

Creamos proceso

```
yes>/dev/null &
```

Mostramos su estado usando ps:

```
root@Lubuntu:/home/mario# yes>/dev/null &
[1] 43228
root@Lubuntu:/home/mario# ps
  PID TTY          TIME CMD
 43195 pts/2        00:00:00 sudo
 43196 pts/2        00:00:00 su
 43197 pts/2        00:00:00 bash
 43228 pts/2        00:00:05 yes
 43229 pts/2        00:00:00 ps
root@Lubuntu:/home/mario#
```

Repetimos para crear otro proceso.

```
root@Lubuntu:/home/mario# yes>/dev/null &
[2] 43235
root@Lubuntu:/home/mario# ps
  PID TTY          TIME CMD
 43195 pts/2        00:00:00 sudo
 43196 pts/2        00:00:00 su
 43197 pts/2        00:00:00 bash
 43228 pts/2        00:04:54 yes
 43235 pts/2        00:00:02 yes
 43236 pts/2        00:00:00 ps
root@Lubuntu:/home/mario#
```

Paramos los dos procesos usando `kill -19` y número de proceso.

```
root@Lubuntu:/home/mario# kill -19 43228
root@Lubuntu:/home/mario# kill -19 43235

[1]- Detenido          yes > /dev/null
[2]+ Detenido          yes > /dev/null
root@Lubuntu:/home/mario#
```

Reanudamos el primer proceso usando `kill -18` y número de proceso.

Subrayado en amarillo se puede comprobar que ese proceso sigue corriendo en el tiempo y el proceso inferior esta parado.

```
root@Lubuntu:/home/mario# kill -18 43228
root@Lubuntu:/home/mario# ps
  PID TTY          TIME CMD
 43195 pts/2        00:00:00 sudo
 43196 pts/2        00:00:00 su
 43197 pts/2        00:00:00 bash
 43228 pts/2        00:07:41 yes
 43235 pts/2        00:02:59 yes
 43239 pts/2        00:00:00 ps
root@Lubuntu:/home/mario# pa
pa: orden no encontrada
root@Lubuntu:/home/mario# ps
  PID TTY          TIME CMD
 43195 pts/2        00:00:00 sudo
 43196 pts/2        00:00:00 su
 43197 pts/2        00:00:00 bash
 43228 pts/2        00:07:49 yes
 43235 pts/2        00:02:59 yes
 43255 pts/2        00:00:00 ps
root@Lubuntu:/home/mario#
```

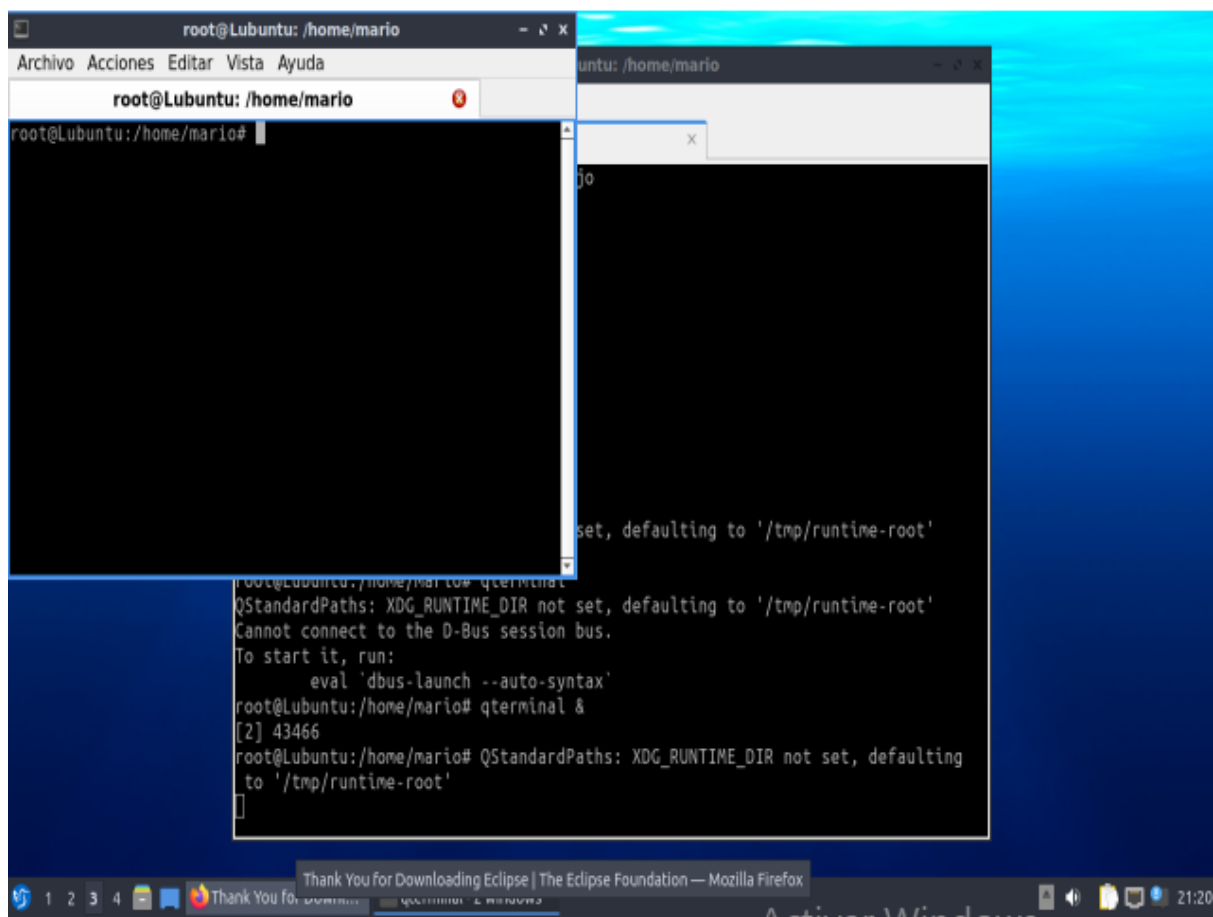
Al usar el comando “bg” comprobamos que el proceso está en segundo plano:

```
root@Lubuntu:/home/mario# bg
[2]+ yes > /dev/null &
root@Lubuntu:/home/mario#
```

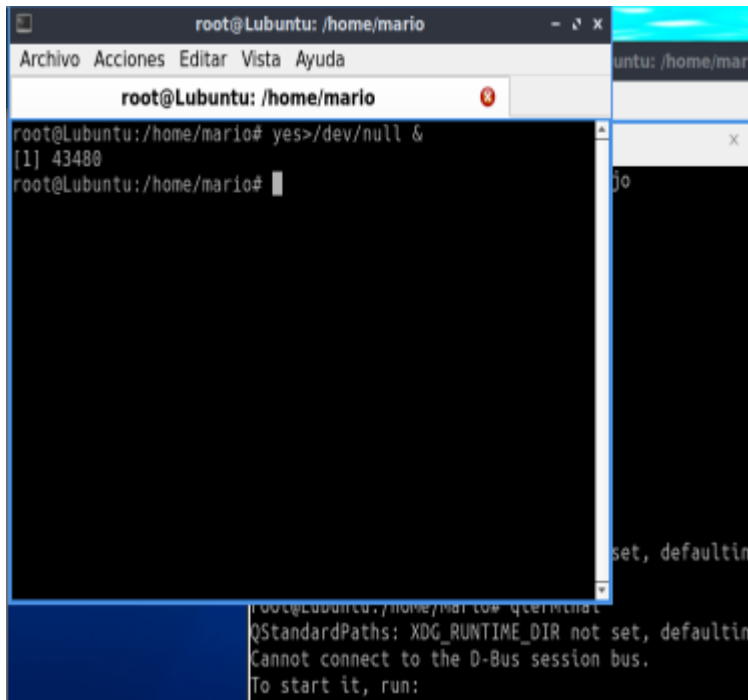
Al usar el comando “fg” se comprueba que el proceso está en primer plano, no especificamos proceso porque el otro proceso está parado, por lo que coge el que queda activo para realizar el “fg”.

```
root@Lubuntu:/home/mario# fg
yes > /dev/null
```

Usamos el comando `qterminal &` para abrir otra terminal desde el terminal “padre” en segundo plano.



Como vemos se ejecutan 2 terminales

A screenshot of a terminal window titled 'root@Lubuntu: /home/mario'. The terminal shows the command 'yes>/dev/null &' being executed, which returns '[1] 43480'. The prompt then returns to 'root@Lubuntu: /home/mario#'.

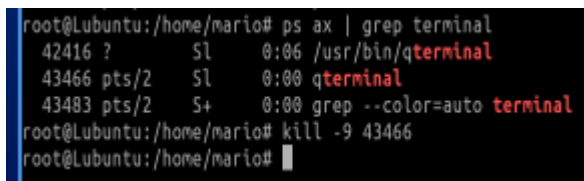
```
root@Lubuntu: /home/mario# yes>/dev/null &
[1] 43480
root@Lubuntu: /home/mario#
```

Ejecutamos un proceso infinito dentro del proceso hijo.

Ahora paramos el proceso del terminal hijo desde el padre.

Lo buscamos con el comando `ps ax | grep terminal`, cogemos su código 43466

Matamos el proceso usando `kill -9 43466`

A screenshot of a terminal window showing the output of 'ps ax | grep terminal' and the execution of 'kill -9 43466'. The process list shows PID 43466 as 'qterminal'. After the kill command, the prompt returns to 'root@Lubuntu: /home/mario#'.

```
root@Lubuntu: /home/mario# ps ax | grep terminal
42416 ?        Sl      0:06 /usr/bin/qterminal
43466 pts/2    Sl      0:00 qterminal
43483 pts/2    S+      0:00 grep --color=auto terminal
root@Lubuntu: /home/mario# kill -9 43466
root@Lubuntu: /home/mario#
```

El proceso infinito que creamos anteriormente desaparece, no queda como proceso zombi en la máquina.

En el caso de que haya un proceso Abuelo/Padre/Hijo y del Hijo haya un proceso infinito, al matar al proceso Padre, el Hijo se mantiene ejecutando el proceso infinito.