



## Lab 6.2: Modificar el ID de proceso máximo - maximum process ID

El comportamiento normal de un sistema **Linux** es que los IDs de proceso comiencen en PID=1 para el proceso **init**, el primer proceso en el sistema, y luego en forma secuencial a medida en que procesos nuevos van siendo creados y terminados (de forma constante).

However, when the PID reaches the value shorted in `/proc/sys/kernel/pid_max`, which is conventionally 32768 (32K), they will wrap around to lower numbers. If nothing else, this means you can't have more than 32K processes on the system since there are only that many slots for PIDs.

El comportamiento normal de un sistema **Linux** es que los IDs de proceso comiencen en PID=1 para el proceso **init**, el primer proceso en el sistema, y luego en forma secuencial a medida en que procesos nuevos van siendo creados y terminados (de forma constante).

Sin embargo, cuando el PID alcanza el valor especificado en `/proc/sys/kernel/pid_max`, el cual es generalmente 32768 (32K), se intentará utilizar números bajos. Si no hay más números disponibles no hay nada que hacer, ya que no es posible tener más de 32K procesos en el sistema (ese es el número máximo par los PIDs).

1. Obtenga el valor actual del PID máximo.
2. Averigue cuáles son los PIDs que está siendo creados actualmente.
3. Configure `pid_max` a un valor menor.
4. Inicie un proceso y vea qué valor de PID se le asigna.

## Solution 6.2

A continuación vamos a usar dos métodos; uno usando **sysctl**, el otro escribiendo valores directamente con **echo** a `/proc/sys/kernel/pid_max`. Note que el método **echo** requiere ser **root**, **sudo** no funcionará. Dejaremos que usted descubra porqué, en caso que no lo sepa todavía.

1. 

```
$ sysctl kernel.pid_max
$ cat /proc/sys/kernel/pid_max
```
2. Type:  

```
$ cat &
[1] 29222
$ kill -9 29222
```
3. 

```
$ sudo sysctl kernel.pid_max=24000
$ echo 24000 > /proc/sys/kernel/pid_max # Esto debe ser realizado como root
$ cat /proc/sys/kernel/pid_max
```
4. 

```
$ cat &
[2] 311
$ kill -9 311
```

Tenga en cuenta que cuando se comienza de nuevo, el kernel comienza en PID=300, no en un número menor. Es posible que se de cuenta que asignar PIDs a los procesos nuevos no es algo trivial; debido a que el sistema puede haber comenzado de nuevo a asignar los números correspondientes, el kernel siempre tiene que verificar que al crear esos nuevos PIDs no están en uso. . El kernel **Linux** tiene una manera muy eficiente de hacer esto, la cual no depende del número de procesos corriendo en el sistema.