



MANEJO DE TAREAS Y PROCESOS EN LINUX.

Ingresa a un shell de línea de comandos de su equipo .

Multitarea (ejecutando más de una tarea a la vez)

Control de trabajos

Control de trabajos se refiere a la habilidad de poner procesos (esencialmente, otra palabra para programas) en background (segundo plano) y ponerlos de vuelta en foreground (primer plano).

En *nix, la principal herramienta para el control de procesos es el shell, seguirá la pista de los procesos por usted, si se aprende como hablar su lenguaje.

Las dos palabras más importantes en ese lenguaje son **fg**, para primer plano, y **bg**, para segundo plano.

Para entender cómo funcionan, use el comando **yes** en el indicador del sistema.

Esto produce el maravilloso efecto de desplazar una larga columna de yes por la parte izquierda de la pantalla, tan rápido que no se pueden seguir. Para pararlo, se podría pulsar **<Ctrl+c>** y matarlo, pero esta vez Ud. oprimirá **<Ctrl+z>**. Parece haberse detenido, pero habrá un mensaje antes del indicador de sistema, más o menos parecido a este:

```
[1]+ Stopped yes
```

Significa que el trabajo **yes** se ha suspendido en el segundo plano. Se puede hacer que siga ejecutándose tecleando **fg** en el indicador de sistema, que lo pondrá en primer plano otra vez. Si se desea, se pueden hacer otras cosas antes, mientras está suspendido. Pruebe unos cuantos **ls** o algo antes de ponerlo en primer plano nuevamente.

Una vez que ha vuelto al primer plano, las yes empezarán a salir otra vez, tan rápido como antes.

(Ahora pulse **<Ctrl+c>** para matarlo de veras).

Analicemos el mensaje que obtuvimos del shell:

```
[1]+ Stopped yes
```

El número entre corchetes es el número de trabajo de este proceso, y se usará cuando se necesite referenciarlo específicamente.

Antes de seguir adelante, iniciemos este trabajo nuevamente y lo mataremos de forma diferente. Esta vez con un comando llamado **kill** que se usa del siguiente modo:

```
kill %1
```

```
[1]+ Stopped yes
```

Ese mensaje sobre el proceso que indica "parado" otra vez induce a error. Para saber si aún está vivo (eso es, tanto en ejecución como congelado en un estado suspendido), teclee **jobs**:

```
jobs
```

```
[1]+ Terminated yes
```

Ahora ya lo sabe: ¡el trabajo ha terminado!

Ahora, ejecute **yes** de nuevo, de esta forma:

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

Pulse **<Ctrl+z>**.

```
[1]+ Stopped yes >/dev/null
```

¿Hay alguna forma de ponerlo en ejecución en segundo plano, mientras deja el indicador del sistema para trabajar de forma interactiva? El comando para hacer eso es **bg**:

```
bg
```

```
[1]+ yes >/dev/null &
```



Licenciatura en Sistemas de Información
Sistemas Operativos - 3er año
Tp 5b – Laboratorio – Procesos en Linux

Después de teclear **bg**, el trabajo **yes > /dev/null** habrá continuado con su ejecución otra vez, pero esta vez en segundo plano.

Ahora hay dos formas diferentes de matarlo: con el comando **kill** que ya se explicó, o poniendo el trabajo en primer plano de nuevo e interrumpirlo con una interrupción, **<Ctrl+c>**. Probemos la segunda forma, sólo para entender la relación entre **fg** y **bg** un poco mejor:

```
fg
yes >/dev/null
```

Pulse **<Ctrl+c>**.

Bueno, se acabó. Ahora, ejecute unos cuantos trabajos simultáneamente, como estos:

```
yes > /dev/null &
[1] 1024
yes | sort > /dev/null &
[2] 1026
yes | uniq > /dev/null
```

Pulse **<Ctrl-z>** para suspenderlo..

```
[3]+ Stopped yes _ uniq >/dev/null
```

Cada uno indica su número de trabajo. Los dos primeros también muestran sus números de identificación de proceso, o **PID**, después del número de trabajo.

Matemos el segundo. Se puede teclear **"kill %2"**, pero eso sería demasiado fácil.

Por el contrario, haga esto:

```
fg %2
yes _ sort >/dev/null
```

Pulse **<Ctrl+c>** para matarlo.

Esto funciona por que el shell automáticamente interpreta un número de trabajo como una petición para poner ese trabajo en primer plano. Se puede indicar los números de trabajo con otros números precedidos por un **"%"**. Ahora teclee **"jobs"** para ver cuáles trabajos quedan en ejecución:

```
jobs
[1]-  Running yes >/dev/null &
[3]+  Stopped yes | uniq >/dev/null
```

El **"-"** indica que ese trabajo número 1 es segundo en la lista para ser puesto en el primer plano, si sólo se teclea **fg** sin dar parámetros. El **"+"** indica que el trabajo especificado es el primero en la lista un **fg** sin parámetros pondrá al trabajo número 3 en el primer plano. Sin embargo, se puede acceder a él llamándolo, si se desea, mediante:

```
fg %1
yes >/dev/null
```

Pulse **<Ctrl+z>** para suspenderlo.

```
[1]+ Stopped yes >/dev/null
```

Al cambiar al trabajo número 1 y luego suspenderlo han cambiado las prioridades de todos los trabajos de usuario. Esto se puede ver con el comando **jobs**:

```
jobs
[1]+ Stopped yes >/dev/null
[3]- Stopped yes | uniq >/dev/null
```

Ahora los dos están parados (porque los dos se han suspendido con **<Ctrl+z>**), y el número 1 es el siguiente en la lista a entrar en el primer plano por defecto. Esto es así porque se le puso en el primer plano manualmente, y luego fue suspendido. El **"+"** siempre se refiere al trabajo más reciente que ha sido suspendido del primer plano. Se puede continuar con su ejecución otra vez:

```
bg
[1]+ yes >/dev/null &
jobs
[1]-  Running yes >/dev/null
[3]+  Stopped yes | uniq >/dev/null
```



Licenciatura en Sistemas de Información
Sistemas Operativos - 3er año
Tp 5b – Laboratorio – Procesos en Linux

Fíjese que ahora está en ejecución, y el otro trabajo se ha movido en la lista y tiene el "+". Ahora matémoslos para que el sistema no esté permanentemente ralentizado por procesos que no hacen nada.

```
kill %1 %3
[3] Terminated yes | uniq >/dev/null
jobs
[1]+ Terminated yes >/dev/null
```

Procesos en Linux

Entre los atributos de un proceso, se encuentran:

- ⑩ Identificador (PID): Número único que el sistema asigna a cada proceso de tal forma que aunque los PID se puedan repetir en el sistema, nunca pueden hacerlo en el mismo momento.
- ⑩ Identificador del proceso padre (PPID).
- ⑩ Identidades del usuario y grupos.
- ⑩ Prioridad del proceso con respecto a otros.
- ⑩ Recursos consumidos por el proceso.
- ⑩ Archivos abiertos
- ⑩ Estado del proceso: Especifica la actividad que esta realizando el proceso.

Visualizando procesos

Antes de manipular procesos en Linux hay que poder ver los procesos en ejecución. Para esto se emplea el comando **ps**.

Sintaxis: **ps <opciones>**

Para ver las opciones del comando **ps** puede emplear el modificador **--help** o bien acceder a la pagina man de **ps**.

Dentro de las opciones más útiles se encuentran:

-a	Muestra todos los procesos inclusive de otras terminales.
l	Formato largo.
-F	Otro formato largo.
f	Muestra jerárquicamente los padres e hijos. (--forest)
x	Muestra procesos que no dependen de una shell.

Ingresa el comando **ps** y a continuación <ENTER>. ¿Qué procesos hay en ejecución?

La primera columna es el PID del proceso el cual nos permitirá identificar el proceso de aquí en adelante.

Ahora ingrese el comando **ps f** y a continuación <ENTER>. ¿Nota la diferencia?

Ahora ingrese el comando **ps l** y a continuación <ENTER>. ¿Qué significa cada columna de esta salida?

Tanto la opción **f** como la **l** tienen una columna **STAT** que indica el estado del proceso. Los posibles estados que puede indicar son:



Licenciatura en Sistemas de Información
Sistemas Operativos - 3er año
Tp 5b – Laboratorio – Procesos en Linux

D	durmiendo ininterrumpible (generalmente por E/S)
R	corriendo (o en cola de listos)
S	durmiendo
T	detenido
W	paginado
X	muerto
Z	defunct ("zombie")

Una variante mas completa de la opción **f** es el comando **ps tree**. (Sólo notará la diferencia ejecutándolo como **root**)



Matando procesos

En ocasiones es necesario finalizar un proceso de manera manual. Esto puede deberse a algún error del programa o nuestro. Para finalizar un proceso se utiliza el comando **kill**.

Sintaxis: **kill** <-señal> PID

Para ver las opciones del comando kill puede emplear el modificador **--help** o bien acceder a la página del manual de kill.

Ingrese el comando **top** y a continuación <ENTER>. Ahora tipee Ctrl+z a continuación ingrese el comando **ps f** y <ENTER>. ¿Qué estado tiene el proceso del top?

Ingrese el comando **kill PID** (donde **PID** es el PID del top mostrado por el **ps f**) y a continuación <ENTER>.

Ingrese nuevamente el comando **ps f** y <ENTER>. ¿Todavía aparece listado el proceso del top? ¿Qué estado tiene el proceso del top?

Para matar incondicionalmente a un proceso se usa el modificador **-KILL** o **-9**.

Ingrese el comando **kill -KILL PID** (donde **PID** es el PID del top mostrado por el **ps f**) y a continuación <ENTER>.

Ingrese nuevamente el comando **ps f** y <ENTER>. ¿Todavía aparece listado el proceso del top? ¿Qué estado tiene el proceso del top?

Además de la señal KILL a un proceso se le pueden enviar otras señales:

HUP	1
KILL	9
TERM	15
CONT	19,18,25
STOP	17,19,23

Es posible matar mas de un proceso a la vez usando el comando **killall**.

Sintaxis: **killall** <señal> proceso

Por ejemplo: **killall -KILL top**

Todo en uno

Existe un comando que permite ver estadísticas de los procesos en ejecución en nuestro sistema. Además de estas estadísticas este comando permite ver los procesos actualmente en el sistema y enviarle señales a estos. Este comando se denomina **top**.

Sintaxis: **top**

Para ver las opciones del comando top puede acceder a la página del manual de top.

Top es un comando interactivo (soporta interacción con el usuario):

Para acceder a la ayuda interna presione la tecla **h**.

Para enviar una señal a un proceso presione la tecla **k**. (Luego le pedirá el PID y la señal a enviar).

Para salir presione **q**.

Una versión mejorada es **htop**, pero es necesario instalarla. La sintaxis para su instalación es: **sudo apt-get install htop**