

[Next](#)
[Up](#)
[Previous](#)
[Index](#)

Siguiente: [Manejo de la prioridad](#)
Subir: [Comandos básicos para la](#)
Anterior: [Ver los procesos en Índice de Materias](#)

Subsecciones

- [Otros comandos](#)
 - [pgrep](#)
 - [pkill](#)
 - [killall](#)
 - [nohup](#)
 - [exec](#)

Señalización de procesos

El comando básico para enviar señales a un proceso es `kill`

- Ctrl-C y Ctrl-Z son atajos para enviar señales SIGINT (2) y SIGTSTP (20)
- `kill -l` lista el conjunto de señales

```
$ kill -l
1) SIGHUP      2) SIGINT      3) SIGQUIT     4) SIGILL
5) SIGTRAP     6) SIGABRT    7) SIGBUS      8) SIGFPE
9) SIGKILL     10) SIGUSR1    11) SIGSEGV     12) SIGUSR2
13) SIGPIPE    14) SIGALRM    15) SIGTERM     17) SIGCHLD
18) SIGCONT    19) SIGSTOP    20) SIGTSTP     21) SIGTTIN
22) SIGTTOU    23) SIGURG     24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM 27) SIGPROF    28) SIGWINCH    29) SIGIO
30) SIGPWR     31) SIGSYS     ....
```

- para ver su significado, ver `man 7 signal`

Sintaxis de kill

`kill [señal] PID`

- *señal* puede indicarse mediante el número o el código:
 - `kill -9` y `kill -KILL` son equivalentes
- las señales más comunes son:
 - SIGHUP (1): cuelgue del terminal o muerte del proceso controlador
 - SIGTERM (15): mata el proceso permitiéndole terminar correctamente
 - SIGKILL (9): mata el proceso sin permitirle terminar
 - SIGSTOP (19): para el proceso
 - SIGCONT (18): continúa si parado
 - SIGINT (2): interrupción de teclado (Ctrl-C)
 - SIGTSTP (20): stop de teclado (Ctrl-Z)
 - SIGQUIT (3): salida de teclado (Ctrl-\)

Algunas características de las señales:

- La señal que se envía por defecto es TERM (15)
 - los procesos pueden ignorar esta señal y no terminar
 - las señales KILL (9) y STOP (19) no pueden ignorarse
- En bash, cuando enviamos una señal SIGHUP a un shell, este se lo reenvía a todos sus hijos
- Cuando cerramos un terminal en un entorno gráfico, o abandonamos una sesión, se envía un SIGHUP a todos sus hijos
- La mayoría de los demonios (*daemons*) responden a la señal SIGHUP volviendo a leer sus ficheros de configuración:
 - en vez de matar y reiniciar un demonio podemos hacer un `kill -HUP` para reiniciarlo

Ejemplos

```
$ yes >/dev/null &
[1] 9848
$ yes >/dev/null &
[2] 9849
$ ps
PID TTY TIME CMD
9834 pts/7 00:00:00 bash
9848 pts/7 00:00:02 yes
9849 pts/7 00:00:01 yes
9850 pts/7 00:00:00 ps
$ kill -STOP 9849
[2]+ Stopped                  yes >/dev/null
$ jobs
[1]-  Running                  yes >/dev/null &
[2]+ Stopped                  yes >/dev/null
$ kill -CONT 9849
$ jobs
[1]-  Running                  yes >/dev/null &
[2]+ Running                  yes >/dev/null &
$ kill -KILL 9848
$ kill -1 9849
[1]-  Matado                   yes >/dev/null
[2]+  Colgar                   yes >/dev/null
```

Otros comandos

pgrep

busca en la lista de procesos para localizar el PID a partir del nombre (similar a `ps | grep`)

- Ejemplo:

```
$ pgrep -u root sshd # PID del proceso sshd de root
```

pgrep

permite enviar señales a los procesos indicándolos por nombre en vez de por PID

- Ejemplo:

```
$ pkill -9 proceso
```

- si hay varios procesos con el mismo nombre los mata a todos
- en vez de un nombre admite un patrón (p.e. `pkill 'l.*'`)
 - tener cuidado con su uso (es fácil matar procesos de forma errónea)

pkill

similar a `pkill`, pero no admite patrones en el nombre, y tiene otras opciones

killall

normalmente, cuando salimos de un login shell (`logout`) o cerramos una terminal, se envía una señal `SIGHUP` a todos los procesos hijos [1](#):

- si lanzamos un proceso en background y salimos de la sesión el proceso se muere al morir el shell desde el que lo iniciamos

El comando `nohup` permite que un lanzar un comando ignorando las señales `SIGHUP`

```
nohup comando
```

- la salida del comando se redirige al fichero `nohup.out`

nohup

`exec` ejecuta un comando reemplazando al shell desde el que se lanza

Ejemplos:

```
$ yes > /dev/null &
[1] 14724
$ yes > /dev/null &
[2] 14725
$ ps
```

```
PID TTY TIME CMD
7083 pts/3 00:00:00 bash
14724 pts/3 00:00:02 yes
14725 pts/3 00:00:02 yes
14726 pts/3 00:00:00 ps
$ pgrep yes
14724
14725
$ pkill -9 yes
$ ps
PID TTY TIME CMD
7083 pts/3 00:00:00 bash
14730 pts/3 00:00:00 ps
[1]- Matado          yes > /dev/null
[2]+ Matado          yes > /dev/null
```

Más ejemplos:

```
$ nohup yes > /dev/null &
[1] 9620
$ kill -HUP 9620
$ ps
PID TTY TIME CMD
8293 pts/5 00:00:00 bash
9620 pts/5 00:00:13 yes
9621 pts/5 00:00:00 ps
$ kill 9620
[1]+ Terminado          nohup yes > /dev/null
```

[Next](#) [Up](#) [Previous](#) [Index](#)

Siguiente: [Manejo de la prioridad](#) **Subir:** [Comandos básicos para la](#) **Anterior:** [Ver los procesos en Índice de Materias](#)

[Administración de Sistemas e Redes <ASR.USC\[at\]gmail.com>](#)

[Tomás Fernández Pena <tf.pena\[at\]usc.es>](#)

Última actualización: 19-10-15 11:23 por tomas



Curso de Administración de Sistemas y Redes por [Tomás Fernández Pena](#) se distribuye bajo la licencia [Creative Commons Reconocimiento-Compartir baixo a mesma licencia. 3.0 España](#). Trabajo original en [persoal.citius.usc.es/tf.pena/ASR](#).