

---

# Introducción a los Procesos en GNU/Linux

# *Nota de Copyright*

---

© 2005 Diego Chaparro. Algunos derechos reservados.

Este trabajo se distribuye bajo la licencia Creative Commons Attribution-ShareAlike. Para obtener la licencia completa, véase <http://creativecommons.org/licenses/by-sa/2.1/es>

# Procesos

- ◆ El intérprete de comandos nos permite ejecutar procesos.
- ◆ Cada proceso tiene un identificador de proceso, llamado pid
- ◆ **ps**
  - ◆ El comando ps nos permite ver los procesos en ejecución
  - ◆ Opciones útiles:
    - f Procesos vistos en forma de árbol
    - a Muestra todos los procesos
    - u Muestra el usuario y la fecha de inicio
    - x Muestra los procesos asociados a un terminal

# Procesos

## ◆ top

- ◆ Es otra forma de ver los procesos en ejecución
- ◆ Además muestra información sobre memoria y memoria swap
- ◆ Tiene muchos comandos interactivos:

h	Ayuda
M	Ordenar procesos por memoria
P	Ordenar procesos por uso de CPU
I	Muestra las medias de uso

# Procesos

## ◆ nice

- ◆ Podemos especificar la prioridad con la que se va a ejecutar un determinado comando
- ◆ Sintaxis: *nice [opcion] comando*
- ◆ Por defecto la prioridad es 10. Podemos especificarle un valor entre -20 (máxima prioridad) y 19 (mínima prioridad)
- ◆ Ejemplo: *nice -15 ls -la*

# Procesos

## ◆ &

- ◆ Los comandos que ejecutamos con & se ejecutan en segundo plano

## ◆ jobs

- ◆ Los comandos o conjuntos de comandos que ejecutamos son tratados por la shell como “trabajos”, los cuales podemos suspender y volver a reanudar cuando queremos.
- ◆ Con el comando fg %, o fg % podemos traerlos a primer plano si estaban suspendidos o en segundo plano. Se puede especificar el número de trabajo (lo que aparece entre corchetes al ejecutar en background), o el nombre del comando (ej: vi, man, ...)
- ◆ Con el comando bg podemos mandar a segundo plano un proceso suspendido.
- ◆ Pulsando CTRL+z podemos suspender un programa que esté en primer plano

# Procesos

## ◆ señales

- ◆ Una señal es una comunicación que se le envía a un proceso
- ◆ Una señal puede ser un aviso de que ha ejecutado una instrucción ilegal, para indicarle que termine, ...
- ◆ El comando **kill** es usado para mandar señales a un proceso
- ◆ Sintaxis: `kill [opcion] ID`
- ◆ El ID puede ser el pid, `%num_job` o `%nombre_job`
- ◆ Opciones:
  - l Muestra todas las señales disponibles
  - 9 Manda la señal SIGKILL
- ◆ La señal por defecto es la 15 (SIGTERM)