Sergio Cobo García

SISTEMAS INFORMÁTICOS

U3 ADMINISTRACIÓN DE SOFTWARE BASE

ACTIVIDAD PRÁCTICA 6: GESTIÓN DE PROCESOS

Tabla de contenido

[Procesos en Windows 2](#_Toc123946875)

[Procesos en Unix/Linux 6](#_Toc123946876)

Procesos en Windows

Forma

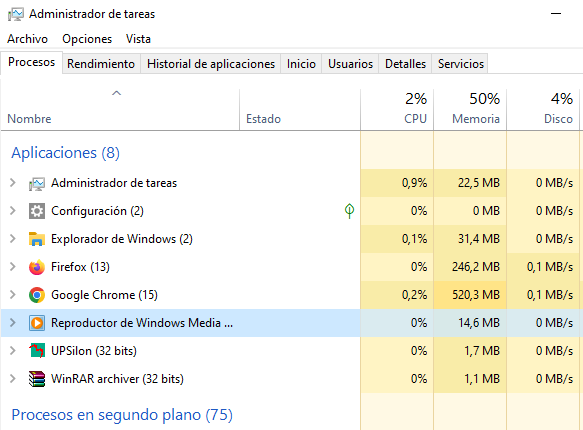
|  |  |
| --- | --- |
| 01 | El administrador de tareas tiene diferentes puntos de entrada. Explicad todas las maneras posibles para acceder al administrador de tareas. |

Las formas que hay para abrir el administrador de tareas son:

* Ctrl + Alt + Supr y seleccionamos Administrador de tareas.
* Click derecho encima de la barra de tareas y clicar en Administrador de tareas.
* Entrar en inicio y buscar Administrador de tareas.
* En la barra de búsqueda escribirnos *taskmgr*.

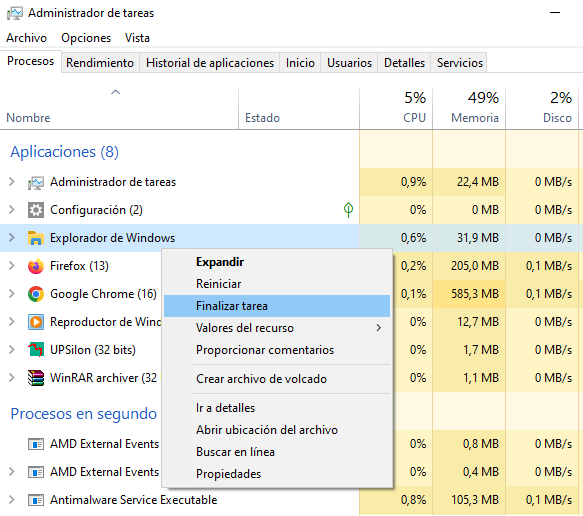
|  |  |
| --- | --- |
| 02 | Ejecutad un programa, como por ejemplo el Windows Media Player, e indicad el nombre del ejecutable, que consumo de CPU y que uso de memoria tiene. |

El Windows Media Player hace un consumo del 0% de la CPU y hace un uso de 14,6 MB de la memoria RAM.



|  |  |
| --- | --- |
| 03 | ¿Como se elimina un proceso determinado? Eliminad el proceso **explorer** y mirad que pasa. ¿Como podemos solucionar esta situación? |

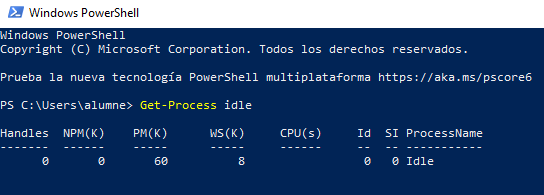
Para eliminar un proceso en el Administrador de tareas clicamos encima del proceso a eliminar y damos en Finalizar tarea.



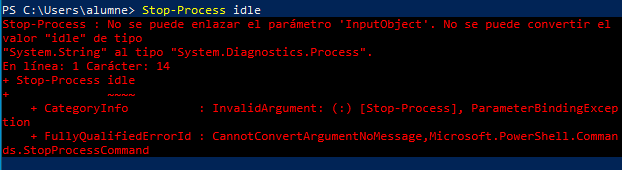
|  |  |
| --- | --- |
| 04 | ¿Cuál es el proceso IDLE en Windows? ¿Cuál es su comportamiento?  ¿Como podemos mostrar, mediante el comando Get-Process el proceso IDLE?  Intentad eliminarlo, mediante el comando **Stop-Process**. ¿Qué pasa? |

El proceso IDLE es un proceso especial ya que es el que se ejecuta cuando en el sistema no hay ningún otro proceso ejecutándose, sirve para representar el tiempo en el que en el sistema no hay ninguna tarea realizándose.

Para mostrar el proceso IDLE ejecutamos *Get-Process idle*.

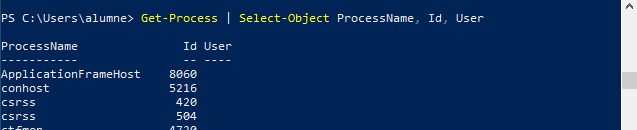


Al intentar detener el proceso IDLE con *Stop-Process idle* nos mandará un mensaje de error, ya que no está permitida la detención ni eliminación del mismo al ser necesario para el correcto funcionamiento del sistema.

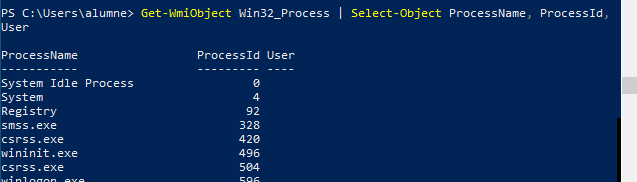


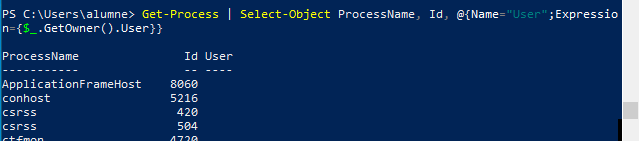
|  |  |
| --- | --- |
| 05 | ¿Como podemos mostrar, mediante el comando **Get-Process**, el usuario que ha ejecutado cada proceso y el pid de cada uno de los procesos? |

El comando a usar sería el siguiente, aunque he probado otros comandos y me ha sido imposible mostrar el usuario que ha ejecutado cada proceso.

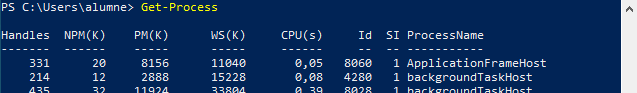


He probado los siguientes comandos:





|  |  |
| --- | --- |
| 06 | Indicad el significado de las columnas cuando lanzamos el comando **Get-Process**. |

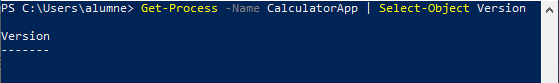


Las columnas son:

* Handles: Se utilizan para acceder a archivos, procesos… Cada vez que un proceso necesita acceder a un objeto, utiliza un handle para hacerlo.
* NPM: Muestra el número de objetos de memoria privada que está utilizando cada proceso. La memoria privada es la memoria asignada a un único proceso.
* PM: Sirve para saber la cantidad de memoria privada que están utilizando cada uno de los procesos.
* WS: Muestra la cantidad de memoria privada y compartida que está utilizando el proceso.
* CPU: Se refiere a la cantidad de tiempo que cada proceso ha estado ejecutándose en el procesador.
* Id: Es el identificador único de cada proceso en ejecución del sistema, en otras palabras, el PID.
* SI: Es la suma de Identificadores. Muestra la cantidad de recursos del sistema que están siendo utilizados por un proceso.
* ProcessName: Se refiere al nombre del proceso.

|  |  |
| --- | --- |
| 07 | ¿Como podemos enseñar, mediante el comando **Get-Process**, la versión del proceso que estemos ejecutando? |

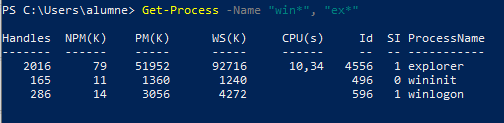
Podemos enseñar la versión de un proceso usando el siguiente comando.



En algunos casos como en el del ejemplo anterior no nos mostrará la versión del proceso. Puede darse el caso de que el proceso forme parte del mismo sistema operativo y no tenga una versión específica, como es el caso de la calculadora.

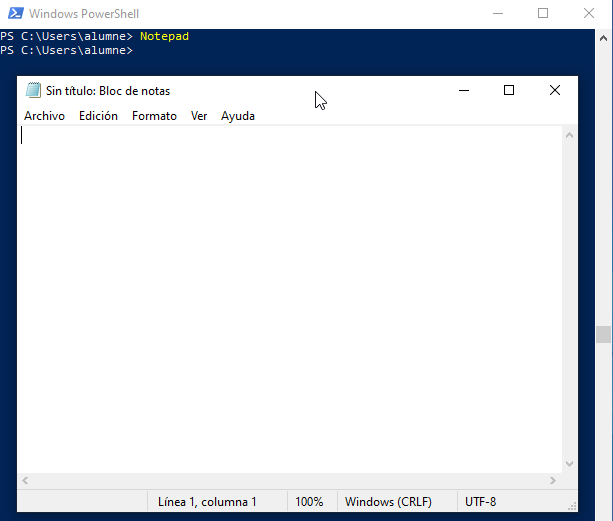
|  |  |
| --- | --- |
| 08 | ¿Como podemos mostrar, mediante el comando **Get-process**, todos los procesos que empiece su nombre con la cadena ‘win’ y ‘ex’? |

Podemos hacer la búsqueda de los procesos que empiecen por ‘win’ o ‘ex’ con el siguiente comando.

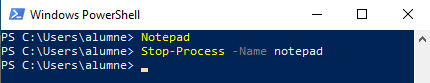


|  |  |
| --- | --- |
| 09 | Mediante la consola PowerShell, lanzad el programa ‘Notepad’.  A continuación, paradlo, también desde el PowerShell mediante el PID del programa. |

Lanzamos el programa insertando *Notepad* o bien *notepad.exe*.



Para detener el Notepad usamos el comando *Stop-Process -Name notepad*.



Procesos en Unix/Linux

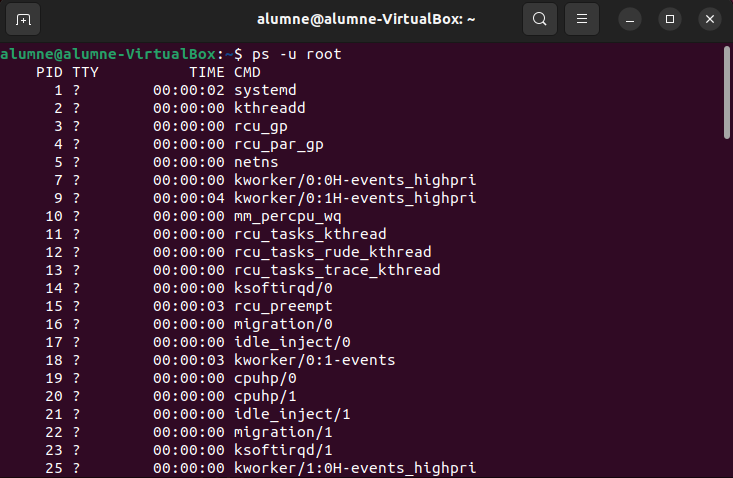
|  |  |
| --- | --- |
| 10 | Indicad que hace cada uno de los siguientes modificadores del comando **ps**:  -l, -f, -u, -p, -ef, -e, -C, -o, -aux, -eo |

Modificadores:

* -l: Lista el formato largo de la información de los procesos
* -f: Muestra la información completa de los procesos.
* -u: Muestra los procesos de un usuario concreto.
* -p: Permite especificar uno o más IDs de procesos para mostrar su información.
* -ef: Muestra la información completa de todos los procesos que se están ejecutando en el sistema.
* -e: Permite visualizar la información de todos los procesos del sistema.
* -C: Muestra la información sobre todos los procesos que están ejecutando un programa concreto.
* -o: Permite especificar los campos de la información de los procesos que el usuario desea que le muestre.
* -aux: Muestra información detallada de cada proceso.
* -eo: Tiene el mismo propósito que la opción -o pero además muestra la información solicitada sobre todos los procesos en el sistema.

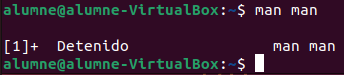
|  |  |
| --- | --- |
| 11 | Escribid un comando que liste todos los procesos el usuario root. |

El comando que lo realiza es el *ps –u root*.

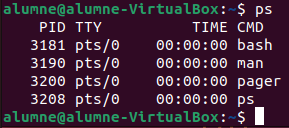


|  |  |
| --- | --- |
| 12 | Entrad en el manual de la máquina y pulsad Ctrl+Z. ¿Qué ha pasado? Haced lo necesario para continuar la consulta del manual e indicadlo. |

Con Ctrl+Z se suspende el proceso actual.



Con *ps* muestro los procesos que se están ejecutando.



Con *fg* se reanuda la ejecución del proceso suspendido y lo trae al primer plano. Si hay más de un proceso suspendido usamos *fg %n*, donde n es el número del proceso que quieres reanudar.



|  |  |
| --- | --- |
| 13 | Iniciad el proceso **man find** y suspendedlo con Ctrl+Z.  Ejecutad **xclock** en background. Utilizad **jobs** para listar los procesos en segundo plano y los procesos parados.  Utilizad el comando **fg** para mandar el man find a primer plano. Salid normalmente con q.  Utilizad fg para mandar xclock al foreground y acabadlo con Ctrl+C.  Ejecutad xclock otra vez, pero esta vez en primer plano (no podréis usar la shell). Suspendedlo con Ctrl+Z y describid que pasa. |

Ejecutamos *man find* y hacemos Ctrl+Z para suspenderlo.



Iniciamos el xclock en segundo plano poniendo *xclock &*.



El comando *jobs* se utiliza para listar los procesos tanto en segundo plano como parados.



La orden *jobs -s* lista los procesos que se encuentran detenidos.



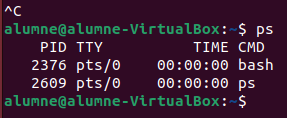
Para mandar el man find a primer plano ejecutamos *fg man find*.



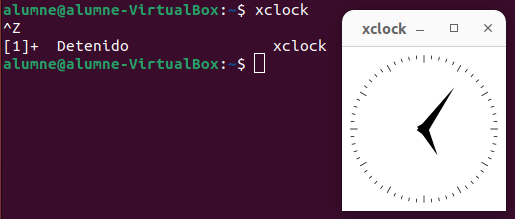
Mando el xclock a primer plano con *fg xclock*.



Seguidamente lo he parado con Ctrl+C y al hacer ps ya no aparece el proceso.

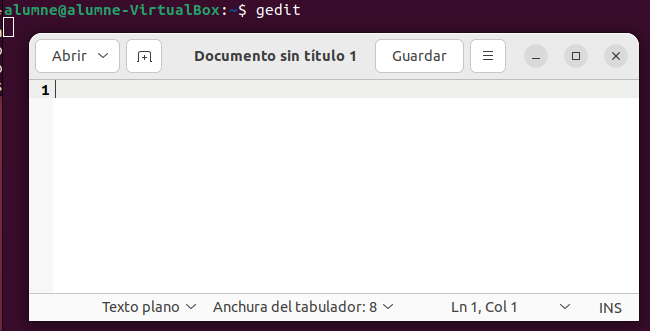


Vuelvo a abrir *xclock* y lo suspendo con Ctrl+Z. Me ha dejado suspenderlo sin problema.

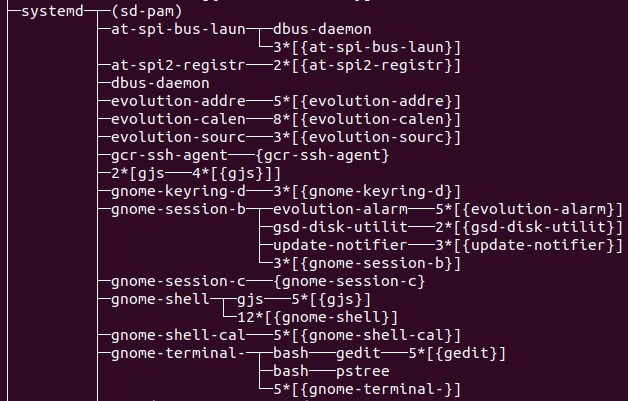


|  |  |
| --- | --- |
| 14 | Abrid un terminal y ejecutad el comando **gedit**. Desde otra terminal ejecutad **pstree** y mirad quien es el proceso padre de gedit. Matad el padre de gedit. ¿Qué pasa? Volved a hacer lo mismo, pero en este caso ejecutad gedit en segundo plano. Matad el padre. ¿Qué pasa ahora? |

Ejecuto el comando *gedit* para abrir un nuevo documento.



Lanzo el comando *pstree*, en él se muestra que el proceso padre de gedit es systemd.



Para matar el proceso padre del proceso gedit introducimos *kill systemd*. No nos permite eliminarlo porque systemd es un proceso esencial para el sistema operativo para que el sistema funcione correctamente.

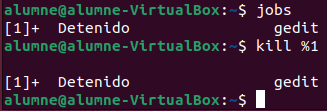


Si lo volvemos a intentar a hacer, pero esta vez iniciando gedit en segundo plano seguirá sin dejarnos, ya que como lo he dicho anteriormente el proceso systemd es esencial para el sistema.



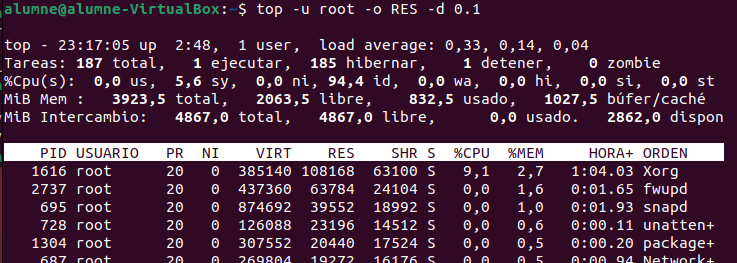
|  |  |
| --- | --- |
| 15 | ¿Como podríamos matar un proceso a partir de su número de tarea/job? |

Para matar un proceso con su job usamos el comando *kill %NumeroJob* (por ejemplo, kill %1).



|  |  |
| --- | --- |
| 16 | Aplicad los parámetros necesarios, de forma interactiva, al comando top para que se muestren únicamente los procesos del usuario root, los ordene descendentemente por ocupación de memoria y actualice los datos cada 0.1 segundos. |

El comando necesario para realizar esa búsqueda es *top -u root -o RES -d 0.1*.



|  |  |
| --- | --- |
| 17 | ¿Qué prioridad asigna el comando **nice** cuando no indicamos este valor de prioridad? Cread un proceso con la máxima prioridad posible y uno con la mínima posible utilizando este comando. ¿Como podemos consultar la prioridad de este proceso? |

La prioridad por defecto es 0.

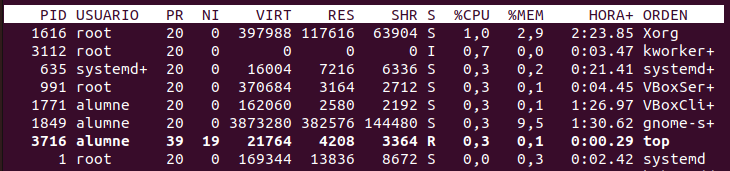
He creado un proceso con *nice -n -20 top*, pero al haberme asignado una prioridad de 0 en vez de -20 he hecho un *renice* para cambiar la prioridad de 0 a -20.





He creado un nuevo proceso con la prioridad 19, en este caso me ha asignado la prioridad que le he puesto.

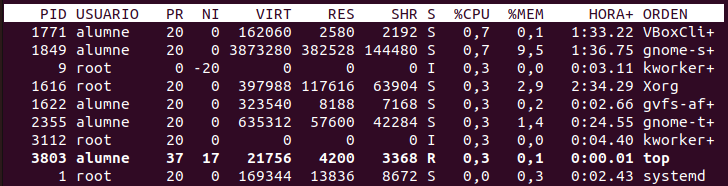




|  |  |
| --- | --- |
| 18 | Cread un proceso con una prioridad muy baja y modificadla utilizando el comando renice. ¿Como haríais para modificar la prioridad de todos vuestros procesos? ¿Qué opción del comando top os permite llevar a cabo esta misma acción? |

Creo un proceso con prioridad 17.



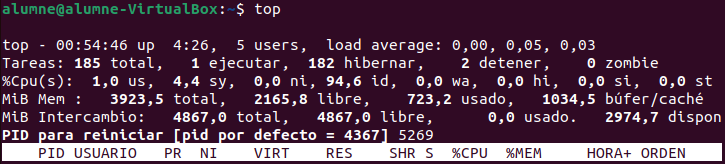


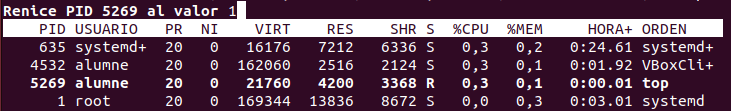
Utilizo *renice -n +10 PID* para modificar la prioridad de 17 a 10.

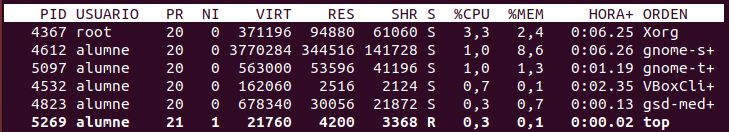


Para entrar en el modo interactivo introducimos el comando *top*.

Dentro del comando top solo está permitido cambiar la prioridad de un único proceso a la vez pulsando *r* e introduciendo el PID.





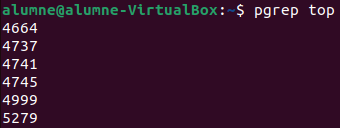


|  |  |
| --- | --- |
| 19 | ¿Cuál es la función de los comandos pidof, pgrep y pkill? |

El comando *pidof* se usa para obtener el PID de un proceso en ejecución en el sistema.

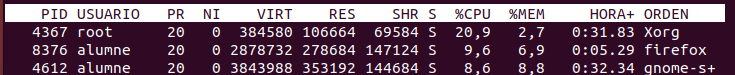


El comando *pgrep* sirve para buscar procesos que coincidan con un patrón determinado y mostrar su número de proceso.



El comando *pkill* se usa para matar procesos en un sistema Linux.

Ejecutamos Firefox por ejemplo manualmente y si introducimos en la terminal el comando top veremos que Firefox ya se está ejecutando.



El comando *pkill firefox* lo elimina.

