

8. Procesos

- Al ejecutar una orden (comando) se lanza un proceso. Por lo tanto un proceso es una orden en ejecución.
- El propio shell con el que se está trabajando es un proceso lanzado por el sistema. Es equivalente al escritorio de Windows. El escritorio es un proceso lanzado por el sistema operativo para permitir la comunicación entre el usuario y el ordenador. En Unix-Linux es lo mismo, con la diferencia de que este sistema asigna un número de identificación a cada proceso lanzado por el usuario

8. Procesos

- Una tarea puede ser ejecutada de dos modos diferentes:
 - **Foreground o modo inmediato:**
 - Procesos interactivos con el usuario en el terminal.
 - No más de un programa ejecutándose en este modo.
 - **Background o forma desentendida:**
 - Al ejecutarlos el shell devuelve el prompt inmediatamente permitiendo seguir trabajando en el terminal mientras se ejecuta.
 - Se lanza un trabajo en background escribiendo **&** al final del comando.
 - El sistema visualiza en pantalla el PID y controla su ejecución sin requerir la presencia del usuario.
 - Para procesos que son muy largos que no generan salida a pantalla.
 - Si generan salida por pantalla, sus mensajes interferirán en el trabajo que se esté realizando en primer plano (foreground).

8. Procesos

Unix-Linux asigna a cada proceso:

- PID – Identificación de proceso
- PPID – Identificación del proceso padre que lo ha ejecutado

Cualquier comando u orden, no es un proceso hasta que no se ejecuta. En ese momento, se almacena en memoria junto con su PID (Identificación de proceso) que va a ser único durante esa sesión y su PPID (Identificación del proceso padre que lo ha ejecutado) que por lo general será el Shell (sh).

8. Procesos

- **jobs**
 - Muestra una lista de las tareas que se están ejecutando en segundo plano
- **bg [%<numero>]**
 - Para interrumpir una tarea lanzada en foreground: CTRL+Z
 - Reanuda una tarea interrumpida y ponerla en segundo plano
- **fg [%<numero>]**
 - Quita una tarea del segundo plano y la trae a primer plano
- **nice [-n]orden**
 - Permite ejecutar un programa con una prioridad distinta de la normal. Incrementando la prioridad de un proceso, este se ejecutará más rápidamente, a costa de que el resto se vean perjudicados.
 - Solo el administrador del sistema puede aumentar la prioridad de un proceso
`nice -12 who`

8. Procesos

- **ps [-efl]**

- Informa acerca de los procesos que en ese momento se están ejecutando en el sistema
- Si no le pasamos ninguna opción, solo ofrece un pequeño informe de los procesos asociados a nuestro terminal
- Opciones
 - e Informa de todos los procesos que hay en el sistema.
 - f Proporciona una información más completa de cada proceso.
 - l Proporciona toda la información de cada proceso:

- **Kill [-señal] PID [PID..]**

- Sirve para enviar señales a uno o varios procesos identificados por su PID.
- Señal
 - 9 SIGKILL: Provoca la terminación de un proceso.

- **sleep segundos**

- Esta orden se utiliza para detenernos un n° determinado de segundos antes de continuar con la siguiente.

8. Procesos

- **WAIT.**
 - Permite detener un proceso ejecutándose en segundo plano mientras se está ejecutando un proceso hijo.
- **TIME.**
 - Calcula el tiempo que un proceso tarda en ejecutarse.

9. Variables de entorno

- Una **variable de entorno** simplemente es un nombre asociado a una cadena de caracteres.
- Algunas variables son útiles para no tener que escribir muchas opciones al ejecutar un programa y otras las utiliza el propio shell (PATH, PS1,etc).
- Algunos ejemplos de variables de entorno son:
 - \$USER nombre del usuario
 - \$HOME ruta del usuario por defecto
 - \$SHELL shell usado por el usuario
 - \$PATH ruta donde buscará los ejecutables
 - \$PS1 indicador de líneas de comandos
- ***Comandos relacionados con las variables de entorno***
 - **env** visualiza las variable de entorno
 - **set** Muestra variables locales a la shell y de entorno
 - **export [variable]** transporta variables de shell al área de entorno
 - **unset [variable]** elimina variables locales o asigna NULL

11. Variable de entorno PATH

- La variable de entorno PATH contiene una lista con los directorios en los que el shell busca el comando que se escribe desde el teclado.
- La búsqueda se hace en el orden en que están los directorios en la variable PATH.
- Para **ver el contenido de la variable** podemos hacer:
 `echo $PATH`
 `env | grep PATH`
- Para **añadir un directorio a la variable de entorno** PATH para la sesión activa.
 Si desea añadir por ejemplo `/home/user/mis_prog` a la variable PATH, en el shell escriba
 `PATH=$PATH:/home/user/mis_prog`