No se usa terminales graficas por utilizar muchos recursos

Terminales: de 1-6

La 7 es interfaz grafica

COmandos

Ctl-Alt # cambia a terminal #

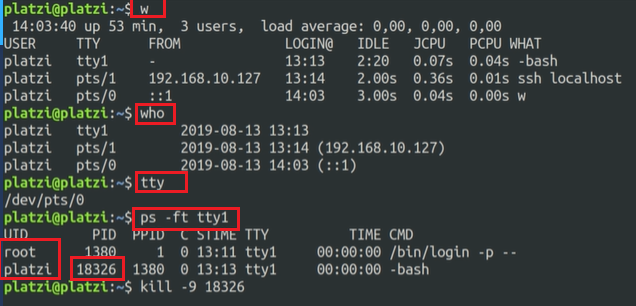
chvt # cambia a terminal #

tty muestra terminal actual

who muestra usuarios conectados e informacion

w igual a who pero mas informacion

kill -9 proc mata proceso



ps -aux | grep platzi

ps -ef | grep platzi

jobs igual a -aux

fg un proceso pausado o en backgroup, lo trae foreground

-nohup genera archivo “nohup.out” con informacion que produjo proceso

nohup commando &

nohup commando & > archivo.txt

**Simbolos especiales**

| pipe envia standard output de un commando a standar input de otro commando

& envia un proceso a background

./ ejecuta un archive

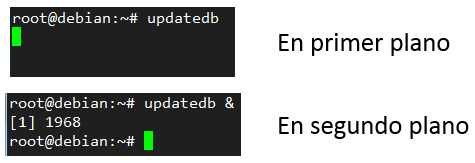
Que corre y quien lo corre



En algunos casos, un [script](https://blog.carreralinux.com.ar/category/scripts/) puede tomar un tiempo considerable en completar su ejecución, o un programa puede ocupar la línea de comandos mientras se encuentre corriendo. Para permitirnos volver a tomar el control de la terminal en cuestión, la shell nos permite colocar procesos en segundo plano. También podemos iniciar directamente un proceso en segundo plano, (background) de manera que no ocupe la terminal mientras corre.

Para que un proceso se inicie en segundo plano, colocaremos el símbolo **&** al fin del mismo.

updatedb &

[](https://blog.carreralinux.com.ar/wp-content/uploads/2016/09/procesos-en-segundo-plano-1-1.png)

Al iniciar un proceso en segundo plano, la terminal queda libre para seguir trabajando en la misma y se nos provee la identificación de dicho proceso y del [PID](https://blog.carreralinux.com.ar/2016/07/introduccion-procesos-en-linux/) del mismo (1 y 1968,).

Si deseamos traer un proceso a primer plano, usamos comando **fg** seguido del identificador del mismo.

Si sucediera que no supiéramos el identificador del proceso en cuestión, podemos valernos del comando **jobs**.

EJEMPLO

Si queremos todos los archivos con la extensión .sh y guardar la lista de los mismos en listadescripts.txt: y mientras lo hace lo ponemos en background (Segundo plano)

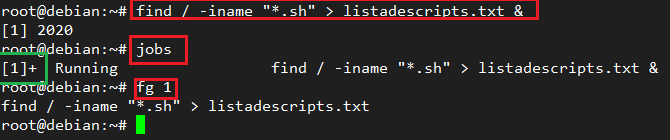
find / -iname "\*.sh" > listadescripts.txt &

Luego utilizamos  **jobs** y lo traeremos de nuevo a primer plano con **fg** seguido del identificador (1 en este caso), la secuencia es

find / -iname "\*.sh" > listadescripts.txt &

jobs

fg 1



2 – Listar procesos en segundo plano y traerlos a primer plano

Si el proceso hubiera sido iniciado en primer plano, podemos pausar su ejecución con Ctrl + z y luego utilizar el comando **bg** para enviarlo a segundo plano (como observamos en la Fig. 3). De ahí en más podemos proceder como explicamos anteriormente para traerlo nuevamente a primer plano si así lo deseamos.

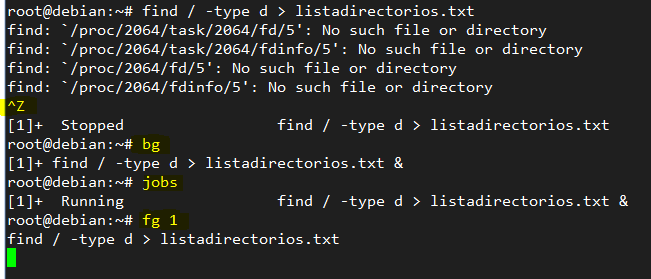
[](https://blog.carreralinux.com.ar/wp-content/uploads/2016/09/procesos-en-segundo-plano-3.png)

Figura 3

– Enviar procesos a segundo plano una vez que fueron iniciados

|  |  |
| --- | --- |
| script ss.sh  #!/bin/bash  echo "hola" | Cree script y que presente hola |
| chmod +x ss.sh  ./ss.sh &  nohup ./ss.sh & | Que sea ejecutable  Ejecutelo  Que genere una salida de que esta hacienda mientras corre en background |

Si estás corriendo un proceso en primer plano (monopolizando el uso del terminal) lo podes pausar primero con CTRL Z, luego pones el comando

bg %1

Con esto reactivas el proceso y lo mandas a segundo plano, liberando así la terminal para seguir trabajando.

También está el comando fg (foreground) que funciona opuesto al bg (background).