Paquetes, Repositorios e Instalación de software en GNU/Linux

#### 

# **Ejercicio A.1**

**Instala el siguiente software en la máquina virtual de GNU/Linux de clase. Para cada programa responde a las siguientes preguntas e indica qué comandos usaste y toma capturas de pantalla que muestran el resultado:**

1. **¿Cómo compruebas si el programa estaba instalado o no? Si estaba instalado, comprueba si lo puedes actualizar. Si no estaba instalado, instálalo y muestra que funciona correctamente.**
2. **¿Qué versión estaba instalada (si había alguna) y cuál es la que se va a instalar? ¿Has instalado la última versión disponible del software o una más antigua (si puedes, indica la versión instalada y la más reciente, indicando incluso la diferencia de fechas si conoces ese dato)?**
3. **¿Cómo instalaste el programa? (repositorio, paquete descargado, PPA, etc.) ¿por qué elegiste esta opción?**
4. **¿Estaba el programa en algún repositorio? Si es afirmativo, ¿en qué repositorio? ¿por qué estaba justo en ese repositorio y no en otro?**
5. **¿Sabes qué dependencias, recomendaciones y/o sugerencias tiene el software a actualizar?**
6. **Elimina totalmente el nuevo software instalado SI NO LO TENÍAS PREVIAMENTE**

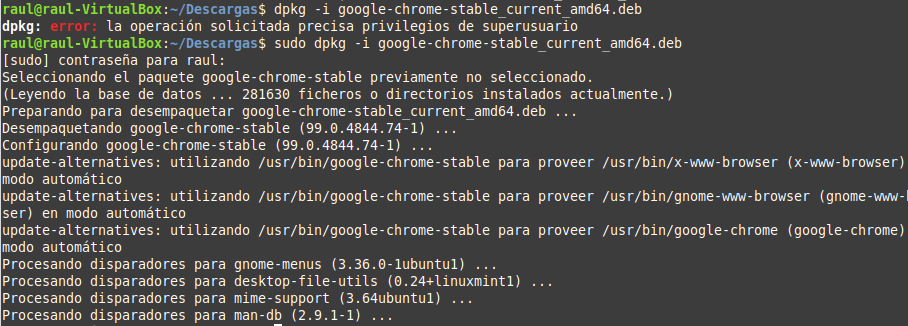
SOFTWARE A INSTALAR:

1. **Chrome**
2. Tras descargar el paquete de manera manual:

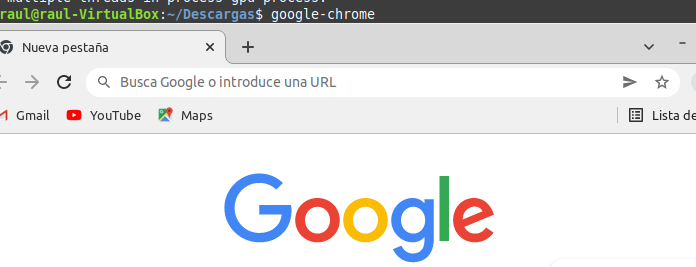


Comprobaremos si este se ha instalado:

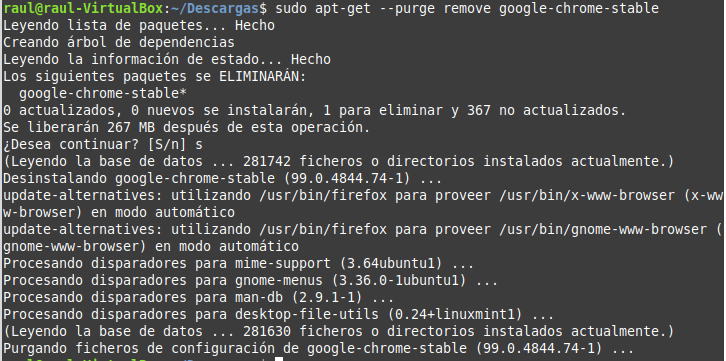


Tras lanzar el comando y no mostrar nada procedemos con el comando de instalación: (Con este comando descomprimimos e instalamos google Chrome )

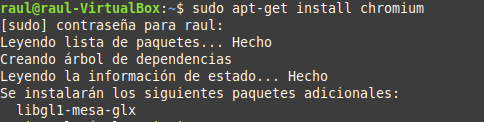
Comprobamos que funciona correctamente:



1. Se instalará la versión más reciente (no estaba instalado)
2. Se instaló mediante descarga manual y posterior descompresión, ya que no está oficialmente comprobado por GNU.
3. No, ya que no pertenece a los repositorios de GNU



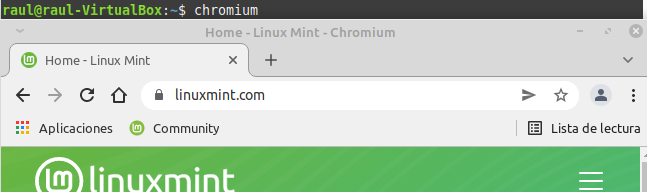
1. **Chromium**
   1. Tras intentar lanzarlo desde la terminal y comprobar que no existe procedemos a instalarlo:



Comprobamos que está instalado:



Comprobamos que funciona correctamente:



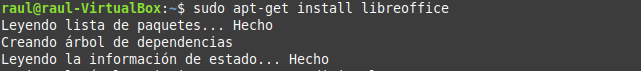
* 1. La versión instalada es:



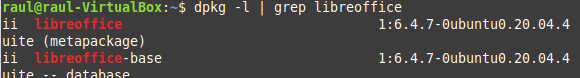
* 1. Se instala desde repositorio, ya que está oficialmente soportado por GNU**(revisar)**
  2. Sí, en el oficial de GNU

Recomiendan las dependencias para desarrolladores

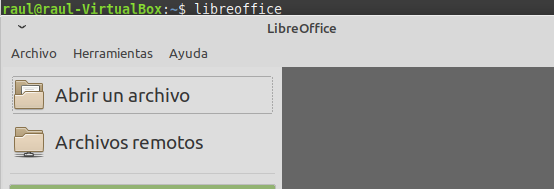
1. **LibreOffice 7.3.x**
   1. Tras intentar lanzarlo desde la terminal y comprobar que no existe procedemos a instalarlo:



comprobamos que se ha instalado correctamente:



lo lanzamos:



* 1. La versión que hemos instalado es la:



Por otro lado la última versión estable es la 7.3.1 (info) y salió el ( 03 de marzo de 2022)

Recomiendan las dependencias para desarrolladores

1. **MS Teams**
   1. Descargamos el paquete con el comando:

wget https://go.microsoft.com/fwlink/p/?linkid=2112886 -O microsoft-teams.deb

Instalamos con el comando:

apt install ./microsoft-teams.deb

Comprobamos que está instalado con:

dpkg -l | grep teams

b. Comprobamos la versión instalada con:

msteams -- version

c. Se recomiendan las dependencias de desarrollador

1. **notepad++ (o equivalente)**
   1. Instalaremos con el comando:

snap install notepad-plus-plus

* 1. Comprobamos que se ha instalado con:

dpkg -l | grep notepad

* 1. comprobamos la versión con:
  2. notepad++ – version
  3. Se recomiendan las dependencias de desarrollador

1. **nvidia-utils**
   1. Instalaremos con el comando:

ubuntu-drivers autoinstall

* 1. Reiniciamos el equipo y comprobamos el estado de la tarjeta gráfica con:

nvidia-smi

* 1. Comprobamos la versión instalada con:

nvidia-utils – version

1. **Skype**
   1. Descargamos el paquete con:

wget <https://go.skype.com/skypeforlinux-64.deb>

* 1. Ejecutamos el instalador con:

dpkg -i skypeforlinux-64.deb

* 1. Comprobamos que está instalado con:

dpkg -l | grep notepad

* 1. No se encuentran las dependencias recomendadas

1. **Thunderbird**
   1. Añadimos el repositorio de mozzila con:
2. **Un antivirus**
3. **Virtualbox**

PASO FINAL (al acabar el ejercicio):

1. Cuando acabes el ejercicio ¿qué paquetes quedan en el sistema que ya no son necesarios? Elimina todos estos paquetes innecesarios
2. Revisa la lista de repositorios que tienes activos en el sistema. Elimina o desactiva los repositorios PPA o de terceros del software que has instalado y luego desinstalado en este ejercicio.

#### 

#### 

# Ejercicio A.2

**Indica en qué directorio(s), en general, encontrarías:**

1. **Los binarios y las librerías de las aplicaciones del usuario instaladas desde repositorio:**

Los binarios se encuentran /usr/bin y las bibliotecas en /usr/local/lib.

1. **Las aplicaciones instaladas de otras fuentes (no en repositorios oficiales)**

En /usr/local/bin.

1. **La configuración de las aplicaciones**

En el directorio /etc.

1. **El manual de las aplicaciones**

En el directorio /usr/man

1. **Los logs o registros de las aplicaciones**

En el directorio /var/log y en sus subcarpetas

1. **Los binarios del sistema (los utilizados por los usuarios y los utilizados por el propio sistema)**

El directorio /sbin contiene los comandos necesarios para bootear el sistema, pero usualmente no son ejecutados por usuarios normales.

1. **Las librerías del sistema**

Se guardan en /lib

1. **Dispositivos externos (pendrives, disco duro externo, etc.)**
2. **Los ficheros temporales**

En /tmp.

1. **Dispositivos del sistema**

En /dev.

1. **El espacio personal de los usuarios**

En /home/usr.

1. **El espacio personal del administrador**

El espacio del administrador es /root.

1. **Datos del sistema (firmware, kernel, etc.)**

Se guardan en /etc/conf.modules

1. **Datos de los procesos del sistema**

En el directorio /proc.

1. **Datos del arranque del sistema**

En el directorio /boot

# Ejercicio A.3

Crear un script que:

1. Debe recibir un argumento (el nombre de un paquete). Si no lo recibe, abortará y mostrará la ayuda
2. Comprobará si el paquete indicado se encuentra instalado en el sistema. Si es así, indicar la versión instalada.
3. Si el paquete está instalado, se mostrará un menú con opciones sobre ese paquete (implementar cada operación): Actualizar, Reinstalar, Eliminar dejando configuración, Eliminar totalmente, Salir.
4. Si no está instalado, comprobar que el paquete está disponible para ser instalado desde algún repositorio. Si es así, ofrecerle la opción para instalarlo. Si no está disponible, mostrarle el resultado de una búsqueda del parámetro indicado.

### 

Procesos

# Ejercicio B.1

Resuelve en GNU/Linux los siguientes apartados. Cuando se pregunte por qué comando se utilizará, debes indicar el comando completo, con las opciones y parámetros necesarios para que funcione:

1. Ejecutar el comando ps aux e indicar qué significa cada columna. ¿Qué estados puede tener un proceso y qué significa cada uno? (si se tienen dudas, usar man ps o ps --help)

Al ejecutar el comando ps aux se nos muestra una tabla con las columnas (USER, PID, %CPU, %MEN, VSZ, RSS, TTY, STAT, START, TIME, COMMAND) y cada uno de ellos significa:

* USER: Es el usuario con el que se ejecuta el proceso
* PID: Se muestra el ID del proceso
* %CPU: porcentaje de tiempo que el proceso estuvo en ejecución desde que se inició
* %MEM: porcentaje de memoria física utilizada
* VSZ: Es la memoria virtual que ocupa el proceso y es medida en KiB
* RSS: "resident set size", es la cantidad de memoria física no swappeada que la tarea a utilizado (en KiB)
* TTY: Es la terminal de control asociada con el proceso .
* STAT: código de estado del proceso (información detallada más adelante)
* START: Es la hora de inicio del proceso.
* TIME: tiempo total de uso de la CPU.
* COMMAND: comando con todos sus argumentos.

1. Ejecutar el comando uptime. ¿Qué significa cada uno de los datos que muestra y qué información puedo obtener de ellos?



* **16:08:16:** Nos muestra la hora actual.
* **up 2:30:** 19:32, información sobre el tiempo que lleva la máquina encendida.
* **1 user:** Número de usuarios conectados actualmente.
* **load average: 0,02, 0,02, 0.00:** La carga del sistema actual al minutos, cinco minutos y quince minutos.

1. **¿Un sistema con 8 cores tiene una carga de 5.3, a qué porcentaje aproximado de rendimiento está funcionando este equipo?**
2. ¿Qué información me muestra el comando jobs? **¿Qué significan los símbolos + y - que aparecen en la lista?**

El comando jobs nos muestra los trabajos o procesos vinculados a la terminal desde donde se ejecuta dicho comando.

En lo referente a los simbolos + y - si observamos el comando jobs al realizar un bg o un bg + este pondrá a ejecutarse al último proceso añadido a la lista en cambio si hacemos bg -

afectará al penúltimo proceso añadido a la lista (Los + y - van cambiando de posición ya que al realizar un cambio en el proceso - este pasará a ser el último en recibir el + de nuevo).

1. **Al ejecutar el comando jobs veo que, por ejemplo, el elemento [3] de la lista está en estado stopped (Terminado, terminated o similar), ¿qué significa esto? ¿cómo puedo hacer que pase a ejecutarse en segundo plano?**

Esto significa que el proceso se encuentra detenido. para pasarlo a segundo plano realizaremos un ctrl+z.

1. He lanzado un comando xeyes que ya se está ejecutando en primer plano, indicar qué tendría que hacer para pasarlo a ejecutar en segundo plano (y que siga ejecutándose)

tras lanzarlo realizaremos un ctrl+z y posteriormente un bg.

1. Lanzar el comando xeyes en segundo plano, y luego volver a pasarlo a primer plano.

Primero lanzaremos el comando ***xeyes &*** de esta manera se iniciará directamente en segundo plano, para luego pasarlo a primer plano haremos un jobs y miraremos su id correspondiente (si existe más de un proceso) y lanzaremos el comando ***fg num*** para pasarlo a primer plano.

1. ¿Cómo puedo hacer que el comando xeyes se siga ejecutando aunque se cierre la shell en la que se ejecutó si aún no lo he lanzado?

Lanzando el xeyes con el comando ***nohup xeyes.***

1. ¿Cómo puedo hacer que el comando xeyes se siga ejecutando aunque se cierre la shell en la que se ejecutó si ya se está ejecutando?

Podemos usar el comando ***disown xeyes.***

1. **Como usuario tienes que lanzar el comando xeyes con la mínima prioridad posible, ¿qué comando utilizarías? ¿qué valor le asignarías y cuánta es la prioridad efectiva final? ¿por qué? ¿qué pasaría en el caso inverso, que quisiera asignarle la máxima prioridad posible antes de lanzarlo?**
2. **Repetir el ejercicio anterior, pero sobre el comando xeyes que ya se está ejecutando (indicar qué comando hay que utilizar para modificar la prioridad de un proceso que ya está en ejecución, y qué valores máximos y mínimos se pueden asignar)**

#### 

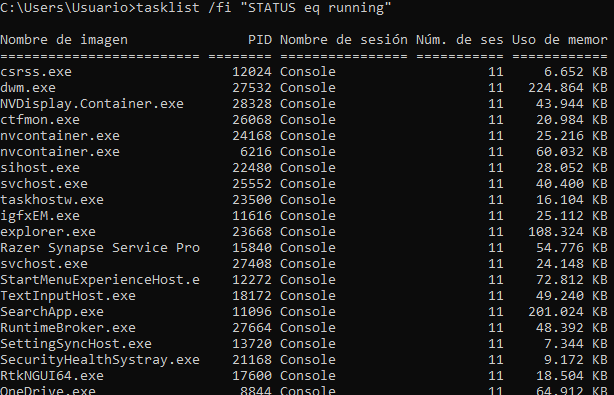
# Ejercicio B.2

Ejercicio de procesos en MS Windows:

1. Lista todos los procesos actuales de Windows

Usaremos elcomando ***tasklist.***

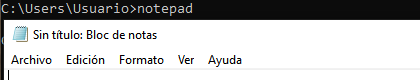
1. Lista sólo los que están en ejecución



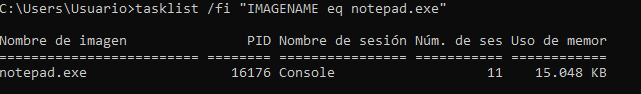
1. Lista sólo los que están suspendidos



1. Abre un notepad en Windows desde línea de comandos



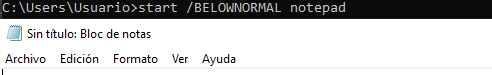
1. Filtra el listado de procesos para obtener sólo información del notepad



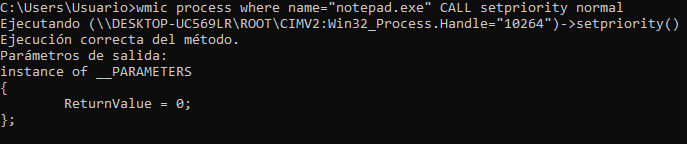
1. Finaliza el notepad filtrando por nombre



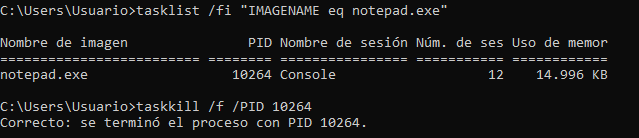
1. Lanza el notepad con prioridad "por debajo de normal"



1. Cambia la prioridad del notepad anterior a Normal



1. Finaliza el notepad filtrando por PID



# Ejercicio B.3

Crea un script de bash con las siguientes características:

1. Recibe un único argumento, que puede ser: CPU, MEM o TIME (si el argumento falta, o se le pasan varios argumentos o no son válidos, entonces por defecto usará la opción CPU).
2. Si el usuario intenta interrumpir o matar (terminar) el script cuando esté en ejecución, se mostrará un mensaje diciendo que no es posible, que elija la opción Salir del menú.
3. Obtiene el proceso que MÁS CONSUMO tenga de %CPU, %MEM o TIME, según se haya indicado en el argumento.

PISTA: para obtener esta información basta con usar un comando similar al siguiente, donde la variable c guarda la columna que se va a coger como referencia para ordenar (según el formato de ps aux, así la %CPU es la columna 3, %MEM es la columna 4 y TIME es la 10):   
info=(`ps aux | tr -s " " | tail -n +2 | sort -hrk $c | head -n 1`)  
De esta forma, info será un array con todos los datos que ofrece ps aux. Por ejemplo, el PID está en ${info[1]}, el usuario en ${info[0]}, el estado en ${info[7]}, etc.

1. Luego muestra un menú que se repite hasta que el usuario indica que quiere salir. Antes de las opciones se mostrará la información sobre ese proceso, como usuario, PID, %CPU, %MEM, estado, tiempo, etc. (si usaste el comando anterior, ya tienes esta información en el array "info"). Las opciones del menú (se deben implementar las operaciones) se mostrarán a continuación, y son:
   1. Interrumpir el proceso
   2. Terminar el proceso
   3. Matar el proceso
   4. Detener el proceso y pasarlo a segundo plano
   5. Reanudar el proceso en segundo plano
   6. Pasar el proceso a segundo plano
   7. Asignarle la peor prioridad posible
   8. Evitar que el proceso muera si se cierra la shell donde se ejecuta.
   9. Salir  
      PISTA: Las señales que se pueden usar son: SIGCONT, SIGINT, SIGLKILL, SIGSTOP (o SIGTSTP), SIGTERM

### 

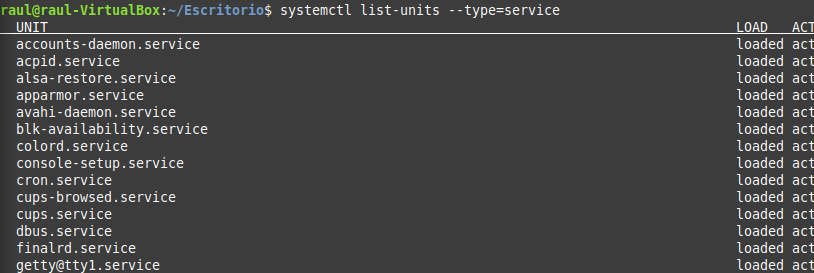
Servicios

# Ejercicio C.1

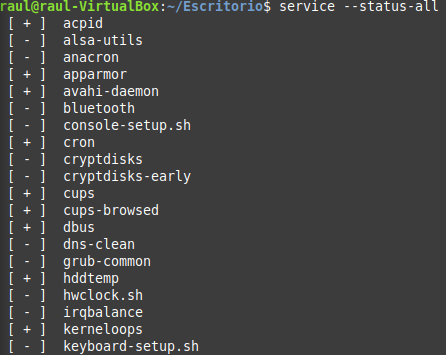
Resuelve los siguientes apartados en GNU/Linux. Cuando se pregunte por qué comando se utilizaría, debes indicar el comando completo, con las opciones y parámetros necesarios para que funcione:

1. Muestra todos los servicios de tu sistema

systemctl list-units --type=service



o en su defecto service --status-all:



1. ¿Cómo sabes si el servicio está activo o no?

Con el comando service --status-all mostrado anteriormente podemos ver los símbolos + y - justo antes del nombre del servicio. Esto nos indica su estado, siendo el + activo y el - inactivo.

1. Elige 3 servicios de la lista y, para cada uno de ellos, indica:
   1. El nombre

Bluetooth

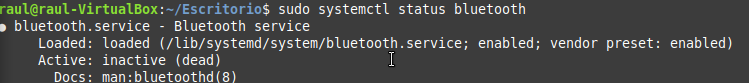
* 1. Investiga un poco para qué se usa

Es una tecnología de red que sirve para la transferencia de voz y datos punto a punto sin conexión u orientada a la conexión entre dos dispositivos digitales diferentes.

* 1. Si está activo o no (¿qué significa esto?)

 Se encuentra inactivo ya que no está siendo usado en este momento.

* 1. Si está habilitado o no (¿qué significa esto?)



El proceso se encuentra habilitado, lo que significa que puede llegar a ser usado.

* 1. Si está enmascarado o no (¿qué significa esto?)



En la columna de estado podemos observar que no se encuentra enmascarado (masked) y esto significa que se puede iniciar.

FALTAN 2 mas

#### 

# Ejercicio C.2

a) Elige un servicio no primordial de Windows y usa comandos sc para obtener datos de este servicio y de su estado actual. Prueba a detenerlo, iniciarlo, pausarlo, reanudarlo, etc. Déjalo en su estado inicial.

b) Repite el apartado anterior usando PowerShell

# Ejercicio C.3

Crear un script en bash que reciba un único argumento (obligatorio) que debe ser el nombre de un servicio. A partir de este nombre:

1. Si el servicio no está en el sistema, da un error y termina la ejecución
2. Si el servicio está en el sistema, se mostrará su estado (si está activo/inactivo, habilitado/deshabilitado o enmascarado/no enmascarado)
3. Luego se mostrarán tres preguntas para cambiar el estado. Por ejemplo, el servicio está activo, ¿desea usted pasarlo a inactivo? (o viceversa, según el estado). Preguntar lo mismo para cambiar el estado de habilitado y enmascarado