Processes Basics Command Exercise

- **Question 1** : on your Linux host, there are many processes running at a time. However, one information can uniquely identify a process.

How is it called?

Expand Me

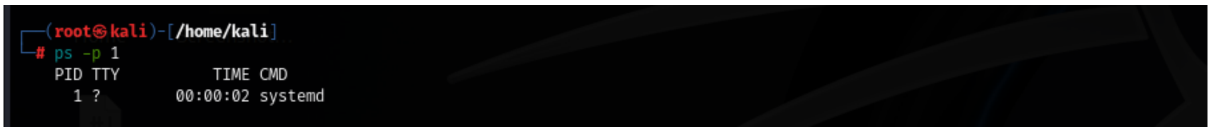
On Linux, a process can be uniquely identified by a PID (or process ID), which can't be assigned to two distinct processes at a time.

- **Question 2** : when your system boots, it starts the very first process on your instance.

How is it called?

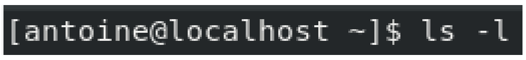
Expand Me

It is called the init process and it is used in order to execute initialization scripts for network, jobs or modules. On recent distributions, it has been replaced by a systemd process.



```
┌──(root㉿kali)-[/home/kali]
└─# ps -p 1
    PID TTY          TIME CMD
      1 ?        00:00:02 systemd
```

- **Question 3** : you currently have a shell terminal open on your host and you execute the following command.



```
[antoine@localhost ~]$ ls -l
```

Internally, what are the system calls invoked to perform such a command?

Expand Me

First, the kernel will fork the current process (i.e the bash interpreter) into a new process. Next, the image of bash process will be replaced by the loaded image of the ls program. Finally, the command is executed.

- **Question 4** : you open a shell terminal on your host by clicking on "Terminal".

In short, describe how the terminal works.

Expand Me

The terminal is a simple interactive process that waits perpetually for user input. When a command is issued, the command is executed by forking into a new process and executing the command in it. In the meantime, the parent process (i.e the terminal itself) waits for termination of the child process. When it has finished, the parent process resumes.

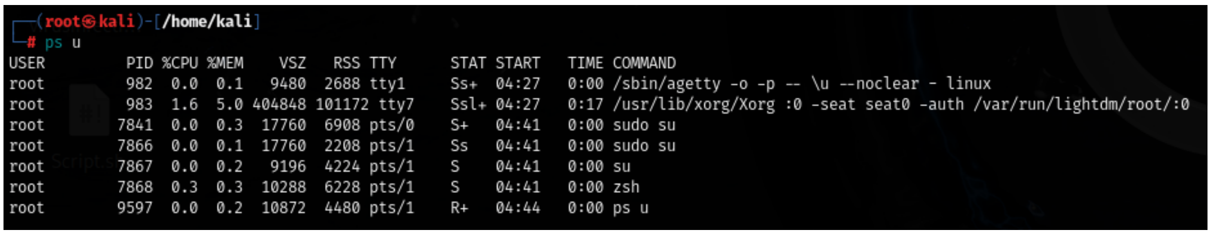
Processes Commands

- **Question 5** : you are asked by your system administrator to identify all processes that you own on the host.

Which command would you run to do that?

Expand Me

The easiest way to do that is to execute the ps command. By default, it won't report tty devices, but you can choose to execute "ps u" to see all processes.



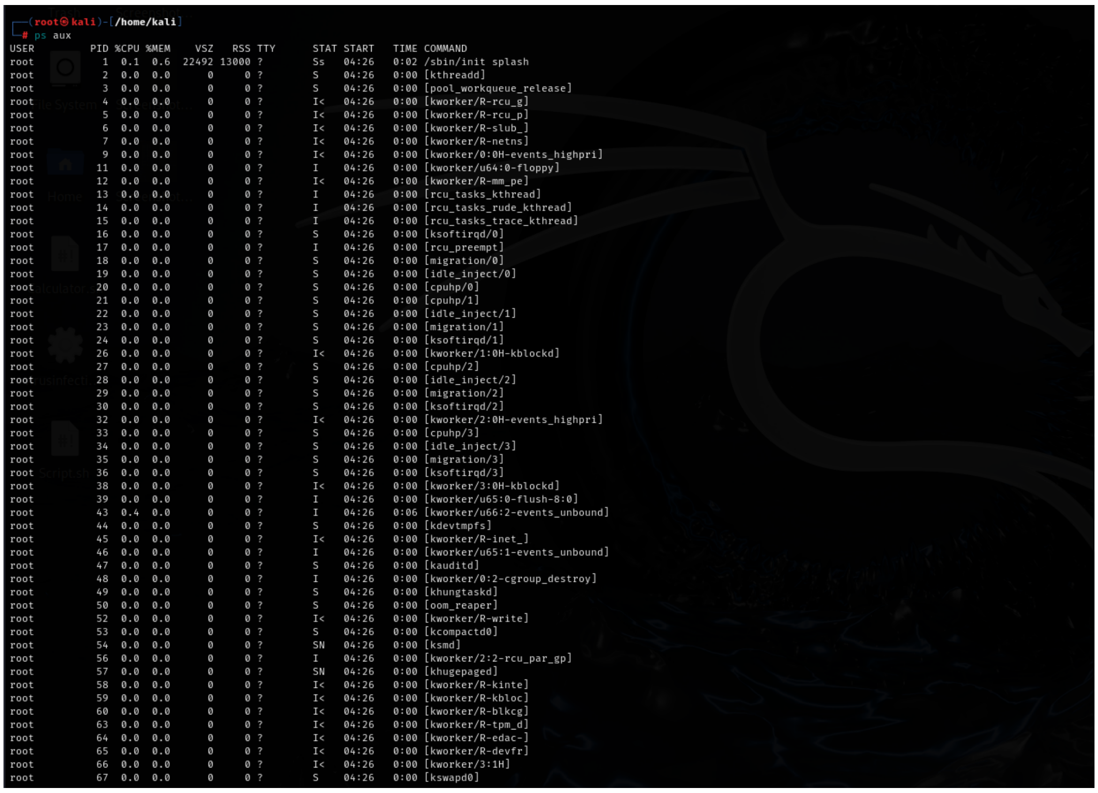
```
┌──(root㉿kali)-[/home/kali]
└─# ps u
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         982  0.0  0.1   9480  2688 tty1     Ss+  04:27   0:00 /sbin/agetty -o -p -- \u --noclear - linux
root         983  1.6  5.0 404848 101172 tty7    Ssl+ 04:27   0:17 /usr/lib/xorg/Xorg :0 -seat seat0 -auth /var/run/lightdm/root/:0
root        7841  0.0  0.3  17760  6908 pts/0    S+   04:41   0:00 sudo su
root        7866  0.0  0.1  17760  2208 pts/1    Ss   04:41   0:00 sudo su
root        7867  0.0  0.2   9196  4224 pts/1    S    04:41   0:00 su
root        7868  0.3  0.3  10288  6228 pts/1    S    04:41   0:00 zsh
root        9597  0.0  0.2  10872  4480 pts/1    R+   04:44   0:00 ps u
```

- **Question 6** : you are asked by your system administrator to identify all the processes on your system.

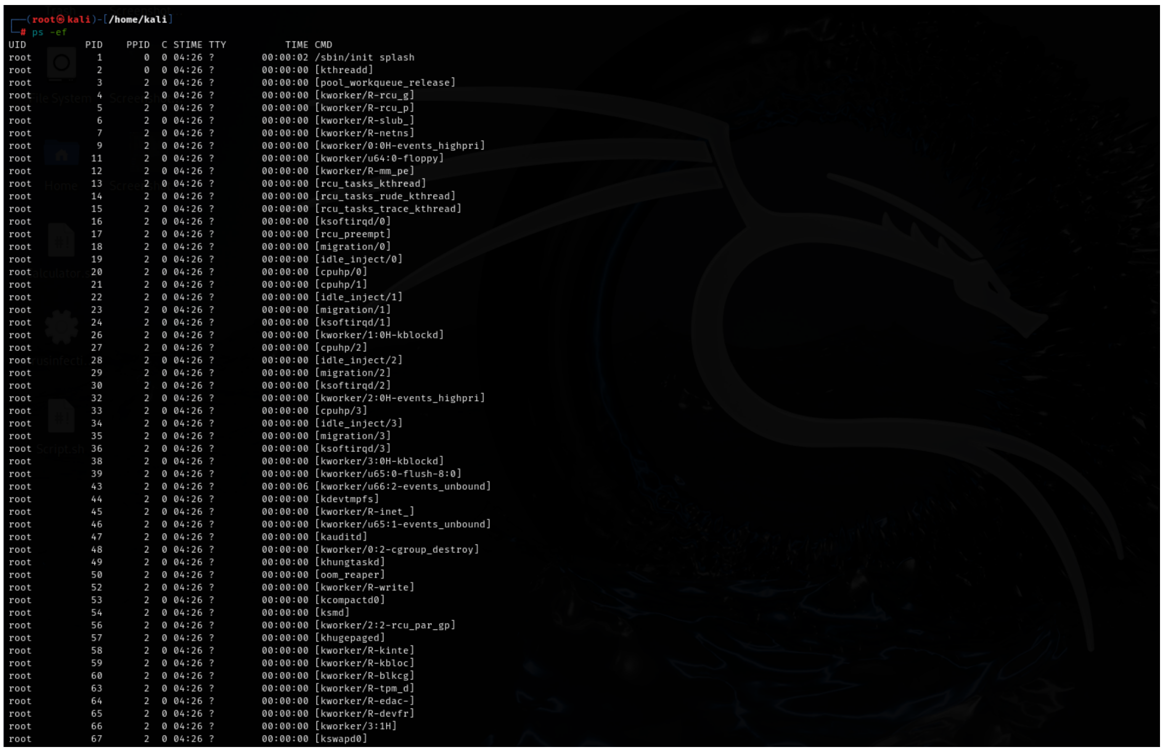
Can you provide two commands that display all processes on the host?

Expand Me

To display all processes on Linux, you can either use "ps aux" (which is a BSD syntax) or "ps -ef" (which is a POSIX syntax)



```
┌──(root㉿kali)-[/home/kali]
└─# ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.1  0.6  22492 13000 ?        Ss   04:26   0:02 /sbin/init splash
root           2  0.0  0.0      0     0 ?        S    04:26   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        S    04:26   0:00 [pool_workqueue_release]
root           4  0.0  0.0      0     0 ?        I<   04:26   0:00 [kworker/R-rcu_g]
root           5  0.0  0.0      0     0 ?        I<   04:26   0:00 [kworker/R-rcu_p]
root           6  0.0  0.0      0     0 ?        I<   04:26   0:00 [kworker/R-slub_]
root           7  0.0  0.0      0     0 ?        I<   04:26   0:00 [kworker/R-netns]
root           9  0.0  0.0      0     0 ?        I<   04:26   0:00 [kworker/0:0H-events_highpri]
root          11  0.0  0.0      0     0 ?        I    04:26   0:00 [kworker/u64:0-floppy]
root          12  0.0  0.0      0     0 ?        I<   04:26   0:00 [kworker/R-mm_pe]
root          13  0.0  0.0      0     0 ?        I    04:26   0:00 [rcu_tasks_kthread]
root          14  0.0  0.0      0     0 ?        I    04:26   0:00 [rcu_tasks_rude_kthread]
root          15  0.0  0.0      0     0 ?        I    04:26   0:00 [rcu_tasks_trace_kthread]
root          16  0.0  0.0      0     0 ?        S    04:26   0:00 [ksoftirqd/0]
root          17  0.0  0.0      0     0 ?        I    04:26   0:00 [rcu_preempt]
root          18  0.0  0.0      0     0 ?        S    04:26   0:00 [migration/0]
root          19  0.0  0.0      0     0 ?        S    04:26   0:00 [idle_inject/0]
root          20  0.0  0.0      0     0 ?        S    04:26   0:00 [cpuhp/0]
root          21  0.0  0.0      0     0 ?        S    04:26   0:00 [cpuhp/1]
root          22  0.0  0.0      0     0 ?        S    04:26   0:00 [idle_inject/1]
root          23  0.0  0.0      0     0 ?        S    04:26   0:00 [migration/1]
root          24  0.0  0.0      0     0 ?        S    04:26   0:00 [ksoftirqd/1]
root          26  0.0  0.0      0     0 ?        I<   04:26   0:00 [kworker/1:0H-kblockd]
root          27  0.0  0.0      0     0 ?        S    04:26   0:00 [cpuhp/2]
root          28  0.0  0.0      0     0 ?        S    04:26   0:00 [idle_inject/2]
root          29  0.0  0.0      0     0 ?        S    04:26   0:00 [migration/2]
root          30  0.0  0.0      0     0 ?        S    04:26   0:00 [ksoftirqd/2]
root          32  0.0  0.0      0     0 ?        I<   04:26   0:00 [kworker/2:0H-events_highpri]
root          33  0.0  0.0      0     0 ?        S    04:26   0:00 [cpuhp/3]
root          34  0.0  0.0      0     0 ?        S    04:26   0:00 [idle_inject/3]
root          35  0.0  0.0      0     0 ?        S    04:26   0:00 [migration/3]
root          36  0.0  0.0      0     0 ?        S    04:26   0:00 [ksoftirqd/3]
root          38  0.0  0.0      0     0 ?        I<   04:26   0:00 [kworker/3:0H-kblockd]
root          39  0.0  0.0      0     0 ?        I    04:26   0:00 [kworker/u65:0-flush-8:0]
root          43  0.4  0.0      0     0 ?        I    04:26   0:06 [kworker/u66:2-events_unbound]
root          44  0.0  0.0      0     0 ?        S    04:26   0:00 [kdevtmpfs]
root          45  0.0  0.0      0     0 ?        I<   04:26   0:00 [kworker/R-inet_]
root          46  0.0  0.0      0     0 ?        I    04:26   0:00 [kworker/u65:1-events_unbound]
root          47  0.0  0.0      0     0 ?        S    04:26   0:00 [kauditd]
root          48  0.0  0.0      0     0 ?        I    04:26   0:00 [kworker/0:2-cgroup_destroy]
root          49  0.0  0.0      0     0 ?        S    04:26   0:00 [khungtaskd]
root          50  0.0  0.0      0     0 ?        S    04:26   0:00 [oom_reaper]
root          52  0.0  0.0      0     0 ?        I<   04:26   0:00 [kworker/R-write]
root          53  0.0  0.0      0     0 ?        S    04:26   0:00 [kcompactd0]
root          54  0.0  0.0      0     0 ?        SN   04:26   0:00 [ksmd]
root          56  0.0  0.0      0     0 ?        I    04:26   0:00 [kworker/2:2-rcu_par_gp]
root          57  0.0  0.0      0     0 ?        SN   04:26   0:00 [khugepaged]
root          58  0.0  0.0      0     0 ?        I<   04:26   0:00 [kworker/R-kinte]
root          59  0.0  0.0      0     0 ?        I<   04:26   0:00 [kworker/R-kbloc]
root          60  0.0  0.0      0     0 ?        I<   04:26   0:00 [kworker/R-blkcg]
root          63  0.0  0.0      0     0 ?        I<   04:26   0:00 [kworker/R-tpm_d]
root          64  0.0  0.0      0     0 ?        I<   04:26   0:00 [kworker/R-edac-]
root          65  0.0  0.0      0     0 ?        I<   04:26   0:00 [kworker/R-devfr]
root          66  0.0  0.0      0     0 ?        I<   04:26   0:00 [kworker/3:1H]
root          67  0.0  0.0      0     0 ?        S    04:26   0:00 [kswapd0]
```



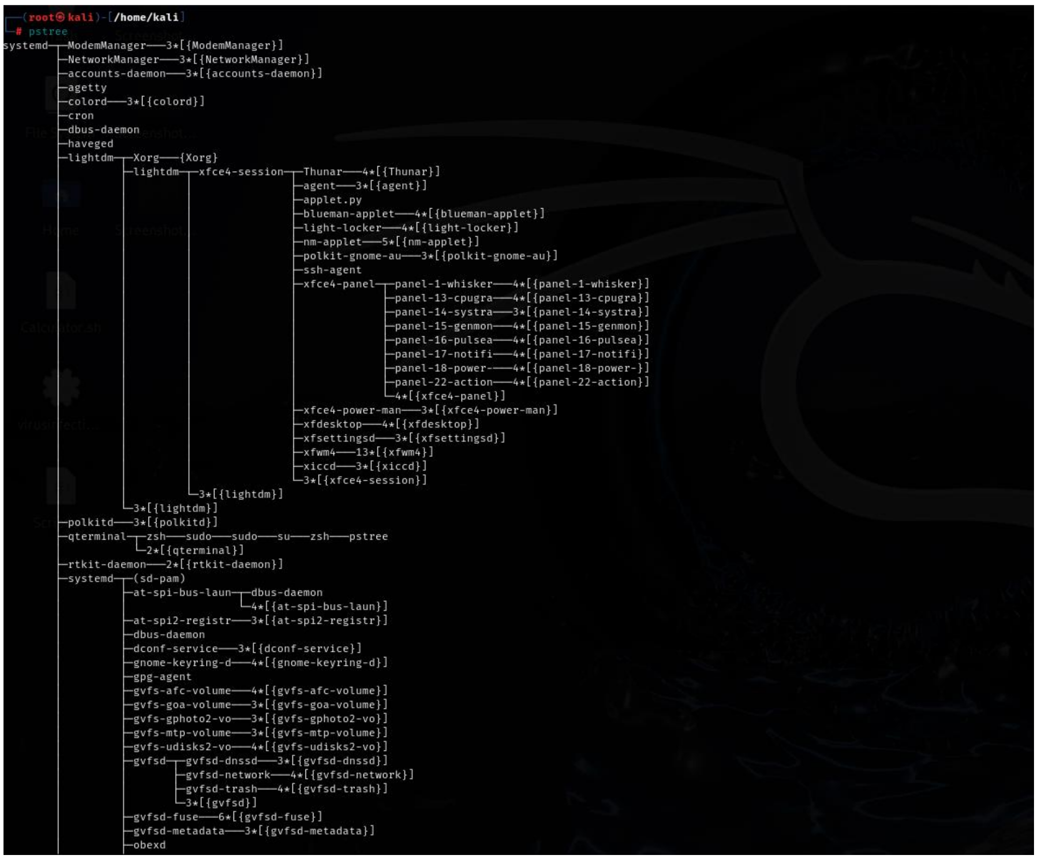
```
┌──(root㉿kali)-[/home/kali]
└─# ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 04:26 ?        00:00:02 /sbin/init splash
root           2       0  0 04:26 ?        00:00:00 [kthreadd]
root           3       2  0 04:26 ?        00:00:00 [pool_workqueue_release]
root           4       2  0 04:26 ?        00:00:00 [kworker/R-rcu_g]
root           5       2  0 04:26 ?        00:00:00 [kworker/R-rcu_p]
root           6       2  0 04:26 ?        00:00:00 [kworker/R-slub_]
root           7       2  0 04:26 ?        00:00:00 [kworker/R-netns]
root           9       2  0 04:26 ?        00:00:00 [kworker/0:0H-events_highpri]
root          11       2  0 04:26 ?        00:00:00 [kworker/u64:0-floppy]
root          12       2  0 04:26 ?        00:00:00 [kworker/R-mm_pe]
root          13       2  0 04:26 ?        00:00:00 [rcu_tasks_kthread]
root          14       2  0 04:26 ?        00:00:00 [rcu_tasks_rude_kthread]
root          15       2  0 04:26 ?        00:00:00 [rcu_tasks_trace_kthread]
root          16       2  0 04:26 ?        00:00:00 [ksoftirqd/0]
root          17       2  0 04:26 ?        00:00:00 [rcu_preempt]
root          18       2  0 04:26 ?        00:00:00 [migration/0]
root          19       2  0 04:26 ?        00:00:00 [idle_inject/0]
root          20       2  0 04:26 ?        00:00:00 [cpuhp/0]
root          21       2  0 04:26 ?        00:00:00 [cpuhp/1]
root          22       2  0 04:26 ?        00:00:00 [idle_inject/1]
root          23       2  0 04:26 ?        00:00:00 [migration/1]
root          24       2  0 04:26 ?        00:00:00 [ksoftirqd/1]
root          26       2  0 04:26 ?        00:00:00 [kworker/1:0H-kblockd]
root          27       2  0 04:26 ?        00:00:00 [cpuhp/2]
root          28       2  0 04:26 ?        00:00:00 [idle_inject/2]
root          29       2  0 04:26 ?        00:00:00 [migration/2]
root          30       2  0 04:26 ?        00:00:00 [ksoftirqd/2]
root          32       2  0 04:26 ?        00:00:00 [kworker/2:0H-events_highpri]
root          33       2  0 04:26 ?        00:00:00 [cpuhp/3]
root          34       2  0 04:26 ?        00:00:00 [idle_inject/3]
root          35       2  0 04:26 ?        00:00:00 [migration/3]
root          36       2  0 04:26 ?        00:00:00 [ksoftirqd/3]
root          38       2  0 04:26 ?        00:00:00 [kworker/3:0H-kblockd]
root          39       2  0 04:26 ?        00:00:00 [kworker/u65:0-flush-8:0]
root          43       2  0 04:26 ?        00:00:06 [kworker/u66:2-events_unbound]
root          44       2  0 04:26 ?        00:00:00 [kdevtmpfs]
root          45       2  0 04:26 ?        00:00:00 [kworker/R-inet_]
root          46       2  0 04:26 ?        00:00:00 [kworker/u65:1-events_unbound]
root          47       2  0 04:26 ?        00:00:00 [kauditd]
root          48       2  0 04:26 ?        00:00:00 [kworker/0:2-cgroup_destroy]
root          49       2  0 04:26 ?        00:00:00 [khungtaskd]
root          50       2  0 04:26 ?        00:00:00 [oom_reaper]
root          52       2  0 04:26 ?        00:00:00 [kworker/R-write]
root          53       2  0 04:26 ?        00:00:00 [kcompactd0]
root          54       2  0 04:26 ?        00:00:00 [ksmd]
root          56       2  0 04:26 ?        00:00:00 [kworker/2:2-rcu_par_gp]
root          57       2  0 04:26 ?        00:00:00 [khugepaged]
root          58       2  0 04:26 ?        00:00:00 [kworker/R-kinte]
root          59       2  0 04:26 ?        00:00:00 [kworker/R-kbloc]
root          60       2  0 04:26 ?        00:00:00 [kworker/R-blkcg]
root          63       2  0 04:26 ?        00:00:00 [kworker/R-tpm_d]
root          64       2  0 04:26 ?        00:00:00 [kworker/R-edac-]
root          65       2  0 04:26 ?        00:00:00 [kworker/R-devfr]
root          66       2  0 04:26 ?        00:00:00 [kworker/3:1H]
root          67       2  0 04:26 ?        00:00:00 [kswapd0]
```

- **Question 7** : what command displays processes as a tree on Linux?

Expand Me

To display all the processes as a process tree, you have to use the “pstree” command.

Background & Foreground Processes

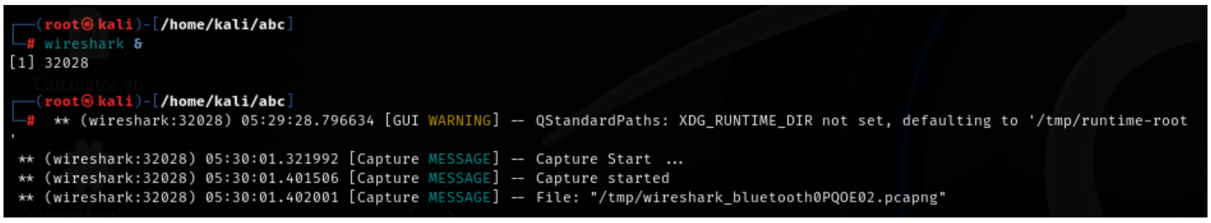


```
┌──(root㉿kali)-[/home/kali]
└─# pstree
systemd─┬─ModemManager───3*[{ModemManager}]
        ├─NetworkManager───3*[{NetworkManager}]
        ├─accounts-daemon───3*[{accounts-daemon}]
        ├─agetty
        ├─colord───3*[{colord}]
        ├─cron
        ├─dbus-daemon
        ├─haveged
        ├─lightdm─┬─Xorg───{Xorg}
        │         ├─lightdm─┬─xfce4-session─┬─Thunar───4*[{Thunar}]
        │         │         │               ├─agent───3*[{agent}]
        │         │         │               ├─applet.py
        │         │         │               ├─blueman-applet───4*[{blueman-applet}]
        │         │         │               ├─light-locker───4*[{light-locker}]
        │         │         │               ├─nm-applet───5*[{nm-applet}]
        │         │         │               ├─polkit-gnome-au───3*[{polkit-gnome-au}]
        │         │         │               ├─ssh-agent
        │         │         │               ├─xfce4-panel─┬─panel-1-whisker───4*[{panel-1-whisker}]
        │         │         │               │             ├─panel-13-cpugra───4*[{panel-13-cpugra}]
        │         │         │               │             ├─panel-14-systra───3*[{panel-14-systra}]
        │         │         │               │             ├─panel-15-genmon───4*[{panel-15-genmon}]
        │         │         │               │             ├─panel-16-pulsea───4*[{panel-16-pulsea}]
        │         │         │               │             ├─panel-17-notifi───4*[{panel-17-notifi}]
        │         │         │               │             ├─panel-18-power-───4*[{panel-18-power-}]
        │         │         │               │             ├─panel-22-action───4*[{panel-22-action}]
        │         │         │               │             └─4*[{xfce4-panel}]
        │         │         │               ├─xfce4-power-man───3*[{xfce4-power-man}]
        │         │         │               ├─xfdesktop───4*[{xfdesktop}]
        │         │         │               ├─xfsettingsd───3*[{xfsettingsd}]
        │         │         │               ├─xfwm4───13*[{xfwm4}]
        │         │         │               ├─xiccd───3*[{xiccd}]
        │         │         │               └─3*[{xfce4-session}]
        │         │         └─3*[{lightdm}]
        │         └─3*[{lightdm}]
        ├─polkitd───3*[{polkitd}]
        ├─qterminal─┬─zsh───sudo───sudo───su───zsh───pstree
        │           └─2*[{qterminal}]
        ├─rtkit-daemon───2*[{rtkit-daemon}]
        ├─systemd─┬─(sd-pam)
        │         ├─at-spi-bus-laun─┬─dbus-daemon
        │         │                 └─4*[{at-spi-bus-laun}]
        │         ├─at-spi2-registr───3*[{at-spi2-registr}]
        │         ├─dbus-daemon
        │         ├─dconf-service───3*[{dconf-service}]
        │         ├─gnome-keyring-d───4*[{gnome-keyring-d}]
        │         ├─gpg-agent
        │         ├─gvfs-afc-volume───4*[{gvfs-afc-volume}]
        │         ├─gvfs-goa-volume───3*[{gvfs-goa-volume}]
        │         ├─gvfs-gphoto2-vo───3*[{gvfs-gphoto2-vo}]
        │         ├─gvfs-mtp-volume───3*[{gvfs-mtp-volume}]
        │         ├─gvfs-udisks2-vo───4*[{gvfs-udisks2-vo}]
        │         ├─gvfsd─┬─gvfsd-dnssd───3*[{gvfsd-dnssd}]
        │         │       ├─gvfsd-network───4*[{gvfsd-network}]
        │         │       ├─gvfsd-trash───4*[{gvfsd-trash}]
        │         │       └─3*[{gvfsd}]
        │         ├─gvfsd-fuse───6*[{gvfsd-fuse}]
        │         ├─gvfsd-metadata───3*[{gvfsd-metadata}]
        │         ├─obexd
```

- **Question 8** : what syntax is used on Linux in order to execute a process in the background?

Expand Me

To execute a process in the background, you have to append a “&” sign at the end of the command.

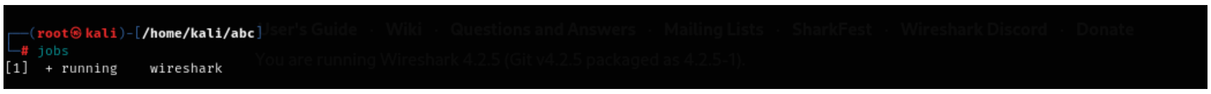


```
┌──(root㉿kali)-[/home/kali/abc]
└─# wireshark &
[1] 32028

┌──(root㉿kali)-[/home/kali/abc]
└─#  ** (wireshark:32028) 05:29:28.796634 [GUI WARNING] -- QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root
'

 ** (wireshark:32028) 05:30:01.321992 [Capture MESSAGE] -- Capture Start ...
 ** (wireshark:32028) 05:30:01.401506 [Capture MESSAGE] -- Capture started
 ** (wireshark:32028) 05:30:01.402001 [Capture MESSAGE] -- File: "/tmp/wireshark_bluetooth0PQOE02.pcapng"
```

- **Question 9** : what is the term that describes a process that was started in a terminal shell?

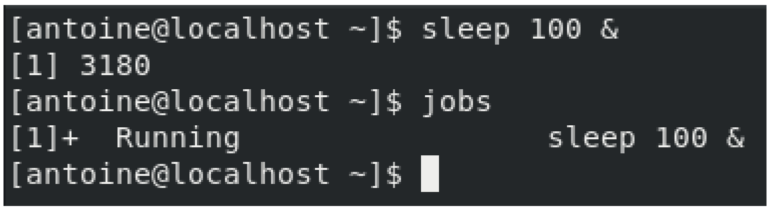
Expand Me

A process executed in a shell is called a “job” and the jobs command displays your current shell jobs.



```
┌──(root㉿kali)-[/home/kali/abc]
└─# jobs
[1]  + running    wireshark
```

- **Question 10** : you executed a command in the background, but you want to have your process executed in the foreground.



```
[antoine@localhost ~]$ sleep 100 &
[1] 3180
[antoine@localhost ~]$ jobs
[1]+  Running                 sleep 100 &
[antoine@localhost ~]$ █
```

What command would you execute?

Expand Me

The job id is 1 so you would execute “fg %1”

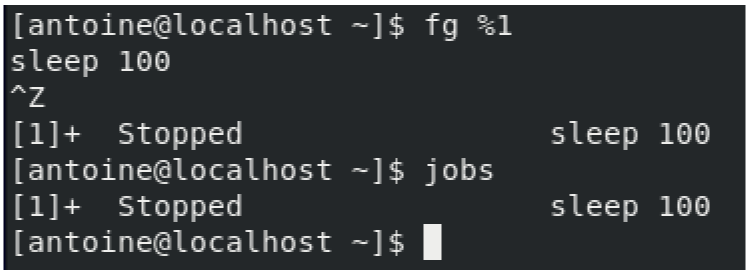
- **Question 11** : your process is now executed in the **foreground**.

What controls would you hit on your keyboard in order to stop the process (and not kill it) ?

Expand Me

In order to stop a process, or to send a SIGSTOP signal to a process, you have to hit Ctrl + Z.

- **Question 12** : your process is now interrupted.

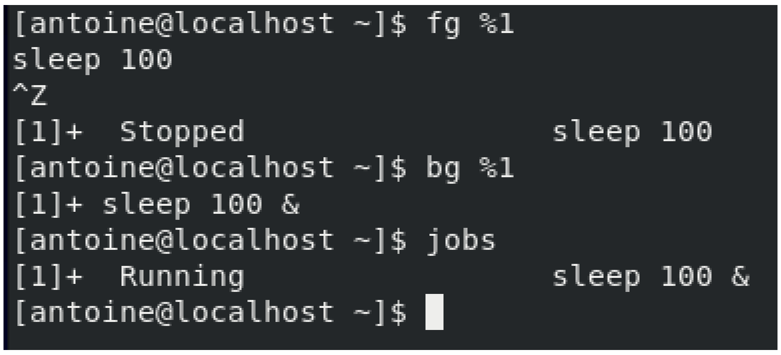


```
[antoine@localhost ~]$ fg %1
sleep 100
^Z
[1]+  Stopped                 sleep 100
[antoine@localhost ~]$ jobs
[1]+  Stopped                 sleep 100
[antoine@localhost ~]$ 
```

How would you resume the execution in the background?

Expand Me

In order to resume the execution, you can execute the “bg %1” command.



```
[antoine@localhost ~]$ fg %1
sleep 100
^Z
[1]+  Stopped                 sleep 100
[antoine@localhost ~]$ bg %1
[1]+ sleep 100 &
[antoine@localhost ~]$ jobs
[1]+  Running                 sleep 100 &
[antoine@localhost ~]$ 
```

- **Question 13** : what keys can you hit on your keyboard in order to send a SIGINT to a process in the foreground?

Expand Me

In order to send a SIGINT to a signal in the foreground, you would have to hit Ctrl + C.

__**Advanced Processes Commands**__

- **Question 14** : what command in used on Linux in order to list all processes given a specific pattern?

Expand Me

To search for processes given a specific pattern, you can use the “pgrep” command with the following syntax “pgrep “

- **Question 15** : what command would you use in order to easily kill (SIGKILL) all processes starting with “fire” ?

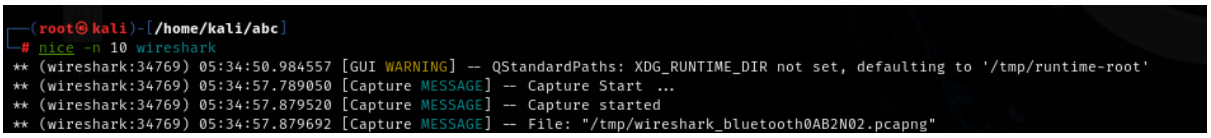
Expand Me

To kill all processes starting with “fire”, you would execute “pkill fire*”

- **Question 16** : on Linux, what command is used in order to execute a process with a custom priority level?

Expand Me

"Nice" is the command used to execute a command with a custom priority, in order for it to use more or less CPU resources.



```
┌──(root㉿kali)-[/home/kali/abc]
└─# nice -n 10 wireshark
** (wireshark:34769) 05:34:50.984557 [GUI WARNING] -- QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
** (wireshark:34769) 05:34:57.789050 [Capture MESSAGE] -- Capture Start ...
** (wireshark:34769) 05:34:57.879520 [Capture MESSAGE] -- Capture started
** (wireshark:34769) 05:34:57.879692 [Capture MESSAGE] -- File: "/tmp/wireshark_bluetooth0AB2N02.pcapng"
```

- **Question 17** : a process has a nice level of 19, is it going to use as much resources as possible?

Expand Me

No, the nicer the process, the more you are willing to share resources with others. As a consequence, the process has a very low priority level.

- **Question 18** : what is the default nice level when processes are created on Linux?

Expand Me

By default, processes are created with a nice level of 0.

- **Question 19**: as a non sudo-user, can you create a process with a nice level of -5?

Expand Me

No, non sudo users are not able to create processes with a nice level lower than the default one assigned. Moreover, when you created a process with a custom nice level, you are not able to lower it, even if it is greater than zero.

- **Question 20** : what command can be used in order to set the priority of a running process on Linux?

Expand Me

To customize the priority of a running process, you have to use the "renice" command with this syntax "renice -n "