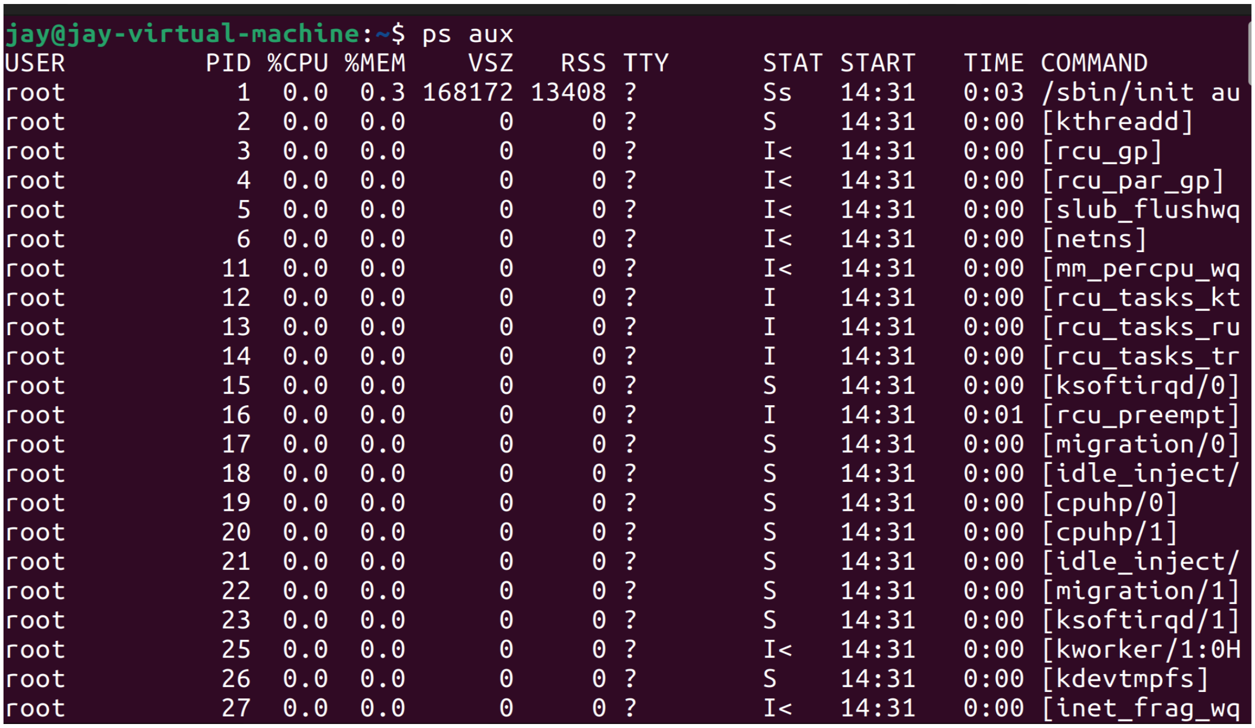
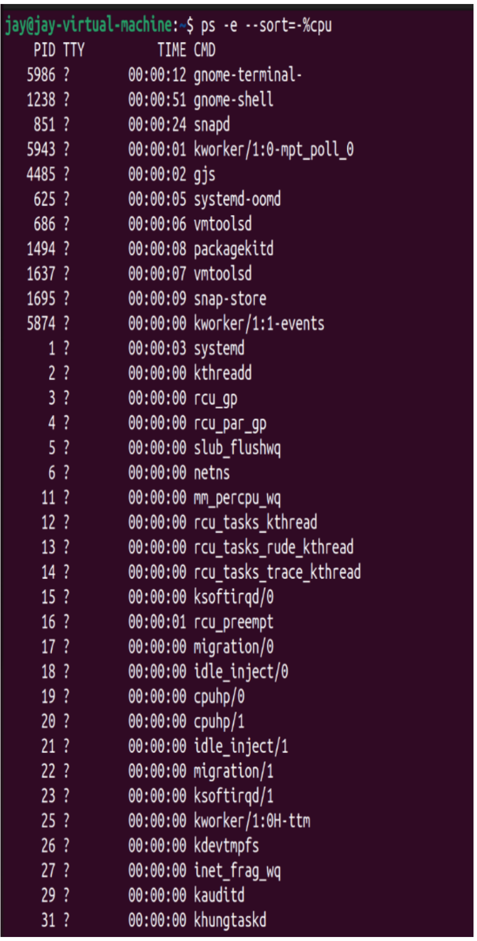
Play with Processes on your Ubuntu VM: Lab 01 Part 1

- 2.

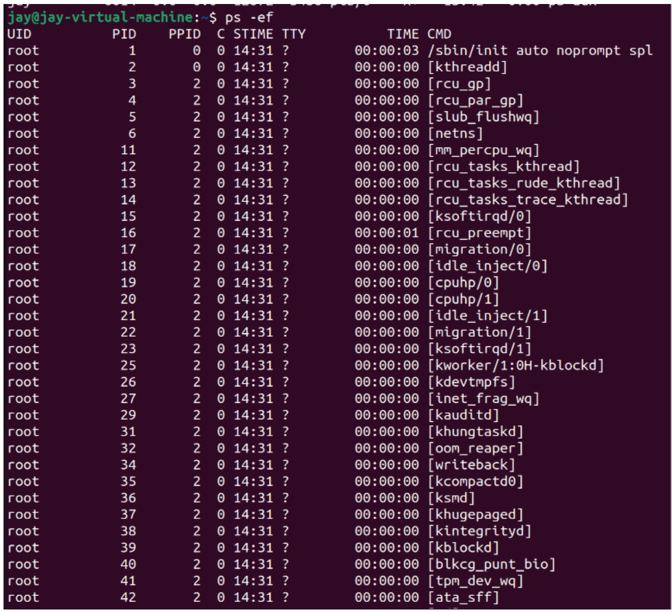


```
jay@jay-virtual-machine:~$ ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.3 168172 13408 ?        Ss   14:31   0:03 /sbin/init au
root           2  0.0  0.0      0     0 ?        S    14:31   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   14:31   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   14:31   0:00 [rcu_par_gp]
root           5  0.0  0.0      0     0 ?        I<   14:31   0:00 [slub_flushwq
root           6  0.0  0.0      0     0 ?        I<   14:31   0:00 [netns]
root          11  0.0  0.0      0     0 ?        I<   14:31   0:00 [mm_percpu_wq
root          12  0.0  0.0      0     0 ?        I    14:31   0:00 [rcu_tasks_kt
root          13  0.0  0.0      0     0 ?        I    14:31   0:00 [rcu_tasks_ru
root          14  0.0  0.0      0     0 ?        I    14:31   0:00 [rcu_tasks_tr
root          15  0.0  0.0      0     0 ?        S    14:31   0:00 [ksoftirqd/0]
root          16  0.0  0.0      0     0 ?        I    14:31   0:01 [rcu_preempt]
root          17  0.0  0.0      0     0 ?        S    14:31   0:00 [migration/0]
root          18  0.0  0.0      0     0 ?        S    14:31   0:00 [idle_inject/
root          19  0.0  0.0      0     0 ?        S    14:31   0:00 [cpuhp/0]
root          20  0.0  0.0      0     0 ?        S    14:31   0:00 [cpuhp/1]
root          21  0.0  0.0      0     0 ?        S    14:31   0:00 [idle_inject/
root          22  0.0  0.0      0     0 ?        S    14:31   0:00 [migration/1]
root          23  0.0  0.0      0     0 ?        S    14:31   0:00 [ksoftirqd/1]
root          25  0.0  0.0      0     0 ?        I<   14:31   0:00 [kworker/1:0H
root          26  0.0  0.0      0     0 ?        S    14:31   0:00 [kdevtmpfs]
root          27  0.0  0.0      0     0 ?        I<   14:31   0:00 [inet_frag_wq
```

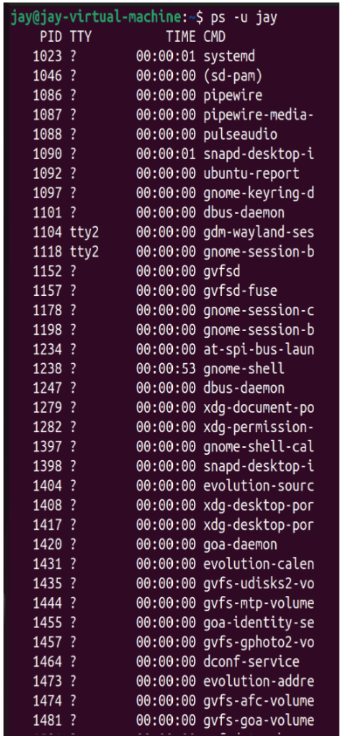
- 4.



```
jay@jay-virtual-machine:~$ ps -e --sort=-%cpu
    PID TTY          TIME CMD
   5986 ?        00:00:12 gnome-terminal-
   1238 ?        00:00:51 gnome-shell
    851 ?        00:00:24 snapd
   5943 ?        00:00:01 kworker/1:0-mpt_poll_0
   4485 ?        00:00:02 gjs
    625 ?        00:00:05 systemd-oomd
    686 ?        00:00:06 vmtoolsd
   1494 ?        00:00:08 packagekitd
   1637 ?        00:00:07 vmtoolsd
   1695 ?        00:00:09 snap-store
   5874 ?        00:00:00 kworker/1:1-events
      1 ?        00:00:03 systemd
      2 ?        00:00:00 kthreadd
      3 ?        00:00:00 rcu_gp
      4 ?        00:00:00 rcu_par_gp
      5 ?        00:00:00 slub_flushwq
      6 ?        00:00:00 netns
     11 ?        00:00:00 mm_percpu_wq
     12 ?        00:00:00 rcu_tasks_kthread
     13 ?        00:00:00 rcu_tasks_rude_kthread
     14 ?        00:00:00 rcu_tasks_trace_kthread
     15 ?        00:00:00 ksoftirqd/0
     16 ?        00:00:01 rcu_preempt
     17 ?        00:00:00 migration/0
     18 ?        00:00:00 idle_inject/0
     19 ?        00:00:00 cpuhp/0
     20 ?        00:00:00 cpuhp/1
     21 ?        00:00:00 idle_inject/1
     22 ?        00:00:00 migration/1
     23 ?        00:00:00 ksoftirqd/1
     25 ?        00:00:00 kworker/1:0H-ttm
     26 ?        00:00:00 kdevtmpfs
     27 ?        00:00:00 inet_frag_wq
     29 ?        00:00:00 kauditd
     31 ?        00:00:00 khungtaskd
```

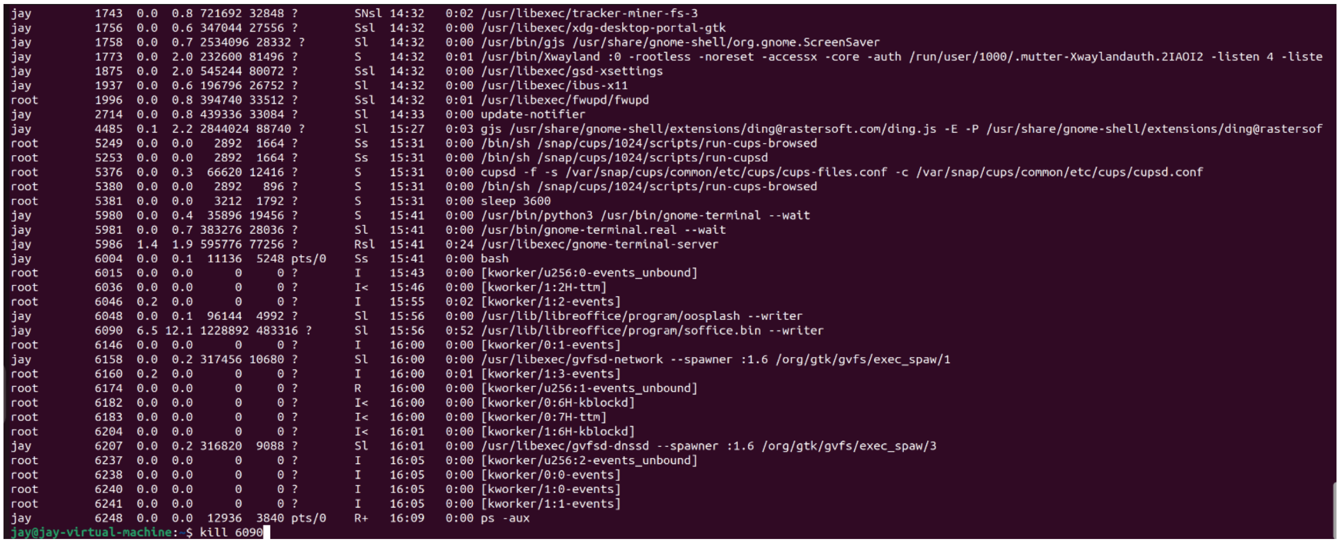


```
jay@jay-virtual-machine:~$ ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 14:31 ?        00:00:03 /sbin/init auto noprompt spl
root           2       0  0 14:31 ?        00:00:00 [kthreadd]
root           3       2  0 14:31 ?        00:00:00 [rcu_gp]
root           4       2  0 14:31 ?        00:00:00 [rcu_par_gp]
root           5       2  0 14:31 ?        00:00:00 [slub_flushwq]
root           6       2  0 14:31 ?        00:00:00 [netns]
root          11       2  0 14:31 ?        00:00:00 [mm_percpu_wq]
root          12       2  0 14:31 ?        00:00:00 [rcu_tasks_kthread]
root          13       2  0 14:31 ?        00:00:00 [rcu_tasks_rude_kthread]
root          14       2  0 14:31 ?        00:00:00 [rcu_tasks_trace_kthread]
root          15       2  0 14:31 ?        00:00:00 [ksoftirqd/0]
root          16       2  0 14:31 ?        00:00:01 [rcu_preempt]
root          17       2  0 14:31 ?        00:00:00 [migration/0]
root          18       2  0 14:31 ?        00:00:00 [idle_inject/0]
root          19       2  0 14:31 ?        00:00:00 [cpuhp/0]
root          20       2  0 14:31 ?        00:00:00 [cpuhp/1]
root          21       2  0 14:31 ?        00:00:00 [idle_inject/1]
root          22       2  0 14:31 ?        00:00:00 [migration/1]
root          23       2  0 14:31 ?        00:00:00 [ksoftirqd/1]
root          25       2  0 14:31 ?        00:00:00 [kworker/1:0H-kblockd]
root          26       2  0 14:31 ?        00:00:00 [kdevtmpfs]
root          27       2  0 14:31 ?        00:00:00 [inet_frag_wq]
root          29       2  0 14:31 ?        00:00:00 [kauditd]
root          31       2  0 14:31 ?        00:00:00 [khungtaskd]
root          32       2  0 14:31 ?        00:00:00 [oom_reaper]
root          34       2  0 14:31 ?        00:00:00 [writeback]
root          35       2  0 14:31 ?        00:00:00 [kcompactd0]
root          36       2  0 14:31 ?        00:00:00 [ksmd]
root          37       2  0 14:31 ?        00:00:00 [khugepaged]
root          38       2  0 14:31 ?        00:00:00 [kintegrityd]
root          39       2  0 14:31 ?        00:00:00 [kblockd]
root          40       2  0 14:31 ?        00:00:00 [blkcg_punt_bio]
root          41       2  0 14:31 ?        00:00:00 [tpm_dev_wq]
root          42       2  0 14:31 ?        00:00:00 [ata_sff]
```



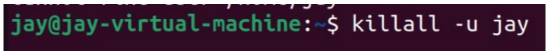
```
jay@jay-virtual-machine:~$ ps -u jay
    PID TTY          TIME CMD
   1023 ?        00:00:01 systemd
   1046 ?        00:00:00 (sd-pam)
   1086 ?        00:00:00 pipewire
   1087 ?        00:00:00 pipewire-media-
   1088 ?        00:00:00 pulseaudio
   1090 ?        00:00:01 snapd-desktop-i
   1092 ?        00:00:00 ubuntu-report
   1097 ?        00:00:00 gnome-keyring-d
   1101 ?        00:00:00 dbus-daemon
   1104 tty2     00:00:00 gdm-wayland-ses
   1118 tty2     00:00:00 gnome-session-b
   1152 ?        00:00:00 gvfsd
   1157 ?        00:00:00 gvfsd-fuse
   1178 ?        00:00:00 gnome-session-c
   1198 ?        00:00:00 gnome-session-b
   1234 ?        00:00:00 at-spi-bus-laun
   1238 ?        00:00:53 gnome-shell
   1247 ?        00:00:00 dbus-daemon
   1279 ?        00:00:00 xdg-document-po
   1282 ?        00:00:00 xdg-permission-
   1397 ?        00:00:00 gnome-shell-cal
   1398 ?        00:00:00 snapd-desktop-i
   1404 ?        00:00:00 evolution-sourc
   1408 ?        00:00:00 xdg-desktop-por
   1417 ?        00:00:00 xdg-desktop-por
   1420 ?        00:00:00 goa-daemon
   1431 ?        00:00:00 evolution-calen
   1435 ?        00:00:00 gvfs-udisks2-vo
   1444 ?        00:00:00 gvfs-mtp-volume
   1455 ?        00:00:00 goa-identity-se
   1457 ?        00:00:00 gvfs-gphoto2-vo
   1464 ?        00:00:00 dconf-service
   1473 ?        00:00:00 evolution-addre
   1474 ?        00:00:00 gvfs-afc-volume
   1481 ?        00:00:00 gvfs-goa-volume
```

5.



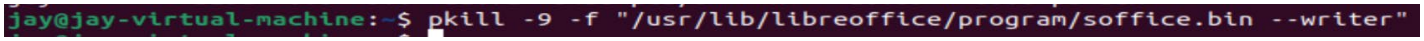
```
jay         1743  0.0  0.8 721692 32848 ?        SNsl 14:32   0:02 /usr/libexec/tracker-miner-fs-3
jay         1756  0.0  0.6 347044 27556 ?        Ssl  14:32   0:00 /usr/libexec/xdg-desktop-portal-gtk
jay         1758  0.0  0.7 2534096 28332 ?       Sl   14:32   0:00 /usr/bin/gjs /usr/share/gnome-shell/org.gnome.ScreenSaver
jay         1773  0.0  2.0 232600 81496 ?        S    14:32   0:01 /usr/bin/Xwayland :0 -rootless -noreset -accessx -core -auth /run/user/1000/.mutter-Xwaylandauth.2IAOI2 -listen 4 -liste
jay         1875  0.0  2.0 545244 80072 ?        Ssl  14:32   0:00 /usr/libexec/gsd-xsettings
jay         1937  0.0  0.6 196796 26752 ?        Sl   14:32   0:00 /usr/libexec/ibus-x11
root        1996  0.0  0.8 394740 33512 ?        Ssl  14:32   0:01 /usr/libexec/fwupd/fwupd
jay         2714  0.0  0.8 439336 33084 ?        Sl   14:33   0:00 update-notifier
jay         4485  0.1  2.2 2844024 88740 ?       Sl   15:27   0:03 gjs /usr/share/gnome-shell/extensions/ding@rastersoft.com/ding.js -E -P /usr/share/gnome-shell/extensions/ding@rastersof
root        5249  0.0  0.0   2892  1664 ?        Ss   15:31   0:00 /bin/sh /snap/cups/1024/scripts/run-cups-browsed
root        5253  0.0  0.0   2892  1664 ?        Ss   15:31   0:00 /bin/sh /snap/cups/1024/scripts/run-cupsd
root        5376  0.0  0.3  66620 12416 ?        S    15:31   0:00 cupsd -f -s /var/snap/cups/common/etc/cups/cups-files.conf -c /var/snap/cups/common/etc/cups/cupsd.conf
root        5380  0.0  0.0   2892   896 ?        S    15:31   0:00 /bin/sh /snap/cups/1024/scripts/run-cups-browsed
root        5381  0.0  0.0   3212  1792 ?        S    15:31   0:00 sleep 3600
jay         5980  0.0  0.4  35896 19456 ?        S    15:41   0:00 /usr/bin/python3 /usr/bin/gnome-terminal --wait
jay         5981  0.0  0.7 383276 28036 ?        Sl   15:41   0:00 /usr/bin/gnome-terminal.real --wait
jay         5986  1.4  1.9 595776 77256 ?        Rsl  15:41   0:24 /usr/libexec/gnome-terminal-server
jay         6004  0.0  0.1  11136  5248 pts/0    Ss   15:41   0:00 bash
root        6015  0.0  0.0      0     0 ?        I    15:43   0:00 [kworker/u256:0-events_unbound]
root        6036  0.0  0.0      0     0 ?        I<   15:46   0:00 [kworker/1:2H-ttm]
root        6046  0.2  0.0      0     0 ?        I    15:55   0:02 [kworker/1:2-events]
jay         6048  0.0  0.1  96144  4992 ?        Sl   15:56   0:00 /usr/lib/libreoffice/program/oosplash --writer
jay         6090  6.5 12.1 1228892 483316 ?      Sl   15:56   0:52 /usr/lib/libreoffice/program/soffice.bin --writer
root        6146  0.0  0.0      0     0 ?        I    16:00   0:00 [kworker/0:1-events]
jay         6158  0.0  0.2 317456 10680 ?        Sl   16:00   0:00 /usr/libexec/gvfsd-network --spawner :1.6 /org/gtk/gvfs/exec_spaw/1
root        6160  0.2  0.0      0     0 ?        I    16:00   0:01 [kworker/1:3-events]
root        6174  0.0  0.0      0     0 ?        R    16:00   0:00 [kworker/u256:1-events_unbound]
root        6182  0.0  0.0      0     0 ?        I<   16:00   0:00 [kworker/0:6H-kblockd]
root        6183  0.0  0.0      0     0 ?        I<   16:00   0:00 [kworker/0:7H-ttm]
root        6204  0.0  0.0      0     0 ?        I<   16:01   0:00 [kworker/1:6H-kblockd]
jay         6207  0.0  0.2 316820  9088 ?        Sl   16:01   0:00 /usr/libexec/gvfsd-dnssd --spawner :1.6 /org/gtk/gvfs/exec_spaw/3
root        6237  0.0  0.0      0     0 ?        I    16:05   0:00 [kworker/u256:2-events_unbound]
root        6238  0.0  0.0      0     0 ?        I    16:05   0:00 [kworker/0:0-events]
root        6240  0.0  0.0      0     0 ?        I    16:05   0:00 [kworker/1:0-events]
root        6241  0.0  0.0      0     0 ?        I    16:05   0:00 [kworker/1:1-events]
jay         6248  0.0  0.0  12936  3840 pts/0    R+   16:09   0:00 ps -aux
jay@jay-virtual-machine:~$ kill 6090
```

Figure 1: I used the kill command to close the LibreOffice Writer app I had opened. To use this, I needed to use the PID.



```
jay@jay-virtual-machine:~$ killall -u jay
```

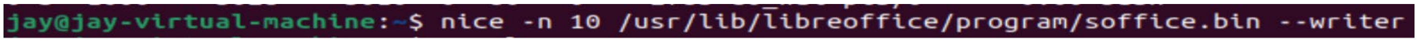
Figure 2: I used the killall command to terminate all processes that I was running. And so I had to restart ubuntu.



```
jay@jay-virtual-machine:~$ pkill -9 -f "/usr/lib/libreoffice/program/soffice.bin --writer"
```

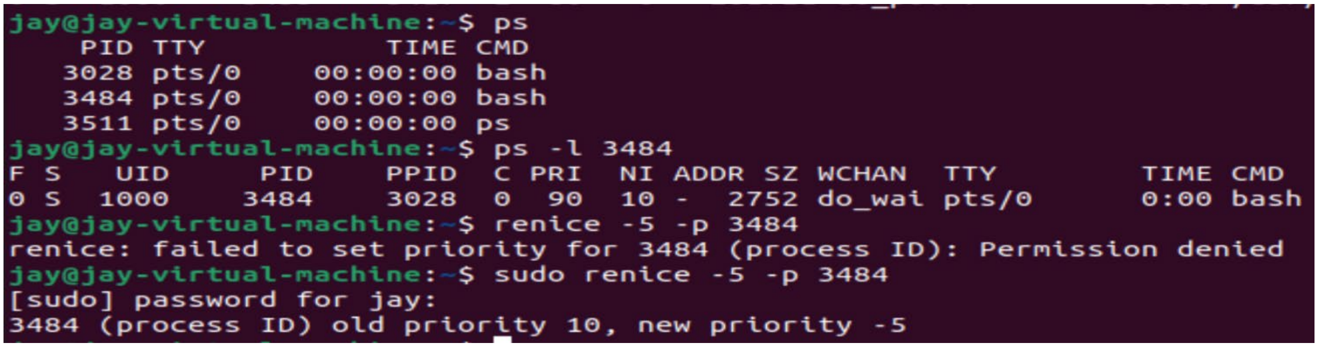
Figure 3: I used the pkill command to SIGKILL a specific command which is why I used -f.

6.



```
jay@jay-virtual-machine:~$ nice -n 10 /usr/lib/libreoffice/program/soffice.bin --writer
```

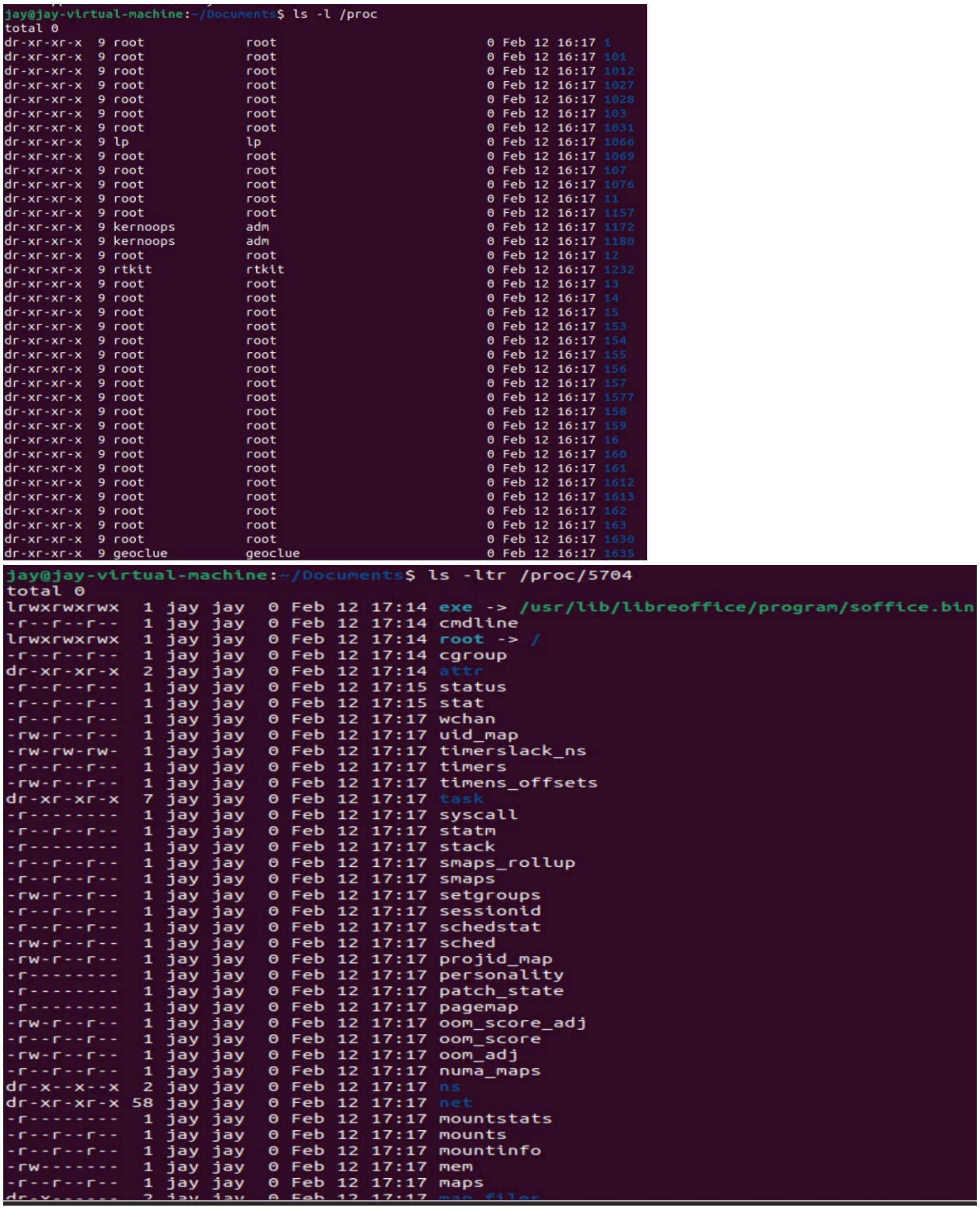
Figure 4: I used the nice command to assign a priority of 10 to the LibreOffice Writer command. Afterwards, it instantly opened a new document in the application.



```
jay@jay-virtual-machine:~$ ps
    PID TTY          TIME CMD
   3028 pts/0    00:00:00 bash
   3484 pts/0    00:00:00 bash
   3511 pts/0    00:00:00 ps
jay@jay-virtual-machine:~$ ps -l 3484
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY        TIME CMD
0 S  1000    3484    3028  0  90  10 -  2752 do_wai pts/0      0:00 bash
jay@jay-virtual-machine:~$ renice -5 -p 3484
renice: failed to set priority for 3484 (process ID): Permission denied
jay@jay-virtual-machine:~$ sudo renice -5 -p 3484
[sudo] password for jay:
3484 (process ID) old priority 10, new priority -5
```

Figure 5: I used the renice command to change the priority level of the command bash (PID: 3484).

7.

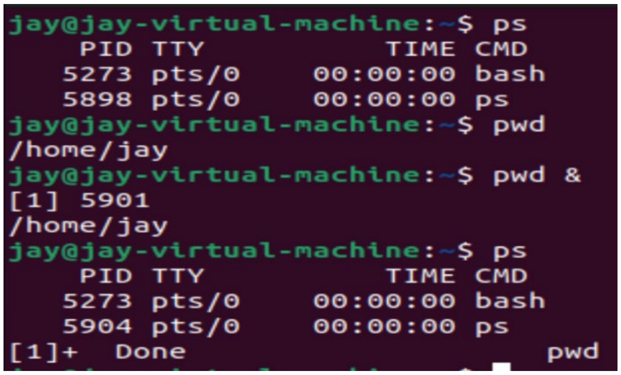


```
jay@jay-virtual-machine:~/Documents$ ls -l /proc
total 0
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 1
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 101
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 1012
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 1027
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 1028
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 103
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 1031
dr-xr-xr-x  9 lp               lp                            0 Feb 12 16:17 1066
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 1069
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 107
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 1076
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 11
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 1157
dr-xr-xr-x  9 kernoops         adm                           0 Feb 12 16:17 1172
dr-xr-xr-x  9 kernoops         adm                           0 Feb 12 16:17 1180
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 12
dr-xr-xr-x  9 rtkit            rtkit                         0 Feb 12 16:17 1232
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 13
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 14
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 15
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 153
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 154
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 155
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 156
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 157
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 1577
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 158
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 159
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 16
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 160
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 161
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 1612
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 1613
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 162
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 163
dr-xr-xr-x  9 root             root                          0 Feb 12 16:17 1630
dr-xr-xr-x  9 geoclue          geoclue                       0 Feb 12 16:17 1635
```

```
jay@jay-virtual-machine:~/Documents$ ls -ltr /proc/5704
total 0
lrwxrwxrwx  1 jay jay  0 Feb 12 17:14 exe -> /usr/lib/libreoffice/program/soffice.bin
-r--r--r--  1 jay jay  0 Feb 12 17:14 cmdline
lrwxrwxrwx  1 jay jay  0 Feb 12 17:14 root -> /
-r--r--r--  1 jay jay  0 Feb 12 17:14 cgroup
dr-xr-xr-x  2 jay jay  0 Feb 12 17:14 attr
-r--r--r--  1 jay jay  0 Feb 12 17:15 status
-r--r--r--  1 jay jay  0 Feb 12 17:15 stat
-r--r--r--  1 jay jay  0 Feb 12 17:17 wchan
-rw-r--r--  1 jay jay  0 Feb 12 17:17 uid_map
-rw-rw-rw-  1 jay jay  0 Feb 12 17:17 timerslack_ns
-r--r--r--  1 jay jay  0 Feb 12 17:17 timers
-rw-r--r--  1 jay jay  0 Feb 12 17:17 timens_offsets
dr-xr-xr-x  7 jay jay  0 Feb 12 17:17 task
-r--------  1 jay jay  0 Feb 12 17:17 syscall
-r--r--r--  1 jay jay  0 Feb 12 17:17 statm
-r--------  1 jay jay  0 Feb 12 17:17 stack
-r--r--r--  1 jay jay  0 Feb 12 17:17 smaps_rollup
-r--r--r--  1 jay jay  0 Feb 12 17:17 smaps
-rw-r--r--  1 jay jay  0 Feb 12 17:17 setgroups
-r--r--r--  1 jay jay  0 Feb 12 17:17 sessionid
-r--r--r--  1 jay jay  0 Feb 12 17:17 schedstat
-rw-r--r--  1 jay jay  0 Feb 12 17:17 sched
-rw-r--r--  1 jay jay  0 Feb 12 17:17 projid_map
-r--------  1 jay jay  0 Feb 12 17:17 personality
-r--------  1 jay jay  0 Feb 12 17:17 patch_state
-r--------  1 jay jay  0 Feb 12 17:17 pagemap
-rw-r--r--  1 jay jay  0 Feb 12 17:17 oom_score_adj
-r--r--r--  1 jay jay  0 Feb 12 17:17 oom_score
-rw-r--r--  1 jay jay  0 Feb 12 17:17 oom_adj
-r--r--r--  1 jay jay  0 Feb 12 17:17 numa_maps
dr-x--x--x  2 jay jay  0 Feb 12 17:17 ns
dr-xr-xr-x 58 jay jay  0 Feb 12 17:17 net
-r--------  1 jay jay  0 Feb 12 17:17 mountstats
-r--r--r--  1 jay jay  0 Feb 12 17:17 mounts
-r--r--r--  1 jay jay  0 Feb 12 17:17 mountinfo
-rw-------  1 jay jay  0 Feb 12 17:17 mem
-r--r--r--  1 jay jay  0 Feb 12 17:17 maps
dr-x------  2 jay jay  0 Feb 12 17:17 map_files
```

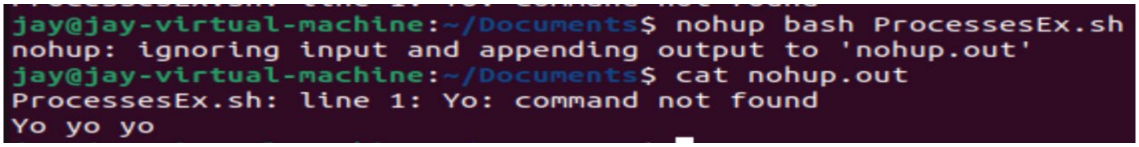
Figure 6: I can use the /proc command to list information about all files and directories. I used the "ls -ltr /proc/5704" command to find information for a specific process.

8.



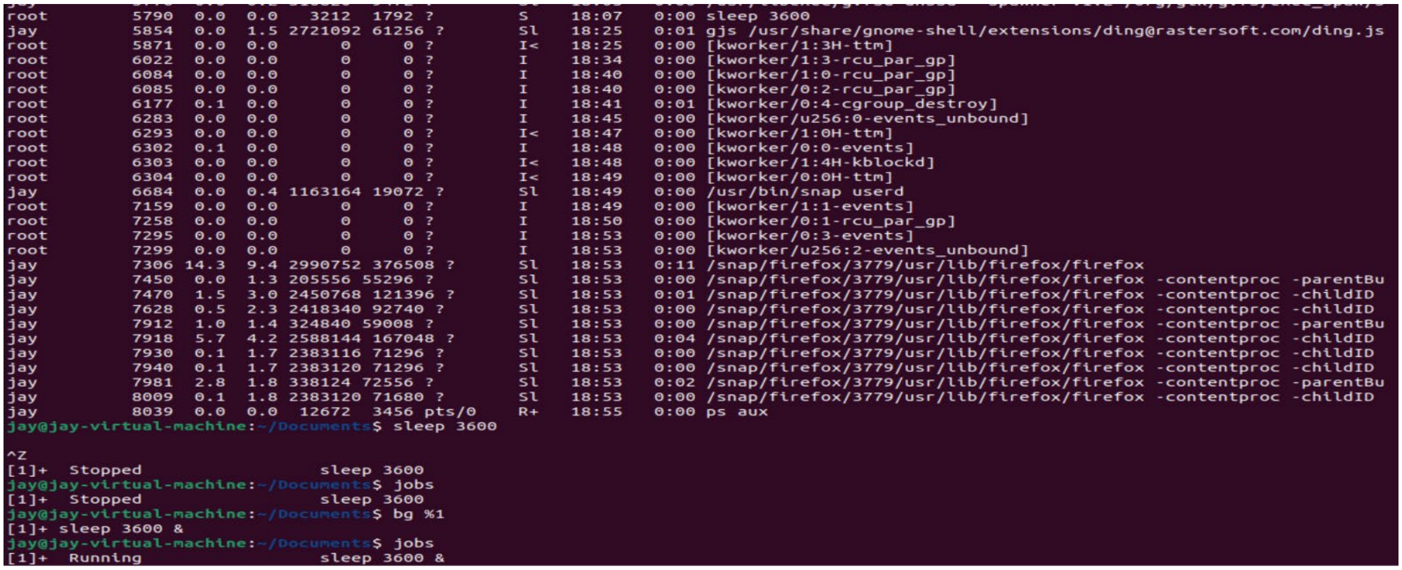
```
jay@jay-virtual-machine:~$ ps
    PID TTY          TIME CMD
   5273 pts/0    00:00:00 bash
   5898 pts/0    00:00:00 ps
jay@jay-virtual-machine:~$ pwd
/home/jay
jay@jay-virtual-machine:~$ pwd &
[1] 5901
/home/jay
jay@jay-virtual-machine:~$ ps
    PID TTY          TIME CMD
   5273 pts/0    00:00:00 bash
   5904 pts/0    00:00:00 ps
[1]+  Done                    pwd
```

Figure 7: Using the & command to run a background process.



```
jay@jay-virtual-machine:~/Documents$ nohup bash ProcessesEx.sh
nohup: ignoring input and appending output to 'nohup.out'
jay@jay-virtual-machine:~/Documents$ cat nohup.out
ProcessesEx.sh: line 1: Yo: command not found
Yo yo yo
```

Figure 8: Using the nohup command to run a process.

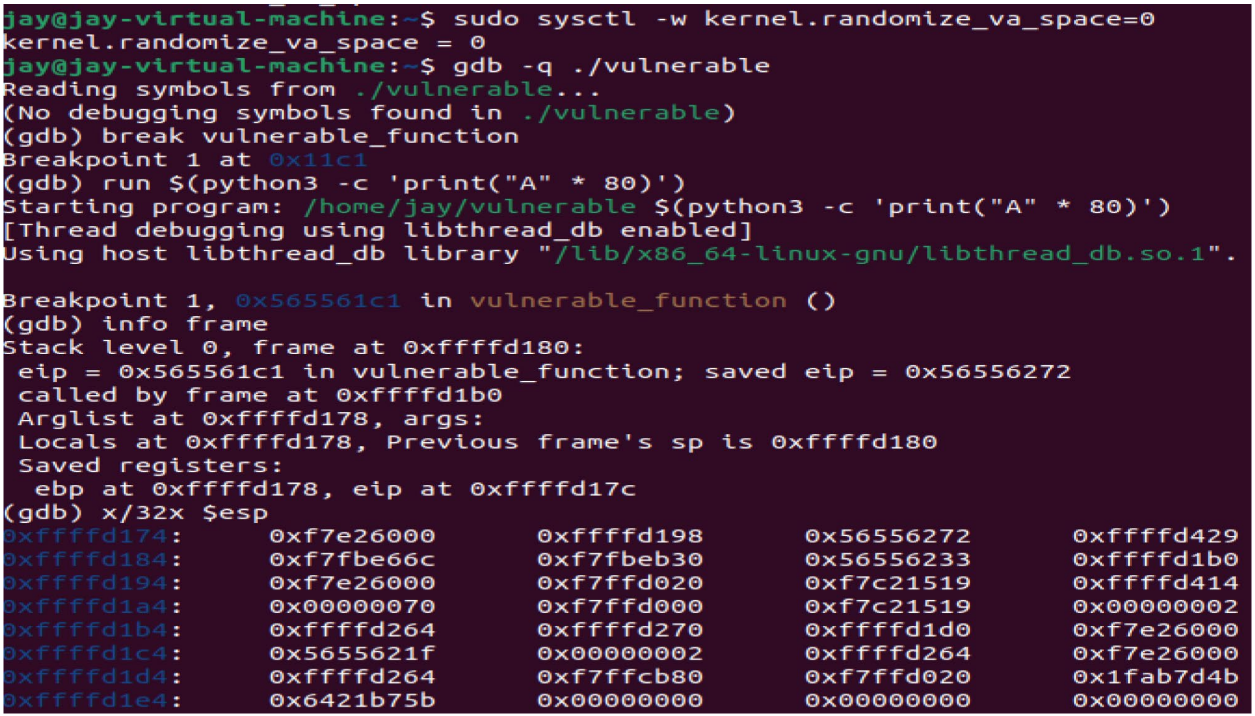


```
root        5790  0.0  0.0   3212  1792 ?        S    18:07   0:00 sleep 3600
jay         5854  0.0  1.5 2721092 61256 ?       Sl   18:25   0:01 gjs /usr/share/gnome-shell/extensions/ding@rastersoft.com/ding.js
root        5871  0.0  0.0      0     0 ?        I<   18:25   0:00 [kworker/1:3H-ttm]
root        6022  0.0  0.0      0     0 ?        I    18:34   0:00 [kworker/1:3-rcu_par_gp]
root        6084  0.0  0.0      0     0 ?        I    18:40   0:00 [kworker/1:0-rcu_par_gp]
root        6085  0.0  0.0      0     0 ?        I    18:40   0:00 [kworker/0:2-rcu_par_gp]
root        6177  0.1  0.0      0     0 ?        I    18:41   0:01 [kworker/0:4-cgroup_destroy]
root        6283  0.0  0.0      0     0 ?        I    18:45   0:00 [kworker/u256:0-events_unbound]
root        6293  0.0  0.0      0     0 ?        I<   18:47   0:00 [kworker/1:0H-ttm]
root        6302  0.1  0.0      0     0 ?        I    18:48   0:00 [kworker/0:0-events]
root        6303  0.0  0.0      0     0 ?        I<   18:48   0:00 [kworker/1:4H-kblockd]
root        6304  0.0  0.0      0     0 ?        I<   18:49   0:00 [kworker/0:0H-ttm]
jay         6684  0.0  0.4 1163164 19072 ?       Sl   18:49   0:00 /usr/bin/snap userd
root        7159  0.0  0.0      0     0 ?        I    18:49   0:00 [kworker/1:1-events]
root        7258  0.0  0.0      0     0 ?        I    18:50   0:00 [kworker/0:1-rcu_par_gp]
root        7295  0.0  0.0      0     0 ?        I    18:53   0:00 [kworker/0:3-events]
root        7299  0.0  0.0      0     0 ?        I    18:53   0:00 [kworker/u256:2-events_unbound]
jay         7306 14.3  9.4 2990752 376508 ?      Sl   18:53   0:11 /snap/firefox/3779/usr/lib/firefox/firefox
jay         7450  0.0  1.3 205556 55296 ?        Sl   18:53   0:00 /snap/firefox/3779/usr/lib/firefox/firefox -contentproc -parentBu
jay         7470  1.5  3.0 2450768 121396 ?      Sl   18:53   0:01 /snap/firefox/3779/usr/lib/firefox/firefox -contentproc -childID
jay         7628  0.5  2.3 2418340 92740 ?       Sl   18:53   0:00 /snap/firefox/3779/usr/lib/firefox/firefox -contentproc -childID
jay         7912  1.0  1.4 324840 59008 ?        Sl   18:53   0:00 /snap/firefox/3779/usr/lib/firefox/firefox -contentproc -parentBu
jay         7918  5.7  4.2 2588144 167048 ?      Sl   18:53   0:04 /snap/firefox/3779/usr/lib/firefox/firefox -contentproc -childID
jay         7930  0.1  1.7 2383116 71296 ?       Sl   18:53   0:00 /snap/firefox/3779/usr/lib/firefox/firefox -contentproc -childID
jay         7940  0.1  1.7 2383120 71296 ?       Sl   18:53   0:00 /snap/firefox/3779/usr/lib/firefox/firefox -contentproc -childID
jay         7981  2.8  1.8 338124 72556 ?        Sl   18:53   0:02 /snap/firefox/3779/usr/lib/firefox/firefox -contentproc -parentBu
jay         8009  0.1  1.8 2383120 71680 ?       Sl   18:53   0:00 /snap/firefox/3779/usr/lib/firefox/firefox -contentproc -childID
jay         8039  0.0  0.0  12672  3456 pts/0    R+   18:55   0:00 ps aux
jay@jay-virtual-machine:~/Documents$ sleep 3600

^Z
[1]+  Stopped                 sleep 3600
jay@jay-virtual-machine:~/Documents$ jobs
[1]+  Stopped                 sleep 3600
jay@jay-virtual-machine:~/Documents$ bg %1
[1]+ sleep 3600 &
jay@jay-virtual-machine:~/Documents$ jobs
[1]+  Running                 sleep 3600 &
```

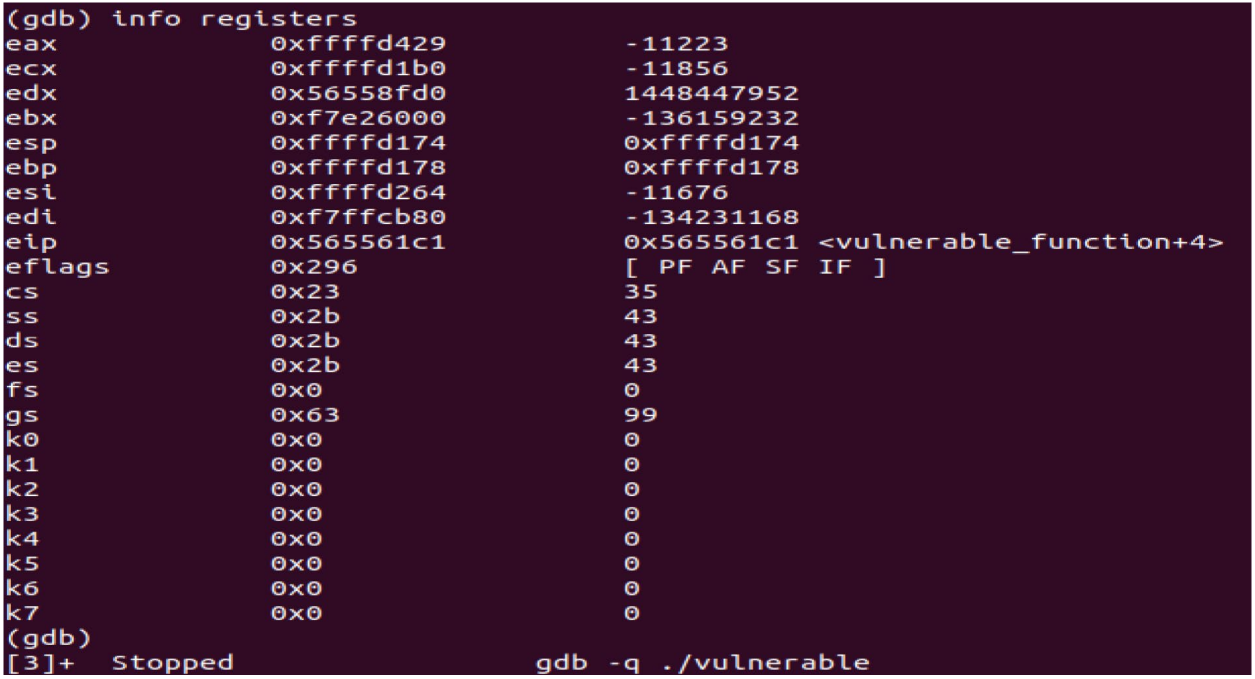
Figure 9: Using the bg command to continue running a process in the background.

Play with Processes on your Ubuntu VM: Lab 01 Part 2

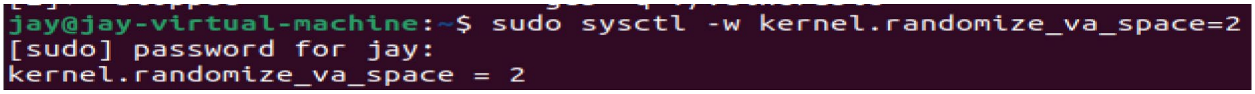


```
jay@jay-virtual-machine:~$ sudo sysctl -w kernel.randomize_va_space=0
kernel.randomize_va_space = 0
jay@jay-virtual-machine:~$ gdb -q ./vulnerable
Reading symbols from ./vulnerable...
(No debugging symbols found in ./vulnerable)
(gdb) break vulnerable_function
Breakpoint 1 at 0x11c1
(gdb) run $(python3 -c 'print("A" * 80)')
Starting program: /home/jay/vulnerable $(python3 -c 'print("A" * 80)')
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, 0x565561c1 in vulnerable_function ()
(gdb) info frame
Stack level 0, frame at 0xffffd180:
 eip = 0x565561c1 in vulnerable_function; saved eip = 0x56556272
 called by frame at 0xffffd1b0
 Arglist at 0xffffd178, args:
 Locals at 0xffffd178, Previous frame's sp is 0xffffd180
 Saved registers:
  ebp at 0xffffd178, eip at 0xffffd17c
(gdb) x/32x $esp
0xffffd174:     0xf7e26000      0xffffd198      0x56556272      0xffffd429
0xffffd184:     0xf7fbe66c      0xf7fbeb30      0x56556233      0xffffd1b0
0xffffd194:     0xf7e26000      0xf7ffd020      0xf7c21519      0xffffd414
0xffffd1a4:     0x00000070      0xf7ffd000      0xf7c21519      0x00000002
0xffffd1b4:     0xffffd264      0xffffd270      0xffffd1d0      0xf7e26000
0xffffd1c4:     0x5655621f      0x00000002      0xffffd264      0xf7e26000
0xffffd1d4:     0xffffd264      0xf7ffcb80      0xf7ffd020      0x1fab7d4b
0xffffd1e4:     0x6421b75b      0x00000000      0x00000000      0x00000000
```



```
(gdb) info registers
eax            0xffffd429          -11223
ecx            0xffffd1b0          -11856
edx            0x56558fd0          1448447952
ebx            0xf7e26000          -136159232
esp            0xffffd174          0xffffd174
ebp            0xffffd178          0xffffd178
esi            0xffffd264          -11676
edi            0xf7ffcb80          -134231168
eip            0x565561c1          0x565561c1 <vulnerable_function+4>
eflags         0x296               [ PF AF SF IF ]
cs             0x23                35
ss             0x2b                43
ds             0x2b                43
es             0x2b                43
fs             0x0                 0
gs             0x63                99
k0             0x0                 0
k1             0x0                 0
k2             0x0                 0
k3             0x0                 0
k4             0x0                 0
k5             0x0                 0
k6             0x0                 0
k7             0x0                 0
(gdb)
[3]+  Stopped                 gdb -q ./vulnerable
```



```
jay@jay-virtual-machine:~$ sudo sysctl -w kernel.randomize_va_space=2
[sudo] password for jay:
kernel.randomize_va_space = 2
```

1. In a multitasking operating system, why is it essential to have a mechanism like ps to list running processes?

- It is essential to have a mechanism like the ps command because it informs the user what processes are running so when there is a problem, it is easier to troubleshoot.

2. Can you think of real-world scenarios where identifying running processes would be crucial for system management or security?

- A real-world scenario where identifying running processes is crucial is when it comes to system performance. For instance, if someone typically runs multiple application or tabs, a virtual machine, and background processes. Their systems might not perform as well as if the user ran less processes, so troubleshooting using the ps command might be beneficial so the user can stop running certain processes.

3. Why is it valuable for system administrators to know the PID of running processes?

- It is valuable for system administrators to know the PID of running processes so they can troubleshoot specific running processes.

4. In terms of system performance optimization, how can understanding CPU and memory usage help in resource management?

- Understanding CPU and memory usage is important for resource management because both components impact a computer's performance. Understanding how each component works and being able to manage them, can improve a systems performance.

5. Explain how processes transition between states and the role of the scheduler.

- Processes transition between states based on their execution status (Start, Ready, Running, Waiting, and Terminate) and resource availability. When open Google Chrome, the program is loaded from your computer's hard drive into RAM (Start). Once the program is in RAM, it is waiting for its turn from the scheduler (Ready). Once the processer gets its turn, it can actively process the program instructions so opening and displaying web page (Run). Sometimes the program has to wait due to network traffic or user input so when it takes time to open up a web page (Wait). When you are finished with your tasks, you close the web browser and so the processor stops working on that program (Terminate). The process scheduler role is important because it decides what task /process to run next on the CPU, thus which process gets CPU time.

6. Explain the importance of CPU time sharing among processes.

- The CPU runs one process at a time, but due to time sharing it creates an illusion of running multiple processes at once (also known as abstraction). This is able to occur due to there being multiple cores in a processor which increase processing power.

7. Discuss the importance of understanding processes and scheduling for

cybersecurity?

- It is important to understand processes and scheduling for cybersecurity because as technology changes, you need to test new processes and/or change the priority of existing processes to ensure an organization’s cyber readiness.

8. What is a buffer overflow and, concisely, how does it happen?

- A buffer overflow is a software coding vulnerability. Ioccurs when the amount of data in the buffer exceeds its storage capacity. And the extra data overflows into adjacent memory locations, corrupting/overwriting the data in those locations.

9. What is the –m32 in this command gcc -o vulnerable.c -m32 vulnerable?

- It is there to compile 32 bits since by default the compiler is configured to compile 64 bits.

10. What does this python command ./vulnerable $(python -c 'print("A" * 80)') do?

- It runs the vulnerable program with a long input to trigger the buffer overflow.

11. What is ASLR and how does it prevent a buffer overflow?

- Address Space Layout Randomization (ASLR) is a security technique used in operating systems. It prevents buffer overflow by randomizing different parts of a program in memory. So every time a program is ran, components such as the stack, heap, and libraries are moved to different addresses in virtual memory.

12. What can you use GDB for in Linux?

- You can use GDB to help debug written programs.

13. Why is GDB used when a program crashes or gets a segmentation fault?

- GDB is used when a program crashes or gets a segmentation fault because it allows you to start and stop your program at any point and view the current values of program variables. Therefore, it helps find errors in programs that crash or get a segmentation fault.