

¿QUÉ SON LOS TRABAJOS EN LINUX?

Un trabajo en Linux es un comando que se ha ejecutado desde de la terminal y aún no ha finalizado. En otras palabras un trabajo es un proceso que depende y está gestionado por la terminal.

Como veremos más adelante la totalidad de trabajos o tareas estarán reflejadas en una tabla de tareas o trabajos.

EJECUTAR UN TRABAJO EN PRIMER PLANO DESDE LA TERMINAL

Al ejecutar un comando simple como por ejemplo:

```
ls
```

El trabajo se ejecuta y termina de forma instantánea. Por lo tanto nunca veremos el trabajo listado en la lista de trabajos. Pero en el caso que ejecutemos el comando:

```
sleep 1000
```

El resultado es muy diferente. Lo es porque se ejecutará un trabajo que dejará la terminal de nuestro equipo inactiva durante 1000 segundos sin que nosotros podamos introducir nuevos comandos.

Si antes de los 1000 segundos quisiéramos ejecutar otro comando tendríamos que abrir otro emulador de terminal, o una terminal virtual.

CANCELAR Y SUSPENDER UN TRABAJO QUE CORRE EN PRIMER PLANO

En el apartado anterior generamos un trabajo que tardará 1000 segundos en finalizar.

Si queremos cancelar o suspender el trabajo que está en ejecución lo podemos hacer del siguiente modo.

Para cancelar el trabajo tan solo tenemos que presionar la combinación de teclas **Ctrl+C**

```
joan@debian:~$ sleep 1000  
^C
```

Para suspender el trabajo, con el fin de poderlo reanudar más tarde, tendremos que presionar la combinación de teclas **Ctrl+Z**

```
joan@debian:~$ sleep 1000  
^Z  
[1]+ Detenido sleep 1000
```

LANZAR VARIOS TRABAJOS EN LINUX PARA QUE SE EJECUTEN DE FORMA SIMULTANEA EN SEGUNDO PLANO

Gnu-Linux es un sistema multitarea, por lo tanto permite ejecutar varios trabajos de forma simultánea y cada uno de estos trabajos será identificado con un número de trabajo.

Para ejecutar trabajos de forma simultanea sin tener que abrir varios emuladores de terminal ni terminales virtuales tendremos que añadir el carácter `&` al final del comando que utilizaremos para ejecutar el trabajo. Por lo tanto en nuestro caso ejecutaremos el siguiente comando:

```
joan@debian:~$ sleep 2000 &  
[1] 54472
```

Justo después de ejecutar el comando veremos que la terminal sigue estando operativa para ejecutar más comandos.

A continuación lanzaremos el segundo trabajo ejecutando el siguiente comando en la terminal:

```
joan@debian:~$ sleep 10000 &  
[2] 55504
```

LISTAR LOS TRABAJOS QUE ESTÁN DETENIDOS O EJECUTÁNDOSE

Siguiendo con el apartado anterior deberíamos tener 2 trabajos activos ejecutándose en segundo plano. Para listarlos tan solo tenemos que ejecutar el comando `jobs` en la terminal:

```
joan@debian:~$ jobs  
[1]- Ejecutando sleep 2000 &  
[2]+ Ejecutando sleep 10000 &
```

Efectivamente vemos que los 2 comandos que lanzamos en el apartado anterior están activos y ejecutándose.

Para ver las opciones adicionales que ofrece el comando `jobs` ejecuten el siguiente comando en la terminal.

```
jobs --help
```

DETENER UN TRABAJO QUE ESTÁ EJECUTÁNDOSE EN SEGUNDO PLANO

Si ahora quisiéramos suspender el **trabajo número 1** tendríamos que usar el comando `fg` y la combinación de teclas `Ctrl+Z`.

Nota: El comando `fg` (foreground o primer plano) traerá a primer plano un trabajo que está ejecutándose en segundo plano. También se puede usar para reanudar en primer plano un trabajo que está suspendido o detenido.

Por lo tanto, para traer a primer plano la tarea 1 que se está corriendo en segundo plano ejecutaremos el siguiente comando en la terminal:

```
fg %1
```

Una vez la tarea esté en primer plano la suspenderemos presionando la combinación de teclas **Ctrl+Z**

```
joan@debian:~$ fg %1
sleep 2000
^Z
[1]+ Detenido sleep 2000
```

Una vez detenida la tarea ejecutaremos el comando `jobs` y veremos que efectivamente hemos detenido el trabajo 1 y el trabajo 2 sigue ejecutándose.

```
joan@debian:~$ jobs
[1]+ Detenido sleep 2000
[2]- Ejecutando sleep 10000 &
```

REANUDAR UN TRABAJO QUE ESTÁ DETENIDO

Para volver a reanudar el trabajo que acabamos de detener podemos usar los comandos `fg` o `bg`.

Nota: El comando `bg` (background o segundo plano) se usa para reanudar un trabajo que está suspendido en segundo plano.

Por lo tanto para reanudar el comando que suspendimos en el apartado anterior ejecutaremos el comando `bg` seguido por el **número de trabajo a reanudar**:

```
joan@debian:~$ bg %1
[1]+ sleep 1000 &
```

Una vez ejecutado el comando se reanudará la ejecución del trabajo en segundo plano. Para comprobarlo ejecutaremos el comando `jobs` en la terminal:

```
joan@debian:~$ jobs
[1]- Ejecutando sleep 2000 &
[2]+ Ejecutando sleep 10000 &
```

Nota: Para reanudar el trabajo que teníamos detenido en primer plano deberíamos haber ejecutado el comando `fg %1`

MATAR UN TRABAJO QUE ESTÁ DETENIDO O EJECUTÁNDOSE

Para matar un trabajo que está detenido o ejecutándose se puede hacer de 2 formas. Lo veremos claro mediante el siguiente ejemplo.

Para matar o cancelar el trabajo 1 podemos traerlo a primer plano ejecutando el siguiente comando en la terminal:

```
fg %1
```

Una vez el trabajo esté ejecutándose en primer plano lo mataremos presionando la siguiente combinación de teclas:

```
Ctrl+C
```

Si ahora listamos los trabajos activos mediante el comando jobs veremos que únicamente hay un solo trabajo activo.

```
joan@debian:~$ jobs
[2]+ Ejecutando sleep 10000 &
```

Otra opción para matar un trabajo es mediante el comando kill. Si ahora queremos matar el trabajo 2 tan solo tendríamos que ejecutar el comando kill seguido de % y el número de trabajo que queremos matar. Por lo tanto para matar el trabajo 2 ejecutaremos el siguiente comando:

```
joan@debian:~$ kill %2
```

En estos momento no quedará ningún trabajo activo. Si listamos los trabajos con el comando jobs veremos que el trabajo 2 está terminado.

```
joan@debian:~$ jobs
[2]+ Terminado sleep 10000
```

Nota: El carácter % es para indicar que se trata de un trabajo y no de un proceso normal.

DESACOPLAR TRABAJOS DE LA TERMINAL

Un trabajo que está ejecutándose en un emulador de terminal finalizará automáticamente en el momento que cerremos la terminal. Para evitar esto existen soluciones como [no-hup](#), screen o tmux. Otra solución es desacoplar el trabajo de la terminal mediante el comando [disown](#).

Para que se entienda de forma más clara lanzaremos 2 trabajos ejecutando los siguientes comandos en la terminal.

```
joan@debian:~$ sleep 10000 &
[1] 144164
joan@debian:~$ sleep 20000 &
[2] 144193
```

Acto seguido listaremos los trabajos mediante el comando jobs.

```
joan@debian:~$ jobs
[1]- Ejecutando sleep 10000 &
[2]+ Ejecutando sleep 20000 &
```

Si queremos que el trabajo 2 continúe activo después de cerrar la terminal ejecutaremos el siguiente comando:

```
joan@debian:~$ disown %2
```

Si ahora listamos los trabajos mediante el comando jobs veremos que el trabajo 2 ha desaparecido:

```
joan@debian:~$ jobs
[1]- Ejecutando sleep 10000 &
```

Nota: Si quisiéramos desacoplar el proceso sin que saliera de la tabla de trabajos deberíamos haber ejecutado el comando `disown -h %2`

Pero aunque haya desaparecido sigue ejecutándose. Y lo podemos comprobar ejecutando el siguiente comando en la terminal:

```
joan@debian:~$ ps -aux | grep sleep
```

```
joan 144164 0.0 0.0 7992 816 pts/2 S 15:17 0:00 sleep 10000
joan 144193 0.0 0.0 7992 892 pts/2 S 15:17 0:00 sleep 20000
joan 144590 0.0 0.0 8656 912 pts/2 S+ 15:18 0:00 grep sleep
```

Si ahora cerramos la terminal y abrimos otra nueva veremos que la tabla de trabajos está completamente vacía.

```
joan@debian:~$ jobs
```

Pero si miramos la tabla de procesos vemos que la tarea que desacoplamos de la terminal sigue ejecutándose sin problema alguno.

```
joan@debian:~$ ps -aux | grep sleep
joan 144193 0.0 0.0 7992 892 pts/2 S 15:17 0:00 sleep 20000
joan 144920 0.0 0.0 8656 912 pts/2 S+ 15:19 0:00 grep sleep
```

Si ahora por ejemplo queremos matar definitivamente el proceso que aun sigue ejecutándose tendremos que ejecutar el siguiente comando.

```
kill 144193
```

De esta forma tan simple y sencilla podemos gestionar trabajos desde la terminal de Linux.