SISTEMAS INFORMÁTICOS UD-4

SISTEMAS OPERATIVOS. GESTION DE USUARIOS Y PROCESOS

GUÍA DE COMANDOS EN SHELL PARA LA GESTIÓN DE PROCESOS

ÍNDICE

[ps (PROCESS STATUS). INFORMACIÓN DE LOS PROCESOS 3](#_Toc128754217)

[top. INFORMACIÓN EN TIEMPO REAL DE LOS PROCESOS 5](#_Toc128754218)

[fg (FOREGROUND), bg (BACKGROUND), jobs Y sleep. CONTROL DE PROCESOS EN PRIMER Y SEGUNDO PLANO 6](#_Toc128754219)

[nice Y renice. PRIORIDAD DE LAS ÓRDENES 7](#_Toc128754220)

[kill. ENVÍO DE SEÑALES 8](#_Toc128754221)

[at. AUTOMATIZACIÓN DE TAREAS EN LINUX 8](#_Toc128754222)

[cron. AUTOMATIZACIÓN PERIÓDICA DE TAREAS EN LINUX 10](#_Toc128754223)

[systemctl. CONTROL DE SERVICIOS 12](#_Toc128754224)

[uptime. TIEMPO DE USO DEL SISTEMA OPERATIVOS, USUARIOS CONECTADOS Y CARGA MEDIA 14](#_Toc128754225)

[free. UTILIZACIÓN DE LA MEMORIA 14](#_Toc128754226)

[vmstat. INFORMACIÓN DE LA CPU, LA MEMORIA, EL USO DE DISCOS… 14](#_Toc128754227)

[df. USO DEL ESPACIO EN DISCO 15](#_Toc128754228)

[w. INFORMACIÓN EXTENDIDA DE USUARIOS CONECTADOS 15](#_Toc128754229)

# ps (PROCESS STATUS). INFORMACIÓN DE LOS PROCESOS

El comando ps nos permite obtener información de los procesos del sistema, tiene la sintaxis: ps [modificaciones]

Texto

Descripción generada automáticamente

Las modificaciones son:

* aux 🡪 Sirve para ver información extendida de los procesos.

Imagen que contiene Calendario

Descripción generada automáticamente

* + UID del usuario propietario del proceso (**USER**).
  + **PID** del proceso.
  + PPID del proceso.
  + Porcentaje de uso de CPU (**%CPU**).
  + Porcentaje de uso de memoria (**%MEM**)
  + Tamaño del proceso en la memoria virtual en KB (**VSZ**)
  + Tamaño de la memoria residente del proceso (en memoria física) en KB (**RSS**)
  + Terminal de lanzamiento (**TTY**)
  + Estado del proceso (**STAT** o **S**)

Tabla

Descripción generada automáticamente

* + Tiempo de inicio del proceso (**Start** Time o STIME)
  + Tiempo de CPU consumido (**TIME**).
  + Orden de ejecución (**COMMAND**).

Se puede asociar a la opción - -sort cputime para ordenar por tiempo consumido de CPU.

Imagen de la pantalla de un computador

Descripción generada automáticamente con confianza baja

* -u 🡪 Sirve para ver los procesos de un usuario en concreto.

Texto

Descripción generada automáticamente

* - -pid 🡪 Sirve para ver los datos de un proceso en concreto.

Texto

Descripción generada automáticamente

* -t 🡪 Sirve para ver los procesos de una terminar virtual.

Texto

Descripción generada automáticamente

* -l 🡪 Listado de procesos.

Se suele filtrar el resultado de ps concatenando el comando grep:



Si quisiéramos ver los procesos en forma de árbol usaríamos el comando pstree.

Texto

Descripción generada automáticamente

# top. INFORMACIÓN EN TIEMPO REAL DE LOS PROCESOS

El comando top nos permite visualizar en tiempo real los procesos, tiene la sintaxis top [opciones].

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Las opciones del comando son:

* -d 🡪 Número de segundos de actualización.
* -u 🡪 Usuario.

Imagen que contiene Texto

Descripción generada automáticamente

* -p 🡪 PID del proceso.

Se puede concatenar para mostrar sólo un número determinado de procesos, por ejemplo, si sólo queremos ver los 3 primeros procesos:

Calendario

Descripción generada automáticamente

En la cabecera del resultado nos aparecen una serie de líneas con información importante:

Texto

Descripción generada automáticamente

* **Primera línea.** Indica la hora actual, el tiempo que lleva el sistema encendido, el número de usuarios conectados y la carga media en intervalos de 1, 5 y 15 minutos.
* **Segunda línea.** Indica el número de tareas (procesos) y su estado: ejecución, hibernación, detenidos y zombie.
* **Tercera línea.** Indica el porcentaje de uso (en segundos -tiempo de uso-) de CPU por parte de usuario, sistema, etc.
* **Cuarta línea.** Indica el tamaño de memoria (total, libre, usada y utilizada por búfer).
* **Quinta línea.** Indica el tamaño de memoria virtual (total, libre, usada y disponible).

# fg (FOREGROUND), bg (BACKGROUND), jobs Y sleep. CONTROL DE PROCESOS EN PRIMER Y SEGUNDO PLANO

Los procesos se pueden ejecutar en primer plano (foreground) o en segundo plano (background).

Si se lanza un proceso en primer plano desde la terminal, perderemos el control de la terminal, para evitar esto hay que añadir un & al final del comando.

Texto

Descripción generada automáticamente

Con el comando jobs podemos ver las tareas que se están ejecutando en segundo plano.

Texto

Descripción generada automáticamente

Con el comando fg podemos pasar una tarea de segundo plano a primer plano, con el comando bg podemos pasar una tarea de primer plano a segundo plano.

La sintaxis es la siguiente:

* fg [%][tarea]



* bg [%][tarea]



Si no se indican argumentos se pasará a primer/segundo plano la tarea más reciente.

Si queremos pasar una tarea de primer a segundo plano primero se debe detener con Ctrl + Z y a continuación lanzarla con bg.

Texto

Descripción generada automáticamente

A la derecha del número de cola de procesos aparece un + y un -, el + indica la tarea más reciente y el - la segunda más reciente.

Con el comando sleep podemos bloquear la ejecución el tiempo que indiquemos, tiene la sintaxis sleep [segundos] al igual que cualquier otro proceso se puede pasar a segundo plano.

Texto

Descripción generada automáticamente

# nice Y renice. PRIORIDAD DE LAS ÓRDENES

El algoritmo de planificación de Linux que determina el orden de ejecución entre los procesos en cola de listos emplea una mezcla de algoritmos como RR, FIFO, etc.

La prioridad real de cada proceso se puede alterar mediante el comando nice que puede tomar un valor desde -20 (máxima propiedad) y 19 (mínima prioridad).

El usuario propietario del proceso podrá disminuir su propiedad, no aumentarla. Sólo el superusuario (root) podrá aumentar la prioridad.

Por defecto los procesos se lanzan con prioridad 0.

La sintaxis del comando es: nice [-n] {+|-} prioridad orden.



Texto

Descripción generada automáticamente

Para modificar la prioridad del proceso se utiliza el comando renice que tiene la sintaxis: renice [-n] prioridad [-p PID’s] [-u usuarios].

Texto

Descripción generada automáticamente con confianza media

# kill. ENVÍO DE SEÑALES

Con el comando kill podemos enviar una señal, la sintaxis es kill -señal PID.

Las señales más utilizadas son:

* **2** o SIGINT 🡪 Interrumpe la ejecución de un proceso.
* **9** o SIGKILL 🡪 Finaliza un proceso.
* **15** o SIGTERM 🡪 Similar al anterior (en este caso la orden podría no ejecutarse si el programa está bloqueado).
* **18** o SIGCONT 🡪 Reanuda la ejecución de un proceso que estaba detenida.
* **19** o SIGSTOP 🡪 Para la ejecución de un proceso.

# at. AUTOMATIZACIÓN DE TAREAS EN LINUX

Con el comando at podemos conseguir que se automatice una tarea (una única vez) para que se ejecute en un determinado momento.

Para poder utilizar el comando hay que instalarlo previamente con sudo apt install at.

La sintaxis del comando es at [hora]. Al introducir el comando podremos introducir la secuencia de acciones que queremos que se ejecuten. Para guardarlo pulsamos Ctrl + D.

Las opciones que se permiten son:

* Para que se ejecute a una hora exacta 🡪 at [hora]. Por ejemplo:
  + at 05:00PM
  + at 03:00AM
* Para que se ejecute a una hora exacta de un día determinado (el siguiente que cumpla la condición) 🡪 at [hora] [día]. Por ejemplo:
  + at 07:00AM Sun
* Para que se ejecute a una hora de un mes y día exactos 🡪 at [hora] [mes] [día]. Por ejemplo:
  + at 11:00AM April 14
* Para que se ejecute a una hora de una fecha exacta 🡪 at [hora] [fecha]. Por ejemplo:
  + at 10:00AM 6/22/2021 o at 10:00AM 6.22.2021
* Para que se ejecute a una hora del mismo día del siguiente mes 🡪 at [hora] next month. Por ejemplo:
  + at 11:00AM next month
* Para que se ejecute a una hora del siguiente día 🡪 at [hora] tomorrow. Por ejemplo:
  + at 11:00AM tomorrow
* Para que se ejecute n horas después de que se introduzca 🡪 at now + [n] hour. Por ejemplo:
  + at now + 1 hour
* Para que se ejecute n minutos después de que se introduzca 🡪 at now + [n] minute. Por ejemplo:
  + at now + 30 minute
* Para que se ejecute n semanas después de que se introduzca 🡪 at now + [n] week.
  + at now + 1 week
* Para que se ejecute n años después de que se introduzca 🡪 at now + [n] year.
  + at now + 1 year
* Para que se ejecute a medianoche 🡪 at midnight.

Texto

Descripción generada automáticamente

Si queremos ver el listado de trabajos podemos usar el comando at con la opción -l.

Texto

Descripción generada automáticamente

Además, este comando tiene otras dos opciones:

* -d 🡪 Para eliminar una tarea.
* -c 🡪 Para una visión detallada de las subtareas.

En ambos casos el comando será at [opción] [númeroTarea].

# cron. AUTOMATIZACIÓN PERIÓDICA DE TAREAS EN LINUX

En caso de que queramos automatizar una tarea para que se repita cada hora, cada semana, cada mes… deberemos utilizar cron que es un servicio o demonio (programa que se ejecuta en 2º plano) que permite lanzar órdenes o scripts que hayan sido definidos previamente en el archivo de texto crontab.

Va a existir un crontab del sistema (accesible por ejemplo mediante nano /etc/crontab):

Texto

Descripción generada automáticamente con confianza baja

Y un crontab para cada usuario, accesible mediante el comando crontab -e:

Texto

Descripción generada automáticamente

Para comprobar la instalación de cron en nuestro equipo debemos usar la instrucción dpkg -l cron:

Texto

Descripción generada automáticamente

También podemos verificar el estado del servicio con los diferentes comandos de systemctl.

El archivo crontab del sistema se compone de:

Texto

Descripción generada automáticamente con confianza media

* Primera columna: Minuto (0 a 59).
* Segunda columna: Hora (0 a 23).
* Tercera columna: Día del mes (1 a 31).
* Cuarta columna: Mes (1 a 12 o las 3 iniciales en inglés: jan, feb, mar, apr, may…).
* Quinta columna: Día de la semana (0 – 6, aunque el domingo puede ser 0 ó 7, también se pueden usar las 3 iniciales de los días en inglés: sun, mon, tue, wed…).
* Sexta columna: Usuario.
* Séptima columna: Comando a ejecutar.

El crontab de usuario es muy similar pero sin la columna usuario ya que las órdenes de este crontab sólo pueden ser escritas por el propio usuario.

Por ejemplo, vamos a crear un script simple:

Texto

Descripción generada automáticamente

Y vamos a hacer que se ejecute cada día a las 19:00:

Texto

Descripción generada automáticamente

Hay que tener en cuenta que cron lee el archivo /etc/crontab cada minuto.

A diferencia de las tareas creadas con at (que de encontrarse el sistema apagado no se ejecutarán nunca ya que se programan para un momento específico) las tareas creadas en crontab se ejecutarán la siguiente vez que se inicie el ordenador en caso de que este se encuentre apagado en el momento en que deberían ejecutarse.

Existen algunas cadenas de texto reservadas para ejecutar procesos durante determinados periodos en el propio Crontab, esto nos será de mucha ayuda:

* @reboot: Ejecuta una vez y nada más iniciarse el equipo.
* @yearly: ejecuta sólo una vez al año: 0 0 1 1 \*
* @monthly: ejecuta una vez al mes y el primer día: 0 0 1 \* \*
* @weekly: todas las semanas, el primer minuto de la primera hora de la semana: 0 0 \* \* 0.
* @daily: todos los días a las 12 de la noche: 0 0 \* \* \*
* @midnight: Tiene el mismo efecto que el anterior.
* @hourly: todas las horas durante su primer minuto: 0 \* \* \* \*

# systemctl. CONTROL DE SERVICIOS

En Linux (salvo Gentoo) existe un servicio llamado systemd o init. El cual es el primer servicio que inicia el Kernel de Linux y del cual penden los demás servicios.

Con systemctl vamos a poder conseguir realizar acciones con los servicios, por ejemplo:

* systemctl help [nombreServicio] 🡪 Permite acceder a información del servicio, similar a man con los comandos.

Texto

Descripción generada automáticamente

* systemctl list-dependencies [nombreServicio] 🡪 Nos muestra el árbol de dependencias de un servicio que se tienen que ejecutar para que ese servicio pueda funcionar.

Texto

Descripción generada automáticamente

* systemctl start [nombreServicio] 🡪 Inicia un servicio. Este comando requiere identificación como sudo.



* systemctl stop [nombreServicio] 🡪 Detiene un servicio. Este comando requiere identificación como sudo.



* systemctl restart [nombreServicio] 🡪 Reinicia un servicio. Este comando requiere identificación como sudo.



* systemctl reload [nombreServicio] 🡪 Recarga la configuración de un servicio sin llegar a reiniciarlo.



* systemctl status [nombreServicio] 🡪 Muestra el estado del servicio.

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

* systemctl enable [nombreServicio] 🡪 Habilita permanentemente un servicio en el arranque. Requiere identificación como sudo.



* systemctl disable [nombreServicio] 🡪 Deshabilita permanentemente un servicio en el arranque. Requiere identificación como sudo.



# uptime. TIEMPO DE USO DEL SISTEMA OPERATIVOS, USUARIOS CONECTADOS Y CARGA MEDIA

El comando uptime muestra cuanto tiempo ha estado funcionando el SO, cuántos usuarios están conectados actualmente en el sistema y la carga media en tiempos de 1, 5 y 15 minutos.

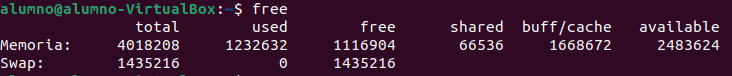
Es igual que la primera línea de top.



# free. UTILIZACIÓN DE LA MEMORIA

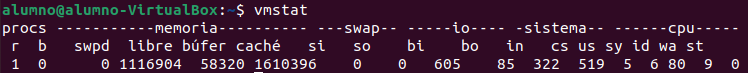
El comando free muestra la cantidad de memoria libre, la cantidad de memoria utilizada y la cantidad de memoria disponible en el sistema.

Es similar a la cuarta línea de top.



# vmstat. INFORMACIÓN DE LA CPU, LA MEMORIA, EL USO DE DISCOS…

Proporciona información sobre el uso de la CPU, la memoria, el intercambio de memoria y el uso de discos. También muestra estadísticas en tiempo real de la cantidad de operaciones de entrada/salida por segundo que se realizan en los dispositivos de almacenamiento.



# df. USO DEL ESPACIO EN DISCO

El comando df muestra el uso del espacio en disco para cada partición en el sistema.

Texto

Descripción generada automáticamente

# w. INFORMACIÓN EXTENDIDA DE USUARIOS CONECTADOS

El comando w muestra información sobre los usuarios que están conectados actualmente al sistema, incluyendo su tiempo de inicio de sesión, su carga de trabajo y el tiempo que han estado inactivos.

