# 

|  |
| --- |
|  |
| Fundamentos Hardware |
|  |
| *UD04*  *Gestores de paquetes y Procesos* |

Fundamentos Hardware

Contenido

[1. Definición 3](#_Toc412132848)

[Gestión 11.3.1 Paquetes Debian 3](#_Toc412132849)

[11.3.1.1 Debian - Adición de paquetes 4](#_Toc412132850)

[11.3.1.2 Debian - Actualización de paquetes 4](#_Toc412132851)

[11.3.1.3 Debian - eliminar paquetes 5](#_Toc412132852)

[11.3.1.4 Debian - Consulta de Paquetes 6](#_Toc412132853)

[11.3.2 Administración de paquetes RPM 6](#_Toc412132854)

[11.3.2.1 RPM - Adición de paquetes 7](#_Toc412132855)

[11.3.2.3 RPM - Eliminación de paquetes 8](#_Toc412132856)

[11.3.2.4 RPM - Consulta de Paquetes 8](#_Toc412132857)

[11.4 Linux Kernel 8](#_Toc412132858)

[11.5 Jerarquía de Proceso 11](#_Toc412132859)

[11.6 Comando ps (Proceso) 12](#_Toc412132860)

[11.8 Comando free 13](#_Toc412132861)

# 1. Definición

Un aspecto de esta tarea es la instalación, desinstalación y actualización de paquetes de software.

Otro aspecto de la administración del software consiste en la gestión de los programas cuando ya están en ejecución, que a menudo se conocen como procesos (procesos).

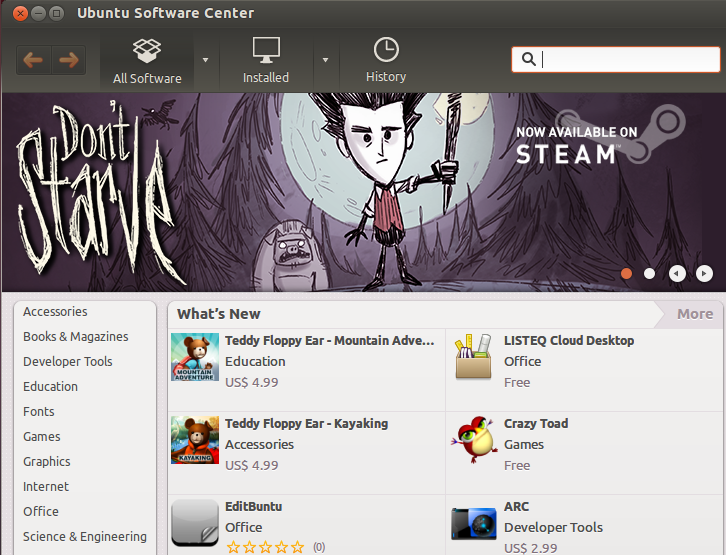
Por último, este capítulo se encarga de los archivos de registro, archivos de log o simplemente logs. En estos archivos se registran todos los detalles que suceden durante la ejecución de los programas, en particular de los programas que se ejecutan de forma automática y en segundo plano.

## 1.1 Gestión Paquetes Debian

La distribución Debian y sus derivados como Ubuntu y Mint, utilizan el sistema de gestión de paquetes de Debian.

Estos paquetes de software se distribuyen como archivos que terminan en ".deb".

La herramienta de nivel más bajo para el manejo de estos archivos es el comando dpkg . Este comando puede ser complicado para los usuarios novatos de Linux, por lo que el Advanced Package Tool, apt-get , un programa de front-end para  dpkg , hace que la gestión de paquetes sea más fácil. Hay otras herramientas de línea de comandos que sirven como front-ends para dpkg , como aptitud , así como GUI front-end como sináptica y software-center (que se muestra a continuación).



## 11.3.1.1 Debian - Adición de paquetes

Los repositorios de Debian contienen más de 65.000 paquetes de software diferentes. Para obtener una lista actualizada de estos repositorios de Internet, puede sudo apt-get update.

Para buscar palabras clave dentro de estos paquetes, puede usar el sudo apt-cache search palabra\_clave .

Una vez que haya encontrado el paquete que desea instalar, se puede instalar con sudo  apt-get install paquete.

Para ejecutar estos comandos, el sistema debe tener acceso a Internet. El apt-cache  busca en repositorios de Internet.

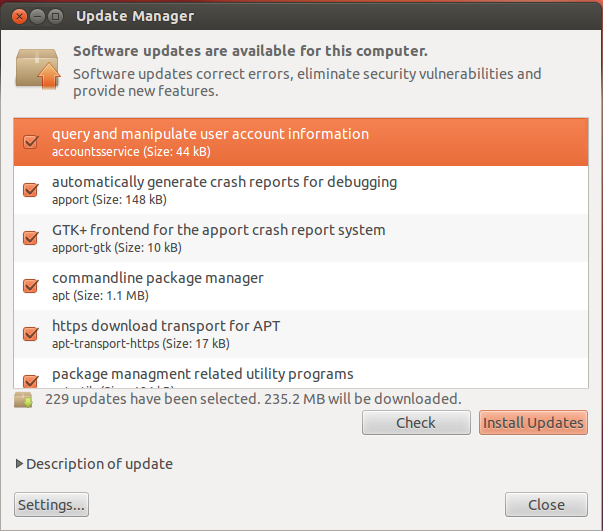
## 11.3.1.2 Debian - Actualización de paquetes

Si desea actualizar un paquete individual, se ejecuta el comando: sudo apt-get install paquete

Si una versión anterior del paquete ya está instalada, será actualizado. De lo contrario, se llevará a cabo una nueva instalación.

Si desea actualizar todos los paquetes posibles, sudo apt-get upgrade.

Los usuarios que inician sesión en una interfaz gráfica puede tener un mensaje en el área de notificación de la update-manager que indica que hay actualizaciones disponibles, como se muestra a continuación:



## 11.3.1.3 Debian - eliminar paquetes

Tenga en cuenta que la eliminación de un paquete de software puede resultar en la eliminación de otros paquetes. Debido a las dependencias entre los paquetes, si elimina un paquete, entonces todos los paquetes que necesitan, o que dependen de ese paquete serán eliminados también.

Si desea eliminar todos los archivos de un paquete de software, a excepción de los archivos de configuración, a continuación, puede ejecutar el sudo apt-get remove paquete de comandos.

Si desea eliminar todos los archivos de un paquete de software, incluidos los archivos de configuración, a continuación, puede ejecutar el sudo apt-get --purge paquete de comandos.

Es posible que desee guardar los archivos de configuración en el caso de que usted planea volver a instalar el paquete de software en un momento posterior.

## 11.3.1.4 Debian - Consulta de Paquetes

Hay varios tipos diferentes de preguntas que los administradores necesitan utilizar

 Para obtener una lista de todos los paquetes que están instalados actualmente en el sistema, ejecute el dpkg  -l.

Para una lista de los archivos que componen un paquete particular, puede ejecutar dpkg -L paquete .

Para consultar un paquete de información, o de su estado, utilice el dpkg -s paquete.

Para determinar si un archivo en particular fue puesto en el sistema de archivos como el resultado de la instalación de un paquete, utilice el  dpkg -S / path / archivo /. Si el archivo fue parte de un paquete, se proporcionará el nombre del paquete. Por ejemplo:

**sysadmin @ localhost: ~ $** dpkg -S /usr/bin/who

coreutils: /usr/bin/who

El ejemplo anterior muestra el archivo /usr/bin/who es parte del paquete coreutils .

## 11.3.2 Administración de paquetes RPM

La **Base Normas Linux** , que es un proyecto de la Fundación Linux, está diseñado para especificar (a través de un consenso) un conjunto de normas que aumentan la compatibilidad entre conformando sistemas Linux. De acuerdo con la Base de Normas Linux, el sistema de gestión de paquetes estándar es de RPM.

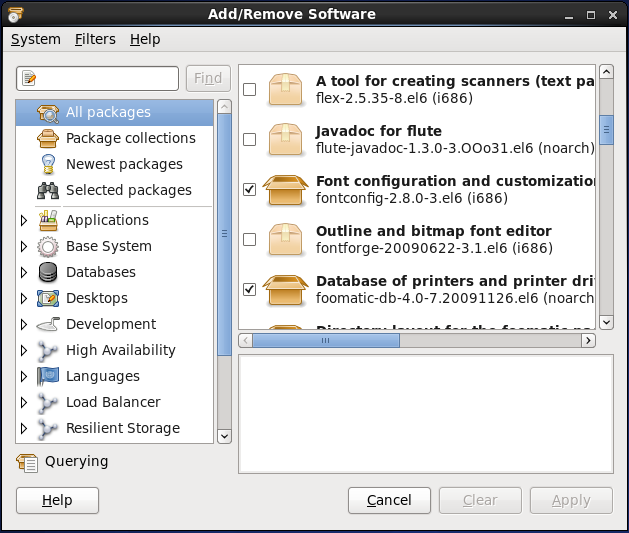
RPM hace uso de un archivo ".rpm" para cada paquete de software. Este sistema deriva de las distribuciones Hat-Red (como Red Hat, CentOS, Fedora ) para administrar el software. Además, otras distribuciones que no son derivados Red Hat (tales como SUSE, OpenSUSE y Mandriva) también utilizan RPM.

Comandos RPM no están disponibles en el entorno de máquina virtual de este curso.

Al igual que el sistema Debian, sistemas de gestión de la pista del paquete RPM dependencias entre paquetes. Seguimiento de dependencias garantiza que al instalar un paquete, el sistema también instalará los paquetes necesarios para ese paquete para funcionar correctamente. Dependencias también aseguran que las actualizaciones y eliminaciones de software se realizan correctamente.

La herramienta de back-end más utilizado para administración de paquetes RPM es el comando  rpm. Mientras el rpm  puede instalar, actualizar, consultar y eliminar paquetes, la línea de comandos herramientas front end como yum y up2date automatizan el proceso de resolución de problemas de dependencia.

Además, hay herramientas de front-end basado en GUI como yumex y gpk-aplicación (que se muestra a continuación) que también hacen que la gestión de paquetes RPM más fácil.



Debe tener en cuenta que la mayoría de los comandos siguientes requerirán privilegios de root. La regla general es que si un comando afecta el estado de un paquete, necesitará tener acceso administrativo. En otras palabras, un usuario normal puede realizar una consulta o una búsqueda, pero para agregar, actualizar o eliminar un paquete requiere que el comando se ejecute como usuario root.

## 11.3.2.1 RPM - Adición de paquetes

Para buscar un paquete de los repositorios configurados, ejecute el yum search palabras clave .

Para instalar un paquete, junto con sus dependencias, ejecute el yum install paquete .

Comandos RPM no están disponibles en el entorno de máquina virtual de este curso.

## 11.3.2.3 RPM - Eliminación de paquetes

Cualquier sistema de gestión de paquetes realiza un seguimiento de las dependencias, si desea eliminar un paquete, entonces usted puede terminar la eliminación de más de uno, debido a las dependencias. La forma más fácil de resolver automáticamente los problemas de dependencia es utilizar un yum comando:

yum remove paquete

Utilizando rpm , no eliminará las dependencia de forma automática.

Comandos RPM no están disponibles en el entorno de máquina virtual de este curso.

## 11.3.2.4 RPM - Consulta de Paquetes

La gestión de Red Hat es similar a la gestión de paquetes de Debian en lo respectivo a las consultas que realizan. Lo mejor es utilizar la herramienta de back-end ( rpm ) en lugar de la herramienta de front-end ( yum ). Mientras que las herramientas de front-end pueden realizar algunas de estas consultas, el rendimiento se resiente porque estos comandos normalmente se conectan a varios repositorios a través de la red, cuando se ejecutan cualquier comando. El rpm realiza sus consultas mediante la conexión a una base de datos que es local y no se conecta a través de la red a los repositorios.

Comandos RPM no están disponibles en el entorno de máquina virtual de este curso.

Para obtener una lista de todos los paquetes que están instalados actualmente en el sistema, ejecute el rpm -qa .

Para una lista de los archivos que componen un paquete en particular rpm -ql paquete.

Para consultar un paquete de información, o de su estado, ejecute el rpm -qi paquete.

Para determinar si un archivo en particular fue puesto en el sistema de archivos como el resultado de la instalación de un paquete, ejecute el rpm -qf /ruta/a/archivo

## 11.4 Linux Kernel

Cuando la mayoría de la gente se refiere a Linux, realmente están refiriendo a GNU / Linux , que define el sistema operativo. El Gnu No es Unix (GNU) parte de esta combinación es proporcionado por un proyecto de la Fundación para el Software Libre. GNU es lo que proporciona el equivalente de código abierto de muchos comandos comunes de UNIX, la mayor parte de la línea de comandos esencial. La parte Linux, de esta combinación, es el núcleo de Linux , que es el núcleo del sistema operativo. El núcleo se carga en el arranque y permanece cargado para gestionar todos los aspectos del sistema en funcionamiento.

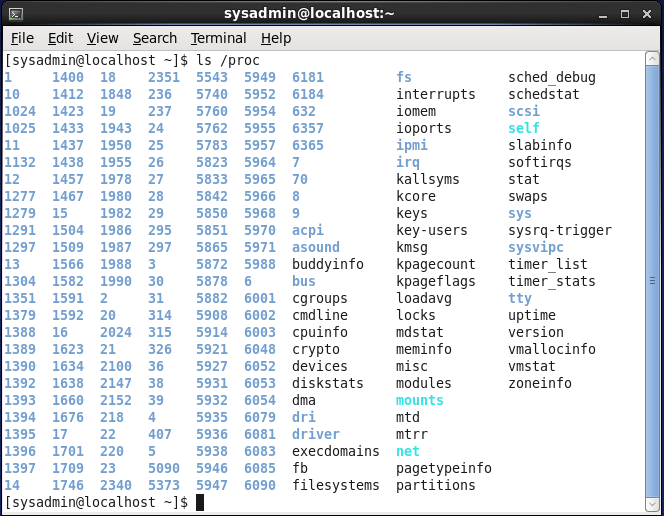
La implementación del núcleo de Linux incluye muchos subsistemas que son una parte del núcleo en sí y otros que se puede cargar de forma modular cuando sea necesario. Algunas de las funciones clave del kernel de Linux incluyen una llamada al sistema de interfaz, gestión de procesos, gestión de memoria, sistema de ficheros virtual, las redes y los controladores de dispositivos.

En pocas palabras, el núcleo acepta comandos del usuario y gestiona los procesos que llevan a cabo esos comandos, dándoles acceso a dispositivos como memoria, discos, interfaces de red, teclados, ratones, monitores y mucho más.

El kernel proporciona acceso a la información sobre la ejecución de los procesos a través de un seudo  sistema de ficheros que es visible bajo el directorio /proc . Los dispositivos de hardware están disponibles a través de archivos especiales bajo el directorio /dev , mientras que la información sobre estos dispositivos se puede encontrar en otro seudo sistema de archivos bajo el directorio /sys .

El directorio /proc  no sólo contiene información sobre los procesos en ejecución, como su nombre indica ( proceso ), también contiene información sobre el hardware del sistema y la configuración actual del núcleo. Ver un ejemplo de salida a continuación:

Tenga en cuenta que la información que se muestra en los ejemplos siguientes será diferente de lo que usted puede ver en el entorno de máquina virtual de este curso.



El resultado de la ejecución de ls /proc muestra más de un centenar de directorios numerados. Hay un directorio numerado para cada proceso que se ejecuta en el sistema, en el que el nombre del directorio coincide con el PID (identificador de proceso) para el proceso en ejecución.

Debido a que el proceso /sbin/init  es siempre el primer proceso, tiene un PID de 1 y la información sobre /sbin/init  se puede encontrar en el /proc/1.

Usted puede ver también que hay una serie de archivos normales en el directorio /proc , como /proc/cmdline , /proc/meminfo y  /proc/modules . Estos archivos proporcionan información sobre el núcleo en ejecución:

* /proc/cmdline   puede ser importante, ya que contiene toda la información que se ha pasado al kernel.
* El /proc /meminfo  información sobre el uso de la memoria por el kernel.
* El /proc/modules  contiene una lista de los módulos actualmente cargado en el núcleo para agregar funcionalidad extra.

Una vez más, rara vez hay una necesidad de ver estos archivos directamente, como otros comandos ofrecen más "amigable" de salida y una forma alternativa para ver esta información.

Si bien la mayoría de los "archivos" /proc  no puede ser modificado, incluso por el usuario root, los "archivos" de /proc/sys  pueden ser modificadas por el usuario root. Estos archivos son especiales ya que cambian el comportamiento del kernel de Linux.

Modificación directa de estos archivos causan sólo cambios temporales en el núcleo. Para realizar cambios permanentes, las entradas se pueden añadir a /etc/sysctl.conf .

Por ejemplo,  /proc/sys/net/ipv4  contiene un archivo llamado icmp\_echo\_ignore\_all . Si ese archivo contiene un cero (0), como lo hace normalmente, entonces el sistema responderá a las solicitudes icmp. Si ese archivo contiene un uno (1), entonces el sistema no responderá a las solicitudes icmp:

**[Usuario @ localhost ~] $** su -

Contraseña: **[root @ localhost ~] #** cat /proc/sys/net/ipv4/icmp\_echo\_ignore\_all

0 **[root @ localhost ~] #** de ping -c1 localhost

localhost.localdomain PING (127.0.0.1) 56 (84) bytes de datos.

64 bytes de localhost.localdomain (127.0.0.1): icmp\_seq = 1 ttl = 64 tiempo = 0.026 ms

--- estadísticas de ping localhost.localdomain ---

1 paquetes transmitidos, 1 recibió, 0% pérdida de paquetes, el tiempo 0 ms

rtt min / avg / max / MDEV = 0.026 / 0.026 / 0.026 / 0.000 ms

**[root @ localhost ~] #** echo 1> /proc/sys/net/ipv4/icmp\_echo\_ignore\_all

**[root @ localhost ~] #** de ping -c1 localhost

localhost.localdomain PING (127.0.0.1) 56 (84) bytes de datos.

--- estadísticas de ping localhost.localdomain ---

1 paquetes transmitidos, 0 recibido, el 100% de pérdida de paquetes, el tiempo 10000ms

## 

## 11.5 Jerarquía de Proceso

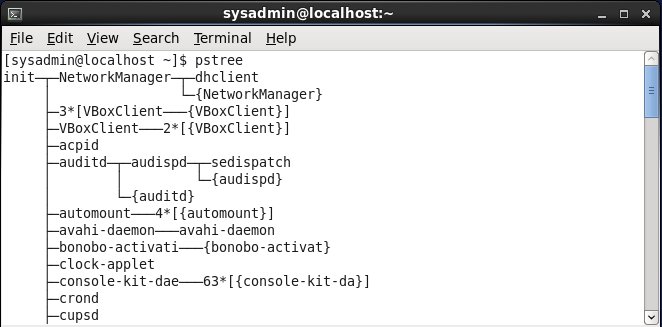
Cuando el núcleo se termina de cargar durante el arranque, se inicia el proceso  /sbin /init  y le asigna un ID de proceso (PID) de 1. Este proceso comienza otros procesos del sistema y a cada proceso se le asigna un PID en orden secuencial.

A medida que el /sbin/init  pone en marcha otros procesos, que a su vez pueden poner en marcha otros procesos. Cuando uno proceso inicia otro proceso, el proceso que realiza la partida se llama el proceso padre y el proceso que se inicia se llama el proceso hijo . Al ver los procesos, el PID padre será etiquetado PPID.

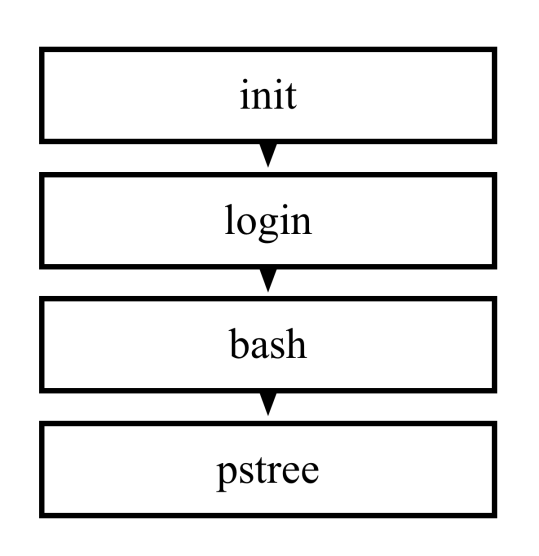
Cuando el sistema ha estado funcionando durante mucho tiempo, se llegará a alcanzar el valor máximo PID , que se puede ver y se configura a través de /proc/sys/kernel/ pid\_max . Una vez que el PID más grande ha sido utilizado, el sistema se reanuda la asignación de valores de PID que están disponibles en la parte inferior de la gama.

Los siguientes gráficos proporcionan un ejemplo y explicación del comando pstree . La salida puede variar de los resultados se pueden ver si se introduce el comando en el entorno de máquina virtual de este curso.

Usted puede "mapear" los procesos en un árbol de familia de padres e hijos.

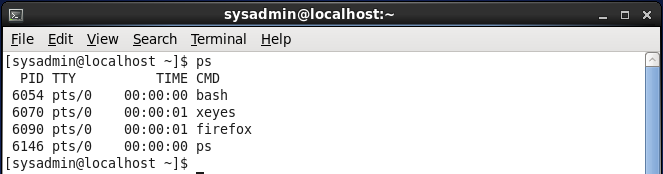


Si se va a examinar la relación de procesos padre e hijo, usando la salida de la orden anterior, usted podría considerar que ser como la siguiente:

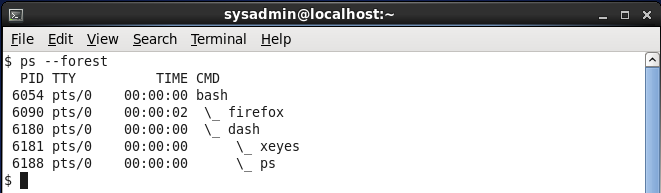


## 11.6 Comando ps (Proceso)

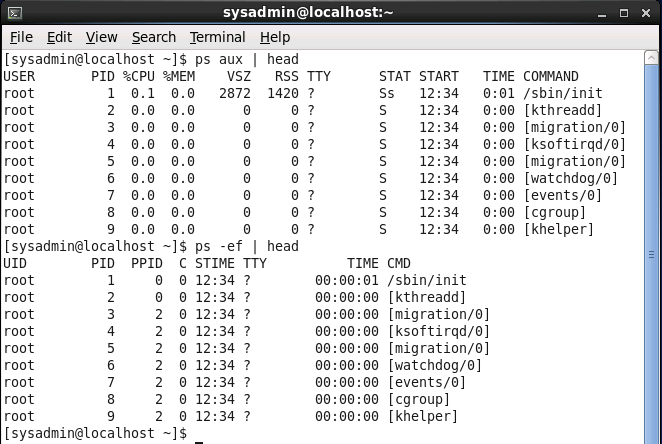
Otra forma de ver los procesos es el comando ps . Por defecto, el ps  sólo mostrará los procesos que se ejecutan en el shell actual.



Similar a la pstree , si ejecuta ps con la opción --forest , entonces se mostrará líneas que indican la relación entre padres e hijos:



Para poder ver todos los procesos en el sistema puede ejecutar ya sea ps aux  o el ps -ef :



Una forma común para ejecutar el ps  es utilizar grep  para filtrar las líneas de la pantalla de salida para que coincidan con una palabra clave, como el nombre del proceso. Por ejemplo, si usted desea visualizar la información sobre el firefox , es posible ejecutar un comando como:

**[Sysadmin @ localhost ~] $** ps -e | grep firefox

6090 pts / 0 00:00:07 firefox

Como usuario root, puede estar más preocupado por los procesos de otro usuario,

Hay diferentes formas de ver los procesos de un usuario individual. El uso de la opción tradicional de UNIX, para ver los procesos de usuario :

**[Root @ localhost ~] #** ps -u nombre de usuario

O utilice el estilo BSD de opciones y ejecutar:

**[Root @ localhost ~] #** ps u U usuario

## 11.8 Comando free

La ejecución de la free  sin ninguna opción, proporciona una instantánea de la memoria utilizada en ese momento.

Si desea supervisar el uso de memoria en el tiempo, entonces se puede ejecutar con la -s  y especificar que el número de segundos. Por ejemplo, la ejecución delibre -s 10 actualizaría la salida cada diez segundos.

Para hacer más fácil la interpretación de lo que el free comando está enviando el -m o –g pueden ser útiles para mostrar la salida, ya sea en megabytes o gigabytes, respectivamente. Sin estas opciones, la salida se muestra en bytes:

**$** free

total utilizada libre buffers compartidos en caché

Mem: 510984 495280 15704 0 60436 258988

- / + buffers / cache: 175 856 335 128

Intercambiar: 1048568 0 1048568

Cuando se lee la salida de free:

* La primera línea es una cabecera descriptiva.
* La segunda línea denominada "Mem:" es las estadísticas de la memoria física del sistema.
* La tercera línea representa la cantidad de memoria física después de ajustar esos valores al no tener en cuenta la memoria que está siendo utilizado por el kernel para tampones y cachés. Técnicamente, esta memoria "utilizado" podría ser "recuperado" si es necesario.
* La cuarta línea de producción se refiere a la memoria "Swap", también conocida como memoria virtual. Este es el espacio en el disco duro que se utiliza como memoria física cuando la cantidad de memoria física es bajo. Efectivamente, esto hace parecer que el sistema tiene más memoria de lo que realmente hace, pero el uso de espacio de intercambio también puede ralentizar el sistema.

Si la cantidad de memoria y de intercambio que está disponible se vuelve muy baja, entonces el sistema comenzará a terminar automáticamente procesos. Esta es una razón por qué es importante para supervisar el uso de memoria del sistema. Un administrador que da cuenta del sistema de convertirse en poca memoria libre puede utilizar la parte superior o matar aponer fin a los procesos de su propia elección, en lugar de dejar que el sistema elija.

Conceptos:

1. Distribuciones que utilizan el sistema de gestión de paquetes Debian
2. Los archivos se distribuyen en archivos .\*\*\*.
3. ¿Qué es dpkg?
4. ¿Qué es apt-get?
5. ¿Enumera herramientas GUI para la gestión de paquetes?
6. Define la utilidad de cada uno de estos comandos
7. apt-get update
8. apt-get install *paquete*
9. apt-get upgrade.
10. apt-get remove *paquete*
11. apt-get --purge *paquete*
12. ¿Qué es rpm?
13. ¿Qué es yum?
14. ¿Qué es el kernel de Linux?
15. ¿Qué tenemos en el directorio /proc?
16. Define jerarquía de procesos Y proceso hijo
17. Comandos:
18. Pstree
19. Ps
20. --forest
21. Ps aux
22. Free