

- **DEFINICIÓN:** Los procesos en Linux son programas ejecutándose en la memoria RAM, a cada uno de ellos se les asigna un número identificador, conocido como Process ID (PID) y un propietario (owner). El proceso init es el primer proceso del sistema y el padre del resto de los procesos.
- **TIPOS DE PROCESOS:**
  - Sin control de terminal o Background (Demonios, Script arranque del sistema, etc).
  - Con control de terminal o Foreground (iniciados desde una consola por un usuario).

- **DESPLIEGUE DE PROCESOS:**

- Las órdenes **ps** (process status) y **pstree** (árbol de procesos) permiten desplegar los procesos actuales basados en /proc por ejemplo:

```
$ ps
```

```
PID TTY STAT TIME COMMAND
5995  2  S   0:00  /bin/login -- myname
5999  2  S   0:00  -bash
6030  2  R   0:00  ps
```

- **Procesos en Background y foreground:**

- Los procesos en background son aquellos que se ejecutan en segundo plano, es decir no están controlados por terminal, para iniciarlos utilice el caracter & al final de la orden por ejemplo:

```
$ ./proceso&
```

```
[1] 4556*
```

- Los procesos en foreground son aquellos que se ejecutan en primer plano y son controlados por el terminal, por ejemplo con las teclas ctrl+z los detiene (stop) y ctrl+c los interrumpe (interrupt).

\* Nota: El primer numero entre [ ] es la secuencia en la cola de procesos en background y el numero siguiente es el PID.

- **Ordenes para el Control de Procesos:**

- **bg** : se usa para enviar los procesos detenidos (stop) al modo de background.
- **fg** : se usa para enviar los procesos en background o detenidos al modo de foreground.
- **jobs**: se utiliza para ver la lista de los procesos en background o detenidos.

\*Nota: puede utilizar el caracter % para referirse a un job específico en las órdenes fg ,bg.

```
$ jobs
[1]- Ejecutando           sleep 1000 &
[2]+ Ejecutando           sleep 1001 &
$ fg %2
sleep 1001
```

- **Enviando Señales a los Procesos**: utilice la orden kill para enviarle señales (signals) a los procesos

- **Sintaxis General**:

**kill -señal PID/JOB ....**

- donde señal puede ser por ejemplo:
  - \$ kill -9 2345 (Elimina el proceso 2345 inmediatamente)
  - \$ kill -SIGTERM 2101 (Termina la ejecución del proceso ).
  - \$ kill -SIGSTOP %1 (Detiene el primer trabajo en la cola de los backgrounds).
- use la opción -l para listar las señales disponibles.
- Nota la variante **killall** permite enviar un SIGTERM a un conjunto de procesos.

- **Asignando prioridades a los Procesos:** utilice las ordenes **nice** y **renice** para ajustar las prioridades asignadas los procesos antes o despues de iniciarlos respectivamente.
- **Sintaxis General:**  
**nice [+ ó -]prioridad orden**  
**renice [+ ó - ]prioridad PID**
- Nota: las prioridades pueden variar entre -20 y 19, siendo el más negativo la más alta prioridad.

- **Monitoreo de Procesos:**

- **top** : es una orden cíclica que ordena los primeros 20 procesos de acuerdo al consumo de memoria y cpu del sistema, además de permitir enviar señales y asignar prioridades.
- **free** : permite ver el uso y la disponibilidad de la Memoria física y compartida, Buffers y SWAP del sistema.
- **uptime** : permite ver en una línea la hora, tiempo de ejecución, cantidad de usuarios y el average de la carga del CPU.

Nota: la orden top incluye también la información de las órdenes free y uptime.