

Gestión de procesos en la terminal de Ubuntu

Salvo los comandos internos, cualquier tarea crea en Linux un proceso identificado con un PID.

Visualización de procesos

Para ver los procesos tenemos el comando

`ps <identificador del proceso>`

Cuyos resultados se estructuran en un listado con las siguientes columnas:

PID	Identificador del proceso
TTY	Nº de la terminal
TIME	Tiempo
Command	Comando que lanza el proceso

El comando ps dispone de decenas de opciones, por ejemplo:

`ps aux` muestra todos los procesos en detalle del sistema

`ps axjf` que mostrará un árbol jerárquico con la ruta del programa al que pertenece el proceso

`ps -u <nombre usuario>` para ver los procesos de un usuario

Para ser más precisos se puede buscar un determinado proceso con el que queramos dar con la ayuda de un pipe y el filtro grep

Por ejemplo, para buscar el proceso SSH usamos:

`ps aux | grep ssh`

Los procesos pueden ejecutar otros procesos (procesos hijos). Los procesos hijos contienen su propio PID pero a su vez también contienen otro campo llamado PPID (Parent Process Identifier) que es el proceso padre al que pertenece el proceso.

Todos los procesos en Linux tienen su proceso padre excepto el proceso 0 que es el que ejecuta el inicio del sistema (PID 1)

Para ver el PPID de los distintos procesos, utilizamos

`ps -ef`

Para ver los procesos en forma de árbol podemos emplear:

`pstree`

Y para ver el PID de un proceso por su nombre empleamos:

`pidof <nombre>`

Para ver el nombre (o comando que ha lanzado) un proceso tenemos que ir al registro de cada proceso almacenado en el directorio `proc`

`nano /proc/<proceso>/cmdline`

Existen también otros programas para ver los procesos como `top` o `htop` que muestran los procesos de manera dinámica

`top`

Es posible ordenar por uno de los campos de la tabla, por ejemplo en sentido descendente por el campo CPU

`top -o +%CPU`

Para comprobar usamos `yes`

`yes > /dev/null & disown`

Y matamos el proceso con

`killall -9 yes`

Estados de un proceso

Los procesos durante su ejecución pasan por distintos estados, algunos de ellos:

- S (sleeping) el proceso está en espera.
- R (running) el proceso está en ejecución.
- T (stop) el proceso se encuentra parado.
- D proceso que se encuentra bloqueado a la espera de un recurso.
- Z (zombie) es un proceso que se encuentra en estado zombie, es decir, que es un proceso que ha finalizado pero que su proceso padre sigue en ejecución y no se ha “dado cuenta” de la circunstancia de su proceso hijo.

Finalización de procesos

A los procesos se les manda señales para modificar su comportamiento a través del núcleo.

Esto se realiza mediante el comando kill con sus correspondientes señales e indicando el número de PID. Algunas de las señales que se pueden mandar a los procesos son:

- SIGINT ó – 2, interrumpe un proceso, equivale a pulsar CTRL+C
- SIGKILL ó – 9, mata un proceso y no hay vuelta atrás.
- SIGTERM ó – 15, es como SIGINT pero de una forma más “ordenada”. Es la señal que se manda por defecto si al comando kill no se le indica ninguna señal.
- SIGCOUNT ó – 18, reanudar un proceso que se ha parado por ejemplo con SIGSTOP.
- SIGSTOP – 19, parar un proceso, es igual que pulsar CTRL+Z

La sintaxis sería por ejemplo:

```
kill -9 4900  
kill -SIGKILL 4900
```

Donde 4900 es el PID del proceso

Con el comando killall también se mandan señales, pero actúa sobre el nombre del programa, y por tanto sobre todos sus procesos. Por ejemplo:

```
killall firefox
```

Matar desde la ventana

```
xkill
```

Prioridad de los procesos

La prioridad de cada proceso se puede observar en la columna PR del comando top junto con la columna NI (nice) que muestra lo contrario (nice será mayor si consume menos recursos lo que suele coincidir con una menor prioridad).

De esta manera modificando el valor de nice de -20 mayor prioridad a +20 menor prioridad podemos influir en la prioridad del proceso. Para ello se usa el comando nice:

```
sudo nice -n -20 <nombreproceso>
```

Para un proceso que se encuentre en ejecución usaremos:

```
renice 20 <nombreproceso>
```