

Visita nuestro sitio SoloLinux.es

MAGAZINE SÓLO LINUX

Nº
12

Tu revista, la revista de tod@s

ENERO 2020



Como usar el comando
nmcli con ejemplos

Instalar **Cuda** en Ubuntu
y derivados

No puedo conectar
Windows con **Samba**

Unix vs Linux

Ifconfig:
command not found

Formatear y **montar** un disco
de más de 2 terabytes en Linux

MANUALES, SCRIPTS, SOFTWARE, HARDWARE, DISTROS LINUX,
SEGURIDAD, REDES Y MUCHO MAS EN LA WEB...

WWW.SOLOWORDPRESS.ES - NÚMERO 02 - DICIEMBRE 2019-ENERO 2020

Numero 2. Proximamente

Últimas novedades

Trucos

Consejos útiles

Manuales paso a paso

Debates abiertos

Opiniones de expertos

Artículos

¡Si crees que
puedes ayudar contacta
con nosotros!

SoloWordpress

Tu revista sobre Wordpress

SoloWordpress

Inicio Contacto

Novedades

Recomendado

Información Nivel Avanzado
Nivel Intermedio Trucos WordPress
Cómo hacer un plugin en WordPress

Más vistos

| | |
|--|---|
| | Actualidad Información Nivel Avanzado Nivel Intermedio Técnico Las Señales de Seguridad |
| | Actualidad Información Nivel Avanzado Nivel Intermedio Técnico De http a https en WordPress |
| | Actualidad Información Nivel Avanzado Nivel Intermedio Técnico De http a https en WordPress |

Buscar ... Buscar

Entradas Recientes

- WordPress y la Seguridad que no puedes controlar
- Nueva actualización 5.3.1
- Las Señales de Seguridad
- De http a https en WordPress
- Las constantes de WordPress 5.3

**Si el formato digital no te
convence, también tenemos
todo el contenido en una
Página Web**

¡Visítanos!

Bienvenido a la Revista SOLOLINUX

Os presento el numero 12 de la revista Sololinux. Este numero tan ansiado para cualquier revista, ello conlleva a que **cumplimos un año**. Desde SOLOLINUX esperamos que os guste este numero.

Igual que en el numero anterior nos gustaría animar a todos nuestros lectores para que nos envíen sus **opiniones sobre el Software Libre o sobre GNU/Linux**, pueden enviarlo a adrian@solistlinux.es, con ello queremos proponer que cada mes se publicada una o varias de esas opiniones sobre lo mencionado en la nueva sección de la

revista OPINIÓN DEL LECTOR. Queremos saber la opinión de todos. Se intentara incluir el máximo de opiniones en cada numero, pero si no sale la tuya este mes no desesperes, al siguiente podria tener un hueco en la revista.

ANÍMENSE Y ENVÍEN SUS OPINIONES. Gracias.

Al igual que lo anteriormente mencionado, nos gustaría promover un espacio en la revista sobre los eventos de Software Libre y GNU/Linux en todo el mundo. Los organizadores de estos eventos pueden ponerse en contacto con migo a través de correo electrónico, adrian@solistlinux.es

Sin mas **quiero agradecer a todos** los que hacéis posible que esta revista siga adelante.

Personalmente agradezco a Sergio todo su trabajo en la multitud de artículos que realiza a lo largo del mes para que esta revista pueda tener suficiente información mes a mes.

Gracias a TOD@S

Compartan esta revista en sus redes sociales o web. Revista digital **SOLOLINUX MAGAZINE**. Tu revista, la revista de todos.

Adrián A. A.

Síguenos en las Redes:



La revista SOLOLINUX esta realizada con Libre Office Impress 6.2.8.

Nuestras Webs:

www.sololinux.es
www.solowordpress.es

EDITORIAL

Edición:

- Adrián Almenar
adrian@solistlinux.es

Redacción:

- Sergio G. B.
(Administrador y redactor artículos Sololinux)
info@solistlinux.es

- Henry G. R
(Redactor artículos SolowordPress)
info@solowordpress.es

Agradecimientos:

Publicidad:

Quieres poner publicidad en la revista, ahora puedes hacerlo de forma muy simple, llegando a todo el mundo con esta revista digital de software libre y GNU/Linux en ESPAÑOL

CON SOLOLINUX MULTIPLICARAS TUS CLIENTES

Para mayor información escribe un email a: adrian@solistlinux.es

Colabora:

Quieres colaborar en la revista. Para mayor información escribe un email a: adrian@solistlinux.es

La **Revista SOLOLINUX**, se distribuye gratuitamente en forma digital para todo el mundo que quiere disfrutar de ella. Si quieres imprimirla es cosa tuya. Si os cobran por ella, os están timando. :)

Contacto:

Para cualquier consulta sobre las revistas, publicidad o colaboraciones escribir un email a:

adrian@solistlinux.es

AYUDANOS A SEGUIR CRECIENDO



Este obra se publica bajo una licencia de Creative Commons Reconocimiento-Compartir-Igual 4.0 Internacional.

MANUALES

- Diferencias entre ZIP y GZIP
- Como usar el comando nmcli con ejemplos
- Instalar Cuda en Ubuntu y derivados
- No puedo conectar Windows con Samba
- Simular comandos en linux
- Instalar y configurar Glances en linux
- Ubicación geográfica de una ip en terminal
- Colorear la salida de comandos con el arcoíris
- Instalar Ganglia Monitoring Server en Ubuntu
- Cómo hacer una prueba de carga en un servidor con Siege
- Uso del comando seq con ejemplos
- Como iniciar el sistema en modo consola

SEGURIDAD

- Generar contraseñas en linux
- AppArmor vs SELinux
- Navegar de forma anónima con Tor y ProxyChains

DISTROS LINUX

- Las mejores distribuciones linux del 2020
- Por qué Manjaro es tan buena distribución

JUEGOS

- Minetest: la mejor alternativa a Minecraft

HARDWARE

- Formatear y montar un disco de más de 2 terabytes en linux



REDES

- Ifconfig: command not found
- Uso del comando ip con ejemplos

NOTICIAS

- Unix vs Linux

SOFTWARE

- Novedades de Mozilla Firefox 72
- Instalar MystiQ Video Converter en Ubuntu y derivados
- Alternativas a Notepad++ en Linux
- Como instalar los plugins de Gedit

SCRIPTS

- Verificar que tenemos habilitado http2 en nuestra web
- Monitorizar el uso de la CPU y avisar por mail
- Información del sistema con Python
- Monitorizar el uso del disco y avisar por mail
- Monitorizar el uso de la memoria y avisar por mail
- Como escribir scripts bash correctamente
- Cómo desfragmentar en linux
- Conocer el estado de fragmentación de un disco





la gama más completa de ordenadores linuxeros

VEN A LINUX CON TOTAL GARANTÍA

... y calidad,
... y total compatibilidad,
... y esmerado soporte,
... y el mejor precio.



Desde 2011, ofreciendo nuestra completa gama de ordenadores pensados para que te olvides del hardware y disfrutes sin problemas del mundo de GNU/Linux y el Software Libre desde el primer momento

descúbrenos en www.vantpc.es

Síguenos para enterarte de todas nuestras noticias, novedades y ofertas



The screenshot shows a news article titled "Lanzan la nueva distribución Tails 3.13" by sololinux on 21/03/2019. The article features a large image of the Tails logo (a purple book with a keyhole) and discusses the release of Tails 3.13, which is based on Debian and includes various improvements and bug fixes. The website has a dark theme with a blue header bar containing links for DESARROLLOS WEB, DISTROS LINUX, ECASH, HARDWARE, MANUALES, NOTICIAS, REDES, SCRIPTS, SOFTWARE, and a search bar. There are also social media sharing buttons and a PayPal donation button.



THANKS!



TU PUBLICIDAD AQUÍ
QUIERES APARECER EN
LA REVISTA, GANAR
CON ELLO MAS VENTAS
EN TU WEB, MAS
SEGUIDORES EN TUS
REDES SOCIALES...



SOLO TIENES QUE
MANDAR UN CORREO A
adrian@sololinux.es
Y TE EXPLICAMOS
COMO



SOLOLINUX

CURSO

Seguridad

en Redes Linux

100% ON LINE Y A TU RITMO

TUTOR
FABIÁN AMPALIO



+54 9 11
6969 9993



SEGUINOS EN
Instagram
@fabianampalio



Diferencias entre ZIP y GZIP

Tanto **Gzip** como **Zip**, son formatos populares si queremos comprimir archivos y ahorrar espacio en nuestro disco, o acelerar la velocidad de transferencia en redes locales, privadas o públicas.

En este artículo vamos a explorar las **principales diferencias entre zip y gzip**, que son los métodos de compresión más utilizados en sistemas operativos Linux, Windows y Mac.



Diferencias entre ZIP y GZIP

La **extensión .gz** es la que se usa en los archivos comprimidos Gzip, la **extensión .zip** es la que utilizan los archivos ZIP.

La verdad es que **zip** es mucho más popular que **gz**, algo lógico pues zip es de Windows (y su maquina monopolista). Por otro lado tenemos a Gzip, que es el formato de compresión de archivos estándar en sistemas Unix, Linux y Mac OS.

ZIP puede empaquetar y comprimir archivos o directorios por sí mismo, Gzip hace el mismo trabajo pero necesita la ayuda del comando tar.

Las múltiples opciones de archivo que ofrecen ZIP y Gzip + tar, permiten a cualquier usuario del sistema combinar múltiples archivos en un solo archivo tar.gz o zip.

Su fórmula de ejecución es muy similar, desde el escritorio de tu distribución linux o Windows puedes hacer múltiples operaciones. Pero ojo, tar.gz puede trabajar en linea de comandos lo que repercute positivamente en su rendimiento y opciones.

Ejemplo...

```
tar -xvpzf archivo.tar.gz
gzip -d archivo.gz
```

A grandes rasgos te puedo decir que la compresión zip no es tan poderosa como gzip. Gzip es mucho más rápido comprimiendo y descomprimiendo, y lo más importante... el archivo resultante de la compresión gzip es mucho menor que si utilizamos zip.

Como punto final vemos sus características.

- **ZIP** es más popular en sistemas Windows.
- **Gzip** es la compresión de archivos estándar en Unix y Linux.
- **Gzip** es mucho más rápido que ZIP al comprimir y descomprimir.
- **ZIP** es una herramienta de compresión todo en uno, **Gzip** necesita el comando Tar para poder archivar.
- **Gzip** ahorra más espacio que si usas ZIP.
- **Gzip** y **ZIP** usan el algoritmo de compresión DEFLATE (compresión de datos sin perdidas).

Personalmente siempre recomiendo GZIP, pero es tu decisión.

Te recomiendo que revises este anterior artículo, donde tienes varios ejemplos de uso del comando tar con gzip.

Como usar el comando nmcli con ejemplos

El **comando nmcli** es una herramienta en **terminal**, que administra el **NetworkManager** de nuestro linux.

Existen aplicaciones para nuestro escritorio preferido muy populares, como **nm-applet** u otros tantos clientes gráficos. Ademas **nmcli** es mucho más rápido y efectivo que cualquier aplicación **GUI**, muy útil para crear, mostrar, editar, eliminar, activar o desactivar las conexiones de red, también es capaz de controlar y mostrarnos el estado del hardware de red.



Como usar el comando nmcli con ejemplos

En este articulo veremos **como usar el comando nmcli con ejemplos**; no olvides que para poder **administrar NetworkManager** correctamente se debe estar ejecutando el servicio (NetworkManager).

Sintaxis esencial del comando nmcli

Antes de comenzar con los ejemplos de uso, vemos un listado de la sintaxis más habitual de nmcli.

| Sintaxis | Uso |
|--------------------------------|--|
| nmcli dev status | Estado de todas las interfaces de red instaladas |
| nmcli con show | Listar las conexiones de red |
| nmcli con show name | Especificar la conexión de red a listar por su nombre |
| nmcli con add con-name name... | Agregar una nueva conexión llamada "name" |
| nmcli con mod name... | Modificar el name de la conexión "name" |
| nmcli con reload | Obligar a NetworkManager que recargue los archivos de configuración |
| nmcli con up name | Activar la conexión "name" |
| nmcli dev dis wlp2 | Desactivar la conexión actual de la interfaz de red indicada (en este caso wlp2) |
| nmcli con del name | Eliminar la conexión y sus archivos de configuración de la conexión indicada (en este caso name) |

Verificar que NetworkManager se está ejecutando

Para saber si NetworkManager se está ejecutando.

```
nmcli -t -f RUNNING general

# ejemplo
sololinux ~ # nmcli -t -f RUNNING general
ejecutando
sololinux ~ #
```

Si queremos ver su estado en general.

```
nmcli general

# ejemplo
sololinux ~ # nmcli general
ESTADO      CONECTIVIDAD      WIFI-HW    WIFI      WWAN-HW    WWAN
conectado   total            activado   activado   activado   activado
sololinux ~ #
```

Ver los dispositivos de red disponibles.

```
nmcli dev status

# ejemplo
sololinux ~ # nmcli dev status
DISPOSITIVO  TIPO      ESTADO          CONEXIÓN
wlp2s0       wifi      conectado      OPUnv0805 1
wlxd06f4a8d013c  wifi      desconectado   --
enp1s0       ethernet  no disponible   --
lo           loopback  sin gestión   --
sololinux ~ #
```

Ver las conexiones disponibles en el sistema

Con el siguiente comando vemos las conexiones disponibles en el sistema, **ojo**, no lo confundas con las **redes wifi disponibles** (se refiere a los archivos de configuración creados).

```
nmcli con show

# ejemplo
sololinux ~ # nmcli con show
NOMBRE          UUID                TIPÓ
DISPOSITIVO
OPUnv0805 1    ba642901-7caa-4d1b-8fed-cff0189de346  802-11-wireless
wlp2s0          603e13c9-1ca1-4dcd-b216-67a6e0980ff3  802-11-wireless  --
Archos Plat    22e8cd9f-20ca-346f-b671-da21845fb22d  802-3-ethernet  --
Conexión cableada 1  bf2c25d5-95c9-4ece-8daf-f4ea2ecf7719  802-11-wireless  --

nmcli con show eth1

# ejemplo
connection.id:          eth1
connection.uuid:         22e8cd9f-20ca-346f-b671-
connection.stable-id:    --
connection.type:         802-3-ethernet
connection.interface-name: eth1
connection.autoconnect:  yes
```

Verificar el estado físico de las interfaces de red

```
nmcli dev status

# ejemplo
sololinux ~ # nmcli dev status
DISPOSITIVO          TIPO    ESTADO      CONEXIÓN
wlp2s0                wifi    conectado   0PUuv0805 1
wlxd06f4a8d013c      wifi    desconectado --
eth1                  ethernet no disponible --
lo                   loopback sin gestión --
sololinux ~ #
```

Modificar el hostname con nmcli

Como alternativa, el comando nmcli también nos permite modificar el **hostname** del sistema.

```
nmcli general hostname

# el resultado es:
sololinux ~ # nmcli general hostname
sololinux
sololinux ~ #
```

Como vemos en el ejemplo anterior, nuestro hostname es «sololinux» así que lo cambiamos a «sololinux.es».

```
nmcli general hostname sololinux.es

# ejemplo de resultado

sololinux ~ # nmcli general hostname sololinux.es
sololinux.es
sololinux ~ #
```

Crear una nueva conexión ethernet con IP por DHCP

Con el siguiente ejemplo agregamos una nueva conexión a la interfaz eth1; esta tomara la información de red IPv4 mediante DHCP al iniciar el sistema.

```
nmcli con add con-name eth1 type ethernet ifname eth1 ipv4.method auto
```

Ejemplo...

```
sololinux ~ # nmcli con add con-name eth1 type ethernet ifname eth1 ipv4.method auto
Connection 'eth1' (c02cb89f-cm85-65a2-9e53-138d56h01g1f) successfully added.
sololinux ~ #
```

Recargar la conexión de red

Por defecto, **NetworkManager** no supervisa continuamente las modificaciones en los archivos de conexión. Con el siguiente comando recargara (reload) la configuración, y tomara los cambios que hayas realizado a mano.

```
nmcli con reload
```

Modificar una conexión con el editor de nmcli

La **herramienta nmcli** pone a nuestra disposición un editor en linea de comandos muy fácil de usar. Con él puedes editar una conexión actual o crear una nueva; lo puedes ejecutar con el siguiente comando.

```
nmcli con edit
```

En la pantalla te aparecen los tipos de conexión disponibles, debes seleccionar uno.

Tipos de conexión...

Tipos de conexión válidos: generic, 802-3-ethernet (ethernet), pppoe, 802-11-wireless (wifi), wimax, gsm, cdma, infiniband, adsl, bluetooth, vpn, 802-11-olpc-mesh (olpc-mesh), vlan, bond, team, bridge, bond-slave, team-slave, bridge-slave, no-slave, tun, ip-tunnel, macvlan, vxlan

```
Ingrese el tipo de conexión: 802-3-ethernet
===[ nmcli interactive connection editor ]==

Agregar una nueva conexión '802-3-ethernet'
```

Te solicitará los datos de la conexión paso a paso.

Deshabilitar IPv6 en la conexión de Ethernet

De manera predeterminada **Ethernet** admite conexiones IPv4 y IPv6, si quieres deshabilitar las IPv6 ejecuta el siguiente comando.

```
nmcli con mod eth1 ipv6.method ignore
```

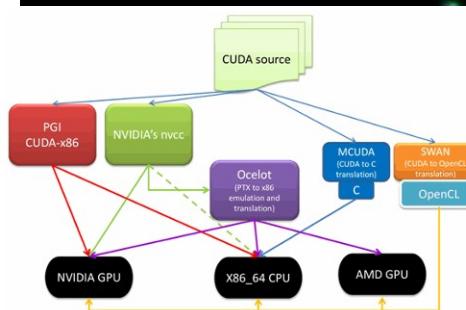
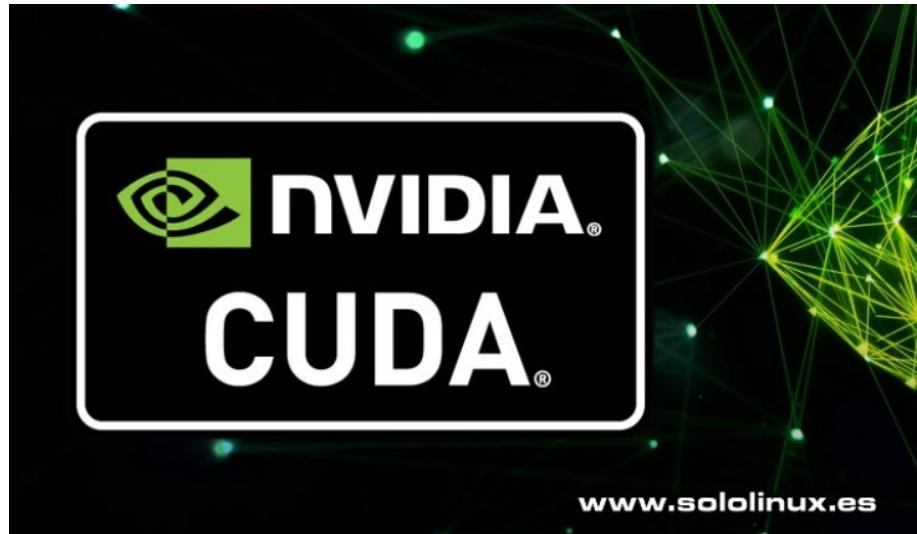
Con este comando damos por concluido el artículo «*comando nmcli con ejemplos*».

Canales de Telegram: [Canal SoloLinux](#) – [Canal SoloWordpress](#)

Instalar Cuda en Ubuntu y derivados

CUDA® es una plataforma de computación paralela desarrollado por **NVIDIA** que opera en unidades de procesos gráficos (**GPU**). Al usar este sistema, los desarrolladores aceleraran de forma contundente sus aplicaciones al aprovechar la potencia de las GPU de Nvidia.

En este artículo veremos como instalar **CUDA NVIDIA** en un sistema **Ubuntu**, pero ojo!!!, bajo tu responsabilidad. Recuerda que el manejo o modificación de drivers puede dar al traste con tu sistema operativo.



Instalar Cuda en Ubuntu y derivados

Te recomiendo que primero instales el ultimo driver de **NVIDIA**, por ejemplo:

```
wget http://es.download.nvidia.com/XFree86/Linux-x86_64/440.44/NVIDIA-Linux-x86_64-440.44.run
```

```
sudo chmod +x NVIDIA-Linux-x86_64-440.44.run
```

Ejecutamos la instalación del driver NVIDIA.

```
./NVIDIA-Linux-x86_64-440.44.run
```

Una vez termine la instalación del ultimo driver vamos a por CUDA.

```
wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/cuda-ubuntu1804.pin
```

de Ubuntu, Linux Mint y derivados.

```
sudo mv cuda-ubuntu1804.pin /etc/apt/preferences.d/cuda-repository-pin-600
```

```
wget http://developer.download.nvidia.com/compute/cuda/10.2/Prod/local_installers/cuda-repo-ubuntu1804-10-2-local-10.2.89-440.33.01_1.0-1_amd64.deb
```

```
sudo dpkg -i cuda-repo-ubuntu1804-10-2-local-10.2.89-440.33.01_1.0-1_amd64.deb
```

```
sudo apt-key add /var/cuda-repo-10-2-local-10.2.89-440.33.01/7fa2af80.pub
```

Actualizamos e instalamos cuda.

```
sudo apt-get update
```

```
sudo apt-get -y install cuda
```

Si te falta alguna librería ejecuta lo que te indico.

```
sudo apt-get install freeglut3 freeglut3-dev libxi-dev libxmu-dev
```

Para finalizar actualizamos el sistema y lo reiniciamos.

```
sudo apt full-upgrade
```

```
reboot
```

Ya tienes instalado CUDA para desarrolladores, enhorabuena.

No puedo conectar Windows con Samba



Bueno amigos, se acabaron las **fiestas** y todos volvemos al trabajo. Al regresar comienzan de nuevo las dudas y con ellas los problemas.

Precisamente ayer, contacto conmigo un fiel **lector de sololinux** que a la vuelta de sus vacaciones tenía un gran problema. Dicho usuario tiene un **servidor CentOS 6 con Samba**, a este servidor se conectan otros sistemas incluyendo maquinas con Windows XP.

Con el cambio de año se decide actualizar el vetusto CentOS 6 a CentOS 7, y cual es la sorpresa al verificar que todas la maquinas conectan a Samba sin problemas exceptuando las que montan Windows XP y algunas con Windows server.

Se verifica Samba y comprobamos que CentOS 7 viene con Samba 4.x, después de muchos chequeos nos damos cuenta que en el **archivo de configuración de Samba**, existen cambios importantes. En este articulo vemos como logramos solucionar el problema.



No puedo conectar Windows con Samba

Por que tenemos este problema?, la respuesta es sencilla y seguro que la comprendes rápidamente. A partir de la versión **Samba 4.5** (septiembre del 2016), la variable **ntlm auth** esta negada por defecto, entonces lo único que tenemos que hacer es habilitarla de nuevo.

Para solucionar el problema abrimos el archivo de configuración.

```
nano -w /etc/samba/smb.conf
```

En la sección **Global Settings...**

```
=====
 Global Settings
=====

[global]

## Browsing/Identification ##

# Change this to the workgroup/NT-domain
# name your Samba server will part of
# workgroup = WORKGROUP

# server string is the equivalent of the
# NT Description field
#         server string = %h server
# (Samba, Ubuntu)
```

Agregamos la siguiente linea.

```
ntlm auth = yes
```

Ejemplo...

```
[global]

workgroup = AGENCY
server string = Samba Server %v
security = user
passdb backend = tdbsam

ntlm auth = yes

encrypt passwords = Yes
smb passwd file =
/var/lib/samba/private/passdb.tdb

[files2]
comment = files
path = /var/www/files2
public = yes
writable = yes
printable = no
browseable = yes
available = yes
valid users = sambauser
```

Guarda el archivo y cierra el editor.

Solo nos falta **reiniciar Samba** y verificar que funciona bien.

```
systemctl restart smb
```

```
systemctl status smb
```

Nota final: si sigues teniendo problemas para conectar con Samba, puedes agregar también la siguiente variable.

```
lanman auth = yes
```

Canales de Telegram: [Canal SoloLinux](#) – [Canal SoloWordpress](#)

**VISITA NUESTRAS
WEBS:**

SolWordpress

SolLinux

Ifconfig: command not found

```
[root@ ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet  netmask 255.255.255.0 broadcast
      inet6 fe80::208:a2ff:fe0c:73e6 prefixlen 64 scopeid 0x2
        ether 00:08:a2:0c:73:e6 txqueuelen 1000 (Ethernet)
        RX packets 115186 bytes 40463043 (38.5 MiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 39 bytes 33570 (3.2 MiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions
        device memory 0x e0000-0x e000f
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0 www.sololinux.es
      inet6 ::1 prefixlen 128 scopeid 0x1<host>
        loop txqueuelen 0 (Local loopback)
        RX packets 13970 bytes 22089191 (21.0 MiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 13970 bytes 22089191 (21.0 MiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions
```

**IFCONFIG
command not found**

Hace ya tiempo que el **comando ip** reemplazo a **ifconfig**, aun así... **ifconfig** sigue siendo un comando indispensable para los que llevamos años en el mundo de los **servidores linux**, supongo que por costumbre.

A pesar de lo dicho, **ifconfig** sigue instalado por defecto en la mayoría de las **distribuciones linux** de escritorio, pero no en las minimal para servidor. Al ejecutar **ifconfig** en linux server recibimos el siguiente aviso:

Ifconfig: command not found

En este articulo veremos como identificar e instalar el comando **ifconfig** en las distros para server más populares, CentOS y derivados, Debian, Ubuntu y derivados. Los manuales oficiales ya no recomiendan su uso, algunos **sysadmin** sí, tu decides.

Ifconfig: command not found

Lo primero que hacemos es identificar el paquete que contiene la herramienta, para ello nos ayudamos de **provides**.

```
yum whatprovides ifconfig
```

Obtenemos un resultado similar a...

```
net-tools-2.0-0.25.20131004git.el7.x86_64 : Basic networking tools
Repositorio       : @anaconda
Resultado obtenido desde:
Nombre del archivo : /usr/sbin/ifconfig
```

En la respuesta que nos ofrece nuestro CentOS 7, podemos ver que el paquete que proporciona **ifconfig** es **net-tools**, así que lo instalamos.

CentOS, RHEL y derivados:

```
yum install net-tools -y
```

Debian, Ubuntu y derivados:

```
yum whatprovides ifconfig
```

Ya tenemos instalado el **comando ifconfig**.

Formatear y montar un disco de más de 2 terabytes en linux

La forma más sencilla de **formatear y montar un disco en linux** es con el **comando fdisk**, pero ten cuidado porque este comando puede producir errores con discos de más de 2 terabytes.

En este artículo de hoy, te explico de forma rápida como el otro día formatee y monté un disco de 4 terabytes sin ningún problema. Sigue los pasos que te indico para lograr el objetivo con éxito.

Formatear y montar un disco de más de 2 terabytes

Lo primero que hacemos es identificar el dispositivo, para ello si que podemos utilizar **fdisk**.

```
fdisk -l
```

Ejemplo de salida...

```
Disk /dev/sdd: 4000.8 GB, 4000887030016 bytes
255 heads, 63 sectors/track, 486401 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disk identifier: 0x00000000

Device Boot Start End Blocks Id System
/dev/sdd1 1 267350 2147483647+ ee GPT
Partition 1 does not start on physical sector boundary.
```

Bien ya lo tenemos identificado, ahora verificamos que tenemos la herramienta **parted** instalada.

```
parted --v
```



Ejemplo de salida...

```
sololinux # parted -v
parted (GNU parted) 3.2
Copyright (C) 2014 Free Software Foundation, Inc.
Licencia GPLv3+: GNU GPL versión 3 o superior
<http://gnu.org/licenses/gpl.html>.
Esto es software libre: es libre de cambiarlo y
redistribuirlo.
NO HAY NINGUNA GARANTÍA, en la medida que
lo permita la ley.
```

Ejecutamos parted en nuestro disco (con la identificación obtenida con fdisk, en nuestro ejemplo /sdd).

```
parted /dev/sdd
```

Creamos la etiqueta GPT.

```
mklabel gpt
```

Ejemplo de salida...

```
(parted) mklabel gpt
```

```
Warning: The existing disk label on /dev/sdd will be
destroyed and all data on this disk will be lost. Do
you want to continue?
```

Yes/No? yes

Ahora configuramos la unidad en terabytes.

```
mkfs.ext4 /dev/sdd1
```

Una vez formateada, agregamos la partición a **fstab** para que linux la monte automáticamente al iniciar el sistema.

```
nano -w /etc/fstab
```

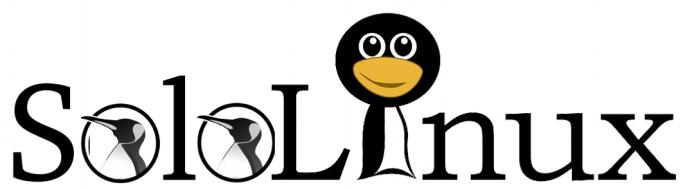
Agrega lo siguiente:

```
/dev/sdd1 /mnt/home ext4 defaults 0 2
```

Guarda el archivo y cierras el editor.

Como puedes ver creamos el punto de montaje en el /home, tu puedes establecer la ruta que quieras. Solo nos falta reiniciar el sistema.

```
reboot
```



Esta revista es de **distribución gratuita**, si lo consideras oportuno puedes ponerle precio.

Tu también puedes ayudar, contamos con la posibilidad de hacer donaciones para la REVISTA, de manera muy simple a través de **PAYPAL**

AYUDANOS A SEGUIR CRECIENDO



Síguenos en las Redes:



Uso del comando ip con ejemplos

Como ya explicamos en un [artículo anterior](#), el **comando ifconfig** esta en desuso. Actualmente tenemos el **comando ip** que es muy similar a **ifconfig** pero más potente y con más funciones.

El **comando ip** es una excelente herramienta que se utiliza en tareas de administración de red. En este artículo vemos algunos de los ejemplos de uso más habituales.



Uso del comando ip con ejemplos

El uso del comando es bastante simple, aun así antes de comenzar con los ejemplos más habituales conocemos su sintaxis, que como puedes observar es bastante simple.

```
ip [ OPTIONS ] OBJECT { COMMAND | help }
```

Información de las interfaces de red

```
ip address
```

Información de una interfaz de red

En este caso solicitamos solo los datos de la wifi, en nuestro ejemplo **wlp2s0**.

```
ip address show wlp2s0
```

Ejemplo de salida...

```
sololinux ~ # ip address show wlp2s0
3: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:13:e8:f1:b0:d7 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.4/24 brd 192.168.0.255 scope global wlp2s0
        valid_lft forever preferred_lft forever
    inet6 fe80::cf4e:2f01:b97d:480/64 scope link
        valid_lft forever preferred_lft forever
```

Estadísticas de uso de los dispositivos de red

```
ip -s link
```

Ejemplo de salida...

```
sololinux ~ # ip -s link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    RX: bytes packets errors dropped overrun mcast
      169254    1864      0      0      0
    TX: bytes packets errors dropped carrier collsns
      169254    1864      0      0      0
2: enp1s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN mode DEFAULT group default qlen 1000
    link/ether 00:03:25:30:66:95 brd ff:ff:ff:ff:ff:ff
    RX: bytes packets errors dropped overrun mcast
      0      0      0      0      0
    TX: bytes packets errors dropped carrier collsns
      0      0      0      0      0
3: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DORMANT group default qlen 1000
    link/ether 00:13:e8:f1:b0:d7 brd ff:ff:ff:ff:ff:ff
    RX: bytes packets errors dropped overrun mcast
      542666194  410802      0      0      0
    TX: bytes packets errors dropped carrier collsns
      22614502  167239      0      0      0
```

Estadísticas de uso de una interfaz en particular

Al igual que en el ejemplo anterior revisamos exclusivamente la wifi, en nuestro caso wlp2s0.

```
ip -s link show wlp2s0
```

Ejemplo de salida...

```
sololinux ~ # ip -s link show wlp2s0
3: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode
DORMANT group default qlen 1000
    link/ether 00:13:e8:f1:b0:d7 brd ff:ff:ff:ff:ff:ff
    RX: bytes   packets   errors   dropped overrun mcast
      546008090  413348     0       0       0       0
    TX: bytes   packets   errors   dropped carrier collsns
      22854139   169557     0       0       0       0
```

Ver el enrutamiento de la red

```
ip route
```

Ejemplo de salida...

```
sololinux ~ # ip route
default via 192.168.0.1 dev wlp2s0  proto static  metric 600
169.254.0.0/16 dev wlp2s0  scope link  metric 1000
192.168.0.0/24 dev wlp2s0  proto kernel  scope link  src 192.168.0.4  metric 600
```

Asignar IP a una interfaz

```
ip a add (ip_address) dev interface
```

Ejemplo...

```
ip a add 192.168.0.75/24 dev wlp2s0
```

Eliminar la IP asignada a una interfaz

```
ip a del (ip_address) dev interface
```

Ejemplo...

```
ip a del 192.168.1.50/24 dev wlp2s0
```

Habilitar una interfaz de red

```
ip link set (interface) up
```

Ejemplo...

```
ip link set wlp2s0 up
```

Deshabilitar una interfaz de red

```
ip link set (interface) down
```

Ejemplo...

```
ip link set wlp2s0 down
```

Monitorizar las interfaces de red

Con esta opción podemos monitorizar el estado de los dispositivos de red, así como sus direcciones y rutas de forma continua.

```
ip monitor
```

Ejemplo...

```
sololinux ~ # ip monitor

3: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP>
    link/ether
3: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP>
    link/ether
192.168.0.1 dev wlp2s0 lladdr c4:6e:1f:e0:f3:e4 STALE
192.168.0.1 dev wlp2s0 lladdr c4:6e:1f:e0:f3:e4 REACHABLE
3: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP>
    link/ether
```

- **STABLE**: Valido pero se requiere su verificación en la próxima transmisión.
- **REACHABLE**: Valido y accesible.
- **RELAY**: Paquete enviado correctamente pero esperando su confirmación.

Ver la MAC de los dispositivos conectados

```
ip neighbour
```

Ejemplo...

```
sololinux ~ # ip neighbour
192.168.0.1 dev wlp2s0 lladdr c4:6e:1f:e0:f3:e4 DELAY
```

Modificar las entradas ARP

Las ARP son el protocolo de resolución de direcciones en red.

Agregar una entrada ARP

```
ip neighbour add (ip_address) dev interface
```

Ejemplo...

```
ip neighbour add 192.168.0.150 dev wlp2s0
```

Eliminar una entrada ARP

```
ip neighbour del (ip_address) dev interface
```

Ejemplo...

```
ip neighbour del 192.168.0.150 dev wlp2s0
```

Ayuda del comando ip

El siguiente comando imprime en pantalla la ayuda de ip.

```
ip help
```

Unix vs Linux



Este artículo viene a colación sobre la alta confusión que existe sobre **Unix** y **Linux**, y es que estoy un poco harto de ver artículos por la red con el título «**Como instalar Unix**», cuando realmente son simples tutoriales de **como instalar Ubuntu** u otras distribuciones linux. Por dios....

No seré yo el que rebata las enseñanzas del actual profesorado, sus motivos tendrán para tal afirmación pero es errónea y ademas grave (los que llevan muchos años en este mundo saben de que hablo). Linux es un derivado de **Unix** eso está claro, pero no tienen nada que ver ni es su concepto, ni en su fin.

Antes de Linux, MacOS o Windows, Unix dominaba el mundo de la informática, y esta no sería lo que es actualmente sin la visión de **Ken Thompson y Dennis Ritchie** que anticipándose a los tiempos, y junto a otros colaboradores como Bell Labs crearon un entorno de programación colaborativo que ahora conocemos como «desarrollo abierto».

Hoy conocemos sus principales diferencias entre Unix y Linux, a ver si se disipan las dudas de muchos usuarios.

Unix vs Linux

No voy a sacar ninguna bandera a favor de **Unix** o **Linux**, son diferentes. Una cosa si puedo afirmar, los linuxeros más viejos del lugar comenzaron con derivados de Unix al estilo BSD o similares. Actualmente quedan pocas alternativas puras, podíamos hablar de **Oracle Solaris, FreeBSD** (excelente sistema operativo) y poco más que sea accesible para un usuario común (no profesional).

Bueno...vamos a dejarnos de **chachara** y vamos al artículo, jaja.

¿Qué es UNIX?

Unix es considerada como la madre de la mayoría de sistemas operativos. Su diseño se basa en la «**Filosofía Unix**»:

- Almacenamiento de datos en texto plano.
- Sistema de archivos jerárquico.
- Manejo de dispositivos (también otros específicos de comunicación) entre procesos (IPC) como archivos.
- Uso de muchas herramientas de software.
- Aplicaciones pequeñas, simples y modulares que se puedan fusionar con un intérprete en línea de comandos.
- No a los programas monolíticos, son pesados y no permiten su diversificación.

Cita sobre la filosofía de Unix:

“Nuestra filosofía no se puede escribir en una sola frase, ya que su principio es la idea de que el poder de un sistema proviene más de las relaciones entre los programas que de los propios programas. Muchas aplicaciones de UNIX hacen su trabajo de forma aislada, pero combinadas con otras se convierten en herramientas muy útiles incluso para usos generales.” – Brian Kernighan y Rob Pike.

Arquitectura de Unix

El **Kernel de Unix** tiene el control total sobre el sistema. Tiene otros subsistemas que manejan los servicios como el sistema de archivos, los recursos, la administración de memoria, aplicaciones de inicio y otras tareas básicas de bajo nivel.

El núcleo es el corazón del sistema operativo, y ejerce como si fuera una interfaz entre el usuario y el **hardware**. Cada subsistema del kernel tiene sus propias características como concurrencia, memoria virtual, paginación y su sistema de archivos virtual.

En las capas externas de la arquitectura Unix, nos encontramos con la **shell**, sus comandos y herramientas. La **shell** (mal llamada terminal o consola) es la interfaz entre el usuario y el núcleo; cuando el usuario introduce comandos, estos son interpretados y llaman a las herramientas o aplicaciones que corresponda.

Ejemplo del sistema operativo Unix es Solaris y HP-UX. Los mayores distribuidores de sistemas UNIX incluyen IBM, HP y SUN.

¿Qué es Linux?

Como ya dijimos anteriormente se confunden los términos Unix y Linux; muchos usuarios se hacen las siguientes preguntas:

- ¿Linux y Unix son lo mismo?
- ¿Cuales son las diferencias entre Unix y Linux?
- ¿Linux es como Unix?
- ¿Linux basado en Unix?

Estas preguntas se pueden responder en una sola linea...

Linux y Unix son diferentes pero tienen similitudes, recuerda que Linux es un clon de Unix.

Linux no es Unix, pero si es similar, podríamos decir que es una continuación de la base del diseño de Unix. Las muchas y excelentes **distribuciones Linux** son el mejor ejemplo de ello, **BSD** (Berkeley Software Distribution) y su derivado **FreeBSD** también es otro buen ejemplo (aunque derivo por otro camino).

Podemos decir que un sistema operativo de tipo Unix, es el que funciona de manera similar a los Unix puros sin que sea necesario que cumpla con la especificación **UNIX (SUS)**, o **POSIX** (interfaz portátil del sistema operativo). De todas formas si se quiere que **The Open Group** lo certifique como Unix, debe cumplir el standard.

Pocos sistemas UNIX están registrados como tal, podemos hablar de macOS, Solaris y AIX. Y si tenemos en cuenta POSIX, Linux lo consideramos como un sistema operativo parecido a Unix, pero no Unix

Si leemos el [archivo readme del kernel Linux](#), nos dice que **Linux es un clon de UNIX** desarrollado desde cero por **Linus Torvalds** y su equipo. **OJO!!!**, este texto es solo para cumplir el standard POSIX, porque si el kernel Linux fue escrito desde cero y se diseño para que trabajara como Unix, pero no contiene el código original de Unix... no lo podríamos considerar como derivado de Unix.

Otro detalle importante que muchos pasan por alto y que deriva en equivocaciones, es que **Linux solo es el núcleo y no el sistema operativo completo**. Otros desarrolladores empaquetan el kernel Linux en sus distribuciones, ahora si que se considera un sistema operativo completo.

No lo olvides, Linux es solo el Kernel, solo las distribuciones Linux se consideran como sistemas operativos. Si lo comparamos con UNIX, este si es un sistema operativo completo pues todo proviene del mismo desarrollador.

En resumen, una distribución Linux es un sistema operativo creado sobre el kernel de Linux al que se adjuntan varias aplicaciones entre las que se encuentra el sistema de administración de paquetes.

Una distribución estándar Linux consta de un kernel Linux, un sistema GNU, utilidades GNU, bibliotecas, compiladores, software adicional, documentación, un sistema de ventanas, un administrador de ventanas y un entorno de escritorio. Normalmente todo es gratuito y de **código abierto**.



Diferencias entre UNIX y Linux

Diferencias de concepto

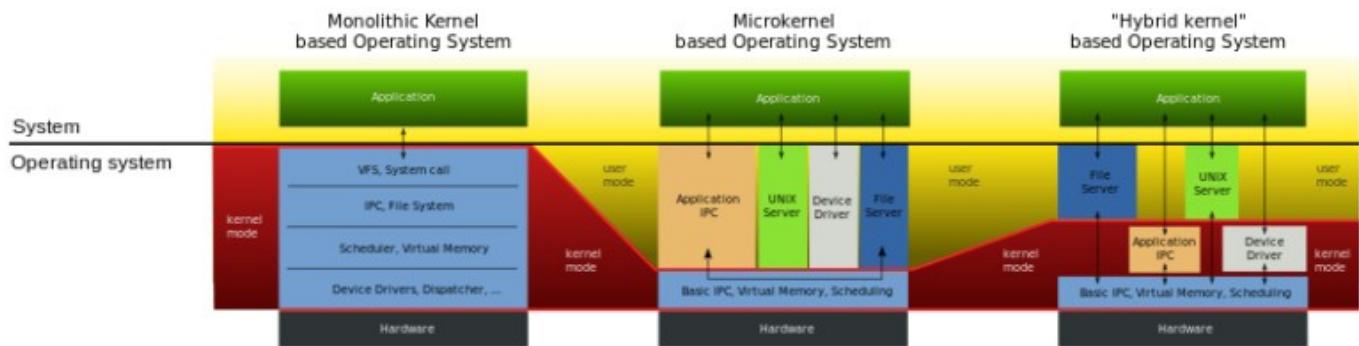
| UNIX | LINUX |
|---|--|
| Sistema operativo original desarrollado por AT&T | Núcleo del sistema operativo GNU / Linux |
| Código original desarrollado por AT&T | Código original desarrollado por Linus y la Fundación GNU |
| La marca UNIX está certificada por Open Group | La marca Linux es propiedad de Linus Torvalds y es administrada por Linux Mark Institute de la Fundación Linux |
| System-V Unix y derivados como IBM-AIX y HP-UX; Berkeley Unix y derivados como FreeBSD y macOS | GNU / Linux y derivados como Debian o Fedora |
| Unix es parcialmente de código abierto con la Licencia BSD. La fuente de System-V Unix tiene licencia comercial patentada | Código abierto con licencia copyleft |
| Los derivados son mantenidos por empresas incluyendo el núcleo | Las distros son mantenidas por la comunidad aplicando el kernel de Linux Torvalds |
| Diseñado para un público minoritario y para un conjunto definido de plataformas y aplicaciones | Diseñado como plataforma escalable de uso general |
| Soporte con coste | Soporte comunitario gratuito / Soporte con coste de proveedores de servicios |
| Interfaces muy estables | Interfaces que evolucionan constantemente con el peligro de inestabilidad u otros problemas derivados |
| Pocas actualizaciones, no son necesarias | Debido a la rápida evolución de las herramientas, las actualizaciones son frecuentes |
| Admite pocos sistemas de archivos | Admite la mayoría de sistemas de archivos |
| Cada variante tiene su potente herramienta de administración de sistema | Las herramientas de administración de sistema son muy limitadas y a veces de terceros |
| Indicada para servidores y usuarios con altos conocimientos | Valida para la mayoría de entornos |
| Las herramientas en línea de comandos y las utilidades gráficas son similares a Linux | Las herramientas en línea de comandos y las utilidades gráficas son similares a Unix |

Diferencias de desarrollo

| | UNIX | LINUX |
|---|--|---|
| Desarrollo | Desarrollado en Bell Labs por Kenneth Lane Thompson, Dennis Ritchie y otros apoyos | Inspirado en MINIX (similar a Unix), Linux fue desarrollado por Linus Torvalds. Al ser de código abierto muchos programadores colaboran con la comunidad linux |
| Escrito en | C y lenguaje ensamblador | C y otros lenguajes de programación |
| OS family | Unix | Unix-like |
| Estado del desarrollo | Actual | Actual |
| Tipo de fuente | Normalmente de código cerrado. Existen sistemas operativos Unix open source como illumos y BSD; FreeBSD, Solaris, OpenIndiana, OmniOS, etc | Open source |
| Lenguaje | Ingles | Muchos idiomas |
| Primer lanzamiento | Fue lanzado externamente en octubre de 1973, pero se venia utilizando en Bell Labs desde 1970 | Septiembre de 1991 |
| Tipo de kernel | Existen versiones monolíticas pero como norma general se compone de varios microkernel | Kernel monolítico (al ser monolítico se vuelve muy pesado en maquinas antiguas) |
| Tipo de licencia | Puede variar, algunas son propietarias y otras free/OSS | GNUv2 (GPL General Public License) |
| Sitio web oficial | https://www.opengroup.org/membership/forums/platform/unix | https://www.kernel.org/ |
| Interfaz de usuario predeterminada | CLI (Command Line Interface) and Graphical (X Windows system) | Unix shell |
| Interfaz en modo texto | Originalmente Bourne shell, pero es compatible con otros interpretes de comandos | Por defecto BASH, pero es compatible con otros interpretes de comandos |
| Coste (\$ / €) | Depende de la empresa propietaria. Ya vimos en esta tabla que también existen versiones open source | Lo puedes conseguir y usar libremente. Existen distribuciones con añadidos específicos, normalmente empresariales que si tienen un coste |
| Algunos ejemplos | IBM AIX, Solaris, HP-UX, Darwin, macOS X, etc... | Debian, Ubuntu, Fedora, Red Hat, Android, etc... |
| Detección y solución de errores | Los errores son poco comunes, si se da el caso, y como suelen ser sistemas propietarios deberás esperar el parche correspondiente | Linux está impulsado la comunidad open, muchos desarrolladores de todo el mundo trabajan continuamente en el código. La detección y solución de errores es bastante rápida |
| Seguridad | Unix se considera muy seguro. Es aún más difícil de infectar que linux. Actualmente no hay virus en propagación activos | Linux se considera muy bien protegido contra el malware. Esto es por la falta de acceso a la raíz, y las rápidas actualizaciones. Desde 2018, no se ha extendido ningún nuevo virus |

El Kernel: Unix vs Linux

Como podemos apreciar en la siguiente imagen, existen tres tipos de kernel: monolítico, micro e híbrido (combinación de monolítico y micro).



En una arquitectura de núcleo monolítico, todo el sistema operativo funciona desde un único núcleo. Este sistema es el más extendido pero tiene la pega de requerir un hardware más potente, pues utiliza una interfaz virtual de alto nivel.

El núcleo de Linux tiene características similares a los de Unix, pero existen puntos que los diferencian en gran medida.

En la arquitectura de microkernel, los servicios centrales del sistema se ejecutan en un solo proceso, el resto de servicios en diferentes procesos. El tamaño del código fuente de los micro es mucho menor, eso repercute de manera considerable en un rendimiento superior.

Características del Kernel

| | Kernel Unix | Kernel Linux |
|------------------------------------|--|--|
| Tipo | El núcleo de Unix puede ser monolítico, microkernel o híbrido, depende del sistema elegido | Monolítico |
| Agregar o eliminar características | El kernel Unix necesita la vinculación estática de los nuevos sistemas que se agregan | Los componentes del núcleo, las unidades de dispositivo, se pueden agregar y eliminar dinámicamente como módulos. Esta característica se conoce como módulos de núcleo cargables (LDM) |
| Streams | En Unix se incluye el subsistema E/S de secuencias como interfaz para escribir controladores de dispositivo, controladores de terminal, etc... | En Linux, no hay subsistemas E/S de secuencias |
| Sistema preventivo | Solo algunos son preventivos, como por ejemplo el de Solaris | Normalmente no |
| Subprocesos múltiples (LWP) | En Unix, el LWP se basa en hilos del kernel | En Linux, LWP se crea llamando a la función clone (). Estos procesos pueden compartir memoria física, archivos abiertos, espacio de direcciones, etc... |
| Hilos del Kernel | Los hilos se utilizan para cambiar los procesos | Solo para ejecutar códigos del kernel de forma periódica |

Comparamos Unix con Linux

Las comparaciones son odiosas, pero una vez que ya conocemos las principales diferencias entre Unix y Linux creo que es necesario.

Linux vs Solaris

- Solaris (**Oracle Solaris**) es un sistema operativo de la familia Unix, lo comparamos con Linux.
- Linux admite más arquitecturas de sistema que Solaris. Linux es más portátil.
- En los últimos años linux hace un gran esfuerzo en la integración del nuevo hardware, Solaris no es tan rápido.
- La estabilidad de Solaris es algo sorprendente, le das con un martillo y no se rompe, ja.

Existen otras diferencias técnicas entre los dos, pero aquí solo hemos hablado de uso y rendimiento.

Conclusión final

Unix es el padre de todos los sistemas operativos actuales eso está claro. La principal diferencia entre los dos no es su presentación, sino en cómo funcionan, principalmente el núcleo.

Linux (y otros sistemas tipo Unix) son libres, los sistemas Unix no lo son (salvo excepciones ya mencionadas). El coste se debe tener en cuenta, y linux en este tema es el mejor.

También destacamos que Linux es más flexible si lo comparamos con los Unix puros, y es uno de los detalles que hace que linux sea tan popular. Los comandos Unix y Linux, no son los mismos pero si muy similares, es como si hablamos de Debian y RHEL.

Después de este artículo me gustaría que se empiecen a corregir informaciones erróneas como esta...

Unix sysadmin since 1991 dice: Ubuntu es una buena opción para comenzar con Unix, ya que es muy fácil de instalar y viene con una buena GUI también.

O como esta otra...

Pasos para instalar Unix: Instale Red Hat Enterprise Linux 7 de 64 bits (RHEL), descargue RHEL desde



Esta revista es de **distribución gratuita**, si lo consideras oportuno puedes ponerle precio.
Tu también puedes ayudar, contamos con la posibilidad de hacer donaciones para la REVISTA, de manera muy simple a través de **PAYPAL**

AYUDANOS A SEGUIR CRECIENDO



Síguenos en las Redes:



Novedades de Mozilla Firefox 72



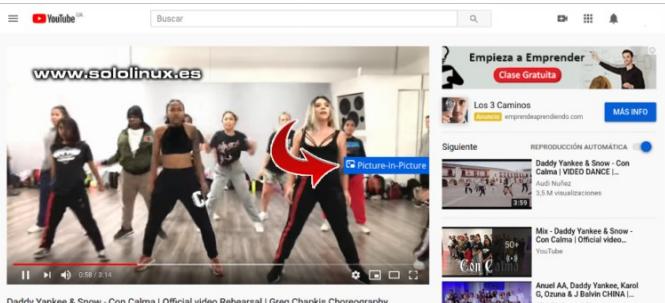
Ante el inminente lanzamiento de **Microsoft Edge 79** (basado en **Chromium**), así como un creciente interés en otros navegadores como **Vivaldi** o **Brave**, **Mozilla Firefox** a reaccionado, y vaya trabajo se han marcado.

Después de probar intensamente la nueva versión de **Firefox Browser 72**, puedo decir que estamos ante un punto y aparte del **navegador web** de Mozilla. En el artículo de hoy veremos las características más importantes, así como la corrección de errores.

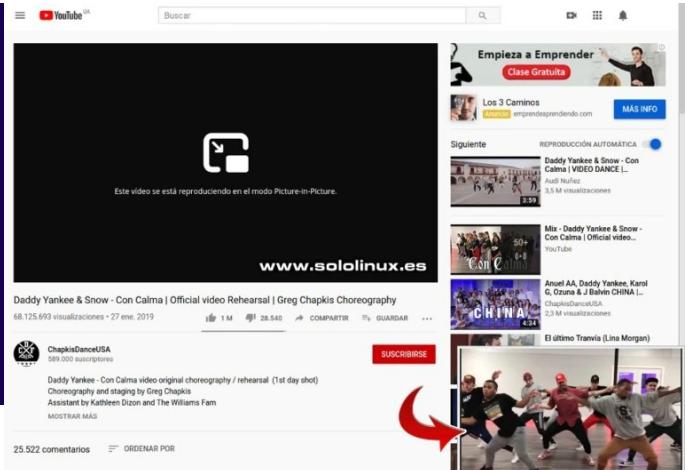
Novedades de Mozilla Firefox 72

La nueva función agregada a Mozilla Firefox linux «**Picture and Picture**», me parece un gran acierto. Aporta soporte multipantalla y un ahorro de consumo del sistema importante. Esta función abre una nueva pantalla de visualización en la parte inferior derecha de nuestro monitor, la puedes ampliar y mover.

Picture and Picture es especialmente útil para visionados en sitios como YouTube o Netflix. Al iniciar la reproducción aparece una pestaña de color azul a la derecha...



Al pulsar el botón se abre una nueva pantalla de visionado.



Otras características a destacar en **Firefox 72** son la **protección de seguimiento mejorada**, y un manejo excelente de las notificación push (muy molestas a mi parecer). Ahora oculta las solicitudes de inserción web en un ícono burbuja de diálogo (azul) en la barra de direcciones.

Las «huellas digitales», que es donde los sitios por donde navegas intentan identificar los datos de tu dispositivo a través de las API legítimas del navegador, ahora están bloqueadas.

A nivel de css, el equipo de Mozilla también a realizado un buen trabajo, incluyendo entre otras cosas **Motion Path** que permite animar un elemento gráfico a lo largo de una ruta personalizada.

Las correcciones en seguridad han sido varias, como por ejemplo la vulnerabilidad crítica 'zero day' identificada como **CVE-2019-17026** y que es explotada 'in the wild'. Otras importantes son **CVE-2019-17016**, **CVE-2019-17017**, **CVE-2019-17024** y **CVE-2019-17025**.

Otras vulnerabilidades menores son desde la **CVE-2019-17018** a la **CVE-2019-17023**, y alguna que otra más exclusiva del rey de los parches, Windows.

A todo lo dicho debemos sumar la velocidad con la que se mueve el navegador, a diferencia de otros que cada día son más pesados, Firefox no.

Puedes descargar la ultima versión desde su [pagina oficial](#).

Verificar que tenemos habilitado http2 en nuestra web

Todos conocemos los beneficios del **protocolo HTTP2** con respecto a **HTTP1**, pero estas seguro que tu sitio web lo tiene habilitado?.

El **protocolo http2** se habilita a nivel de servidor, y te puedo asegurar que existen servidores y VPS que aun continúan utilizando el vetusto **http1**, seguro que más de los que crees.

Hoy vemos un sencillo **script** escrito en **python**, que nos dirá si la url que insertemos trabaja con el protocolo HTTP2, o continua con HTTP1. El único requisito es que tengas instalado **python 3**.

Verificar que tenemos habilitado http2

Creamos el script py.

```
nano http2.py
```

Copia y pega lo siguiente:

```
import socket
import ssl
import csv
import argparse

from urllib.parse import urlparse

socket.setdefaulttimeout(5)

headers = {"user-agent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36"}

def check_http2(domain_name):
    try:
        HOST = urlparse(domain_name).netloc
        PORT = 443

        ctx = ssl.create_default_context()
        ctx.set_alpn_protocols(['h2', 'spdy/3', 'http/1.1'])

        conn = ctx.wrap_socket(
            socket.socket(socket.AF_INET, socket.SOCK_STREAM),
            server_hostname=HOST)
        conn.connect((HOST, PORT))

        pp = conn.selected_alpn_protocol()

        if pp == "h2":
            return {"http2": True}
        else:
            return {"http2": False}
    except Exception as e:
        print(e)

parser = argparse.ArgumentParser()
parser.add_argument("domain", help="display a square of a given number",
                    type=str)
args = parser.parse_args()

print(check_http2(args.domain))
```



Guarda el archivo y cierra el editor.

La sintaxis de uso es muy sencilla (recuerda que debes tener instalado Python 3).

```
python3 [SCRIPT.py] [URL]
```

Verificamos una url como ejemplo.

```
python3 http2.py https://www.sololinux.es
```

La respuesta nos dice que tenemos el protocolo http2 habilitado.

```
sololinux ~ # python3 http2.py
https://www.sololinux.es
{'http2': True}
```

Si seguimos con el protocolo HTTP1 la respuesta será **False** en vez de **True**.

Monitorizar el uso de la CPU y avisar por mail

Al igual que en otras labores, los sitios webs también evolucionan. Vamos aumentando las visitas, agregamos nuevas herramientas o **scripts** a los servidores, mejoramos nuestros sitios, etc...

Todos los añadidos que apliquemos repercuten en el consumo del sistema, y normalmente el componente que más lo sufre es la CPU.

En este artículo de hoy, presentamos un interesante **script bash** que monitoriza el uso de la CPU y nos envía un correo electrónico informándonos de su estado.

Monitorizar el uso de la CPU y avisar por mail

Creamos el script bash.

```
nano /opt/scripts/alerta-cpu.sh
```

Copia y pega el siguiente script (inserta un email valido):

```
#!/bin/bash
cpuuse=$(cat /proc/loadavg | awk '{print $1}')
if [ "$cpuuse" > 80 ]; then
SUBJECT="ATTENTION: CPU Load Is High on $(
hostname) at $(date)"
MESSAGE="/tmp/Mail.out"
# your valid email
T0="mail@midominio.com"
echo "CPU Current Usage is: $cpuuse%" >>
$MESSAGE
echo "" >> $MESSAGE
echo "+-----+" >> $MESSAGE
echo "Top CPU Process Using top command" >>
$MESSAGE
echo "+-----+" >> $MESSAGE
echo "$(top -bn1 | head -20)" >> $MESSAGE
echo "" >> $MESSAGE
echo "+-----+" >> $MESSAGE
echo "Top CPU Process Using ps command" >>
$MESSAGE
echo "+-----+" >> $MESSAGE
echo "$(ps -eo pcpu,pid,user,args | sort -k 1 -r | head -10)" >> $MESSAGE
mail -s "$SUBJECT" "$T0" < $MESSAGE
rm /tmp/Mail.out
fi
```

Guarda el archivo y cierra el editor.

Ahora creamos una **tarea cron** que ejecute el script cada 10 minutos.

```
crontab -e
```

Agregamos la tarea.

```
*/10 * * * * /bin/bash
/opt/scripts/alerta-cpu.sh
```

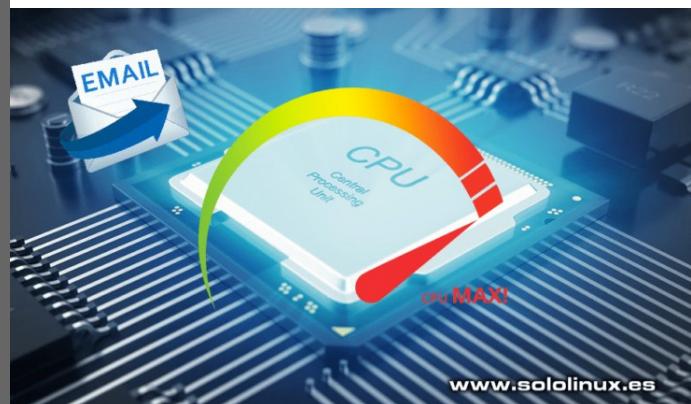
Guarda la tarea y cierra el editor.

Reiniciamos las tareas.

```
# Debian, Ubuntu, y derivados
sudo service cron restart
# CentOS, RHEL, y derivados
systemctl restart crond.service
```

El script se ejecutara automáticamente cada 10 minutos.

Síguenos en las Redes:



www.sololinux.es

Simular comandos en linux



En linux tenemos todo tipo de herramientas y aplicaciones, y como no podía ser menos, también nos beneficiamos de una opción que nos permite ejecutar comandos en nuestra **terminal** sin que realmente se lleve a lanzar la orden.

Este efecto lo conocemos como simulación, y es muy útil a la hora de averiguar que pasará en nuestro sistema al ejecutar un comando. En este artículo vemos un ejemplo de simulación de los dos comandos más utilizados, está claro que hablamos del **comando install** y del **comando remove**.

Simular comandos en linux

Comenzamos con el ejemplo en Debian, Ubuntu, Linux Mint y derivados. La orden se ejecuta con la siguiente sintaxis:

```
sudo apt [comando] [opcional] --simulate
```

Como ves su uso es bastante simple, ahora simulamos (como ejemplo) la instalación del reproductor VLC.

```
sudo apt install vlc --simulate
```

Ejemplo de salida...

```
sololinux ~ # sudo apt install vlc --simulate
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libbasicusageenvironment1 libcddb2 libdirectfb-
  1.2-9 libdvbpsi10 libebml4v5 libgroupsock8 libiso9660-
  8 liblivemedia50
    libmatroska6v5 libmpcdec6 libproxy-tools
libqt5x11extras5 libresid-builder0c2a libsdl-image1.2
libsidplay2v5 libssh2-1
  libupnp6 libusageenvironment3 libva-drm1
libvcdinfo0 libvlc5 libvlccore8 libvncclient1 libxcb-
composite0 libxcb-xv0
  vlc-data vlc-nox
```

Paquetes recomendados:

```
  vlc-plugin-notify vlc-plugin-samba
Se instalarán los siguientes paquetes NUEVOS:
  libbasicusageenvironment1 libcddb2 libdirectfb-
  1.2-9 libdvbpsi10 libebml4v5 libgroupsock8 libiso9660-
  8 liblivemedia50
```

```
  libmatroska6v5 libmpcdec6 libproxy-tools
libqt5x11extras5 libresid-builder0c2a libsdl-image1.2
libsidplay2v5 libssh2-1
  libupnp6 libusageenvironment3 libva-drm1
libvcdinfo0 libvlc5 libvlccore8 libvncclient1 libxcb-
composite0 libxcb-xv0 vlc
  vlc-data vlc-nox
```

0 actualizados, 28 nuevos se instalarán, 0 para eliminar y 1 no actualizados.

Sololinux ~ #

Si queremos comprobar que archivos se borraran del sistema al desinstalar una herramienta...

```
sudo apt remove thunderbird --simulate
```

Ejemplo de salida...

```
sololinux ~ # sudo apt remove thunderbird --simulate
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los siguientes paquetes se ELIMINARÁN:
  thunderbird thunderbird-locale-es thunderbird-
locale-es-es
0 actualizados, 0 nuevos se instalarán, 3 para eliminar y 1 no actualizados.
Remv thunderbird-locale-es-es [1:60.9.0+build1-
Oubuntu0.18.04.2]
Remv thunderbird-locale-es [1:60.9.0+build1-
Oubuntu0.18.04.2]
Remv thunderbird [1:60.9.0+build1-0ubuntu0.18.04.2]
Sololinux ~ #
```

No te preocupes, solo hemos simulado una instalación y una desinstalación, nuestro sistema sigue intacto jajaj. Se permite acortar **--simulate** por **-s**, pero creo que mejor usamos la opción normal, así sabemos lo que estamos haciendo.

En **Centos**, **RHEL** y derivados la cosa cambia, las opciones también. Vemos su sintaxis.

```
yum [comando] --setopt tsflags=test
[herramienta]
```

En este ejemplo simulamos la instalación de nano, esta operación es incluso más realista que con **--simulate** pero con el mismo resultado, todo simulado.

```
yum install --setopt tsflags=test nano
```

Si queremos simular un paquete ***.rpm**.

```
sudo rpm -ivh --test nano-2.9.8-
1.el8.x86_64.rpm
```

Ahora ya sabes como afectará a tu sistema la ejecución de un comando, puedes operar en consecuencia.

Información del sistema con Python

Python es uno de los **lenguajes de programación** más populares, y no es por casualidad. Sencillo, fácil de usar, con muchos módulos, y lo más importante, muy potente.

En el artículo de hoy nos aprovechamos de la librería **Platform de Python**, que dependiendo de la solicitud nos devuelve un resultado de la plataforma del **sistema con Python**. Vemos varios ejemplos de uso y también un script que nos aportara datos interesantes.

Información del sistema con Python

Antes de comenzar un pequeño detalle... Platform funciona en Python 2 y en Python 3, por lo tanto recuerda que si tienes la versión 2 debes ejecutar «**python**» y si tienes la versión 3 «**python3**» (en los ejemplos de uso, utilizamos python3). Comenzamos.

Sistema operativo, versión, y nombre en clave

```
python3 -c 'import platform;
print(platform.linux_distribution())'

o

python -c 'import platform;
print(platform.linux_distribution())'
```

Ejemplo de salida...
(‘CentOS Linux’, ‘7.7.1908’, ‘Core’)

Arquitectura del sistema

```
python3 -c 'import platform;
print(platform.architecture())'
```

Tipo de maquina

```
python3 -c 'import platform;
print(platform.machine())'
```

Hostname

```
python3 -c 'import platform;
print(platform.node())'
```

Plataforma base del sistema (tipo de kernel)

```
python3 -c 'import platform;
print(platform.platform())'
```

Ejemplo de salida...
Linux-3.10.0-1062.9.1.el7.x86_64-x86_64-with-centos-7.7.1908-Core

Tipo de procesador

```
python3 -c 'import platform;
print(platform.processor())'
```



Versión del kernel

```
python3 -c 'import platform;
print(platform.release())'
```

Tipo de sistema operativo

```
python3 -c 'import platform;
print(platform.system())'
```

Versión del sistema operativo

```
python3 -c 'import platform;
print(platform.version())'
```

Información del sistema

```
python3 -c 'import platform;
print(platform.uname())'
```

Versión de Python instalada

```
python3 -c 'import platform;
print(platform.python_version())'
```

Script: Información del sistema con Python

También podemos crear una secuencia de comandos Python a modo de script, que nos muestre información de nuestro sistema Linux.

Creamos el script.

```
nano infosis.py
```

Copia y pega lo siguiente:

```
import platform
print('Uname:', platform.uname())
print()
print('Distribution :',
platform.linux_distribution())
print('Machine :', platform.machine())
print('Node :', platform.node())
print('Processor :', platform.processor())
print('Release :', platform.release())
print('System :', platform.system())
print('Version :', platform.version())
print('Platform :', platform.platform())
```

Guarda el archivo y cierra el editor.

Ejecutamos con...

```
python infosys.py
o
python3 infosis.py
```

Ejemplo de salida...

```
[root@host ~]# python infosis.py
('Uname:', ('Linux', 'host.server.es', '3.10.0-1062.9.1.el7.x86_64', '#1 SMP Fri Dec 6 15:49:49 UTC
2019', 'x86_64', 'x86_64'))
()
('Distribution :', ('CentOS Linux', '7.7.1908', 'Core'))
('Machine :', 'x86_64')
('Node :', 'host.server.es')
('Processor :', 'x86_64')
('Release :', '3.10.0-1062.9.1.el7.x86_64')
('System :', 'Linux')
('Version :', '#1 SMP Fri Dec 6 15:49:49 UTC 2019')
('Platform :', 'Linux-3.10.0-1062.9.1.el7.x86_64-x86_64-with-centos-7.7.1908-Core')
[root@host ~]#
```

Nota final: a partir de Python 3.8, «platform.linux_distribution ()» ya no estará disponible.



Esta revista es de **distribución gratuita**, si lo consideras oportuno puedes ponerle precio.
Tu también puedes ayudar, contamos con la posibilidad de hacer donaciones para la REVISTA, de manera muy simple a través de **PAYPAL**

AYUDANOS A SEGUIR CRECIENDO



Síguenos en las Redes:



Monitorizar el uso del disco y avisar por mail



Existen muchas aplicaciones que monitorizan nuestros **sistemas Linux**, lo controlan todo, por ejemplo, el uso de la CPU, de la memoria, la swap, el disco, y más.

No siempre es necesario tener este tipo de herramientas instaladas, pongamos el caso que tenemos un servidor exclusivo para **backups**; en este caso, con un simple script que nos vigile no sobrepasar el umbral de uso del disco es suficiente.

Hoy vemos un **script bash** que mediante una **tarea cron** supervisara el umbral establecido del disco, y nos enviara un email para tomar las medidas necesarias.

Monitorizar el uso del disco y avisar por mail

Creamos el script bash.

```
nano /opt/scripts/alerta-disco.sh
```

Copia y pega el siguiente script, pero atento a las opciones.

- **if [\$used -ge 75]** : El umbral de uso está establecido en el 75%, modifica según tus necesidades.
- **mail@dominio.com** : inserta tu correo electrónico.

```
#!/bin/sh
df -Ph | grep -vE '^Filesystem|tmpfs|cdrom'
| awk '{ print $5,$1 }' | while read output;
do
  echo $output
  used=$(echo $output | awk '{print $1}' |
sed s/%//g)
  partition=$(echo $output | awk '{print
$2}')
  if [ $used -ge 75 ]; then
    echo "The partition \"$partition\" on $
(hostname) has used $used% at $(date)" |
mail -s "Disk Space Alert: $used% Used On $(
hostname)" mail@dominio.com
  fi
done
```

Guarda el archivo y cierra el editor.

Ahora creamos una tarea cron que ejecute el script cada 15 minutos.

```
crontab -e
```

Añade la siguiente linea.

```
*15 * * * * /bin/bash
/opt/scripts/alerta-disco.sh
```

Guarda el archivo y cierra el editor, solo nos falta reiniciar cron.

```
# Debian, Ubuntu, y derivados
sudo service cron restart
```

```
# CentOS, RHEL, y derivados
systemctl restart crond.service
```

Síguenos en las Redes:



Instalar y configurar Glances en linux



www.sololinux.es

Si como yo, eres de HTOP; la herramienta de monitoreo **Glances** te va a encantar. Con una estructura visual similar a **HTOP**, esta aplicación basada en **Python** te cautivará.

Aprovechándose de las bondades de Python, Glances nos proporciona mucha información del sistema de forma amena y fácilmente comprensible. Procesos de ejecución E / S, memoria, etc. Todo tiene cabida en Glances.

Glances no solo permite monitorizar nuestro sistema a través de la terminal, también podemos usarlo en remoto a través de su potente API o incluso en otras interfaces como **Grafana** o **Prometheus**. En este artículo, veremos cómo instalar y usar Glances para monitorear sistemas locales y remotos.

Instalar y configurar Glances en linux

Glances viene en los repositorios de la mayoría de distribuciones linux actuales, por ejemplo en Ubuntu y derivados lo puedes instalar con...

```
sudo apt install glances
```

También puedes ejecutar su propio instalador.

```
wget -O- https://bit.ly/glances | bash
```

```
o
```

```
curl -L https://bit.ly/glances | bash
```

Una vez instalado ejecuta el siguiente comando para iniciar la herramienta.

```
glances
```

Te mostrara información sobre tu sistema, en un formato similar al de la imagen que tienes a continuación.

Puedes cambiar la visualización con las teclas 1 y 3, con la tecla 2 ocultas el apartado de la izquierda. También puedes filtrar procesos de una forma tan simple como pulsar la tecla enter, y escribir el proceso.

Glances nos ofrece varias opciones con un solo comando, para verlos todos pulsa la tecla h.

```
Glances 2.3 with PSUtil 3.4.2
Configuration file: /etc/glances/glances.conf

a Sort processes automatically
b Bytes or bits for network I/O
c Sort processes by CPU%
l Show/hide alert logs
m Sort processes by MEM%
w Delete warning alerts
p Sort processes by name
x Delete warning and critical alerts
i Sort processes by I/O rate
l Global CPU or per-CPU stats
t Sort processes by CPU times
h Show/hide this help screen
d Show/hide disk I/O stats
T View network I/O as combination
f Show hide filesystem stats
v View cumulative network I/O
n Show/hide network stats
F Show filesystem free space
s Show/hide sensors stats
g Generate graphs for current history
z Show/hide left sidebar
r Reset history
e Enable/disable processes stats
q Quit (Esc and Ctrl-C also work)
D Enable/disable top extended stats
/ Enable/disable short processes name
E Enable/disable Docker stats

ENTER: Edit the process filter pattern
```

Modo cliente / servidor

Para monitorizar un sistema linux en modo remoto debes instalar Glances en los dos sistemas, local y remoto. Una vez instalado en los dos puntos, ejecuta en el sitio remoto el siguiente comando.

glances -s

En el sistema local...

```
glances -c ip-remota #<<<---ip del  
sistema remoto
```

Glances es tan potente que también nos ofrece un sistema de monitorización si tener que instalar nada en nuestra maquina local. El mismo nos brinda su propio servidor web para poder acceder desde cualquier navegador. Para ejecutar esta opción lanza en el sistema remoto el siguiente comando.

glances -w

Ahora en tu navegador local preferido insertamos la ip remota, el puerto definido (61208), y los segundos de actualización.

Ejemplo...

<http://192.168.100.101:61208/5>

Configurar Glances

Puedes configurar la herramienta a través del archivo `glances.conf`. Los nuevos archivos de configuración los debes crear de forma manual, por ejemplo.

```
nano /etc/glances
```

Copia y pega lo siguiente.

user_careful=50

```
user_warning=0  
user_critical=90  
iowait_careful=50  
iowait_warning=70  
iowait_critical=90  
system_careful=50  
system_warning=70  
system_critical=90  
steal_careful=50  
steal_warning=70  
steal_critical=90
```

Puedes modificar los valores según tus preferencias, o personalizar otras secciones. Te dejo una plantilla de configuración completa para que puedas trabajar con ella.

```
#####
# Globals Glances parameters
#####

[global]
# Does Glances should check if a newer version is available on PyPI ?
check_update=true
# History size (maximum number of values)
# Default is 28800: 1 day with 1 point every 3 seconds
history_size=28800

#####
# User interface
#####

[outputs]
# Theme name for the Curses interface: black or white
curse_theme=black
# Limit the number of processes to display in the WebUI
max_processes_display=30

#####
# plugins
#####

[quicklook]
# Set to true to disable a plugin
# Note: you can also disable it from the command line (see --disable-plugin <plugin_name>)
disable=False
# Graphical percentage char used in the terminal user interface (default is |)
percentage_char=|
# Define CPU, MEM and SWAP thresholds in %
cpu_careful=50
cpu_warning=70
cpu_critical=90
mem_careful=50
mem_warning=70
mem_critical=90
swap_careful=50
swap_warning=70
swap_critical=90

[cpu]
disable=False
# Default values if not defined: 50/70/90 (except for iowait)
user_careful=50
user_warning=70
user_critical=90
#user_log=False
#user_critical_action=echo {{user}} {{value}} {{max}} > /tmp/cpu.alert
system_careful=50
system_warning=70
system_critical=90
steal_careful=50
steal_warning=70
steal_critical=90
#steal_log=True
# I/O wait percentage should be lower than 1/# (Logical CPU cores)
# Leave commented to just use the default config (1/#-20% / 1/#-10% / 1/#)
#iowait_careful=30
#iowait_warning=40
#iowait_critical=50
# Context switch limit (core / second)
# Leave commented to just use the default config (critical is 50000*# (Logical CPU cores)
#ctx_switches_careful=10000
#ctx_switches_warning=12000
#ctx_switches_critical=14000
```

```
[percpu]
disable=False
# Define CPU thresholds in %
# Default values if not defined: 50/70/90
user_careful=50
user_warning=70
user_critical=90
iowait_careful=50
iowait_warning=70
iowait_critical=90
system_careful=50
system_warning=70
system_critical=90

[gpu]
disable=False
# Default processor values if not defined: 50/70/90
proc_careful=50
proc_warning=70
proc_critical=90
# Default memory values if not defined: 50/70/90
mem_careful=50
mem_warning=70
mem_critical=90

[mem]
disable=False
# Define RAM thresholds in %
# Default values if not defined: 50/70/90
careful=50
#careful_action_repeat=echo {{percent}} >> /tmp/memory.alert
warning=70
critical=90

[memswap]
disable=False
# Define SWAP thresholds in %
# Default values if not defined: 50/70/90
careful=50
warning=70
critical=90

[load]
disable=False
# Define LOAD thresholds
# Value * number of cores
# Default values if not defined: 0.7/1.0/5.0 per number of cores
# Source: http://blog.scoutapp.com/articles/2009/07/31/understanding-load-averages
#      http://www.linuxjournal.com/article/9001
careful=0.7
warning=1.0
critical=5.0
#log=False
```

```
[network]
disable=False
# Default bitrate thresholds in % of the network interface speed
# Default values if not defined: 70/80/90
rx_careful=70
rx_warning=80
rx_critical=90
tx_careful=70
tx_warning=80
tx_critical=90
# Define the list of hidden network interfaces (comma-separated regexp)
#hide=docker.*,lo
# WLAN 0 alias
#wlan0_alias=Wireless IF
# It is possible to overwrite the bitrate thresholds per interface
# WLAN 0 Default limits (in bits per second aka bps) for interface bitrate
#wlan0_rx_careful=4000000
#wlan0_rx_warning=5000000
#wlan0_rx_critical=6000000
#wlan0_rx_log=True
#wlan0_tx_careful=700000
#wlan0_tx_warning=900000
#wlan0_tx_critical=1000000
#wlan0_tx_log=True

[connections]
# Display additional information about TCP connections
# This plugin is disabled by default
disable=True
# nf_conntrack thresholds in %
nf_conntrack_percent_careful=70
nf_conntrack_percent_warning=80
nf_conntrack_percent_critical=90

[wifi]
disable=False
# Define the list of hidden wireless network interfaces (comma-separated regexp)
hide=lo,docker.*
# Define SIGNAL thresholds in db (lower is better...)
# Based on: http://serverfault.com/questions/501025/industry-standard-for-minimum-wifi-signal-strength
careful=-65
warning=-75
critical=-85

[diskio]
disable=False
# Define the list of hidden disks (comma-separated regexp)
#hide=sda2,sda5,loop.*
hide=loop.*,/dev/loop*
# Alias for sda1
#sda1_alias=IntDisk

[fs]
disable=False
# Define the list of hidden file system (comma-separated regexp)
hide=/boot.*./snap.*
# Define filesystem space thresholds in %
# Default values if not defined: 50/70/90
# It is also possible to define per mount point value
# Example: /_careful=40
careful=50
warning=70
critical=90
# Allow additional file system types (comma-separated FS type)
#allow=zfs
```

```
[folders]
disable=False
# Define a folder list to monitor
# The list is composed of items (list_nb <= 10)
# An item is defined by:
# * path: absolute path
# * careful: optional careful threshold (in MB)
# * warning: optional warning threshold (in MB)
# * critical: optional critical threshold (in MB)
# * refresh: interval in second between two refreshes
#folder_1_path=/tmp
#folder_1_careful=2500
#folder_1_warning=3000
#folder_1_critical=3500
#folder_1_refresh=60
#folder_2_path=/home/nicolargo/Videos
#folder_2_warning=17000
#folder_2_critical=20000
#folder_3_path=/nonexisting
#folder_4_path=/root

[irq]
# This plugin is disabled by default
# Documentation: https://glances.readthedocs.io/en/stable/aoa/irq.html
disable=False

[hddtemp]
disable=False
# Define hddtemp server IP and port (default is 127.0.0.1 and 7634 (TCP))
host=127.0.0.1
port=7634

[sensors]
# This plugin is disable by default because on some system, the PsUtil
# consume a lot of CPU to grab the stats...
disable=True
# Sensors core thresholds (in Celsius...)
# Default values if not defined: 60/70/80
temperature_core_careful=60
temperature_core_warning=70
temperature_core_critical=80
# Temperatures threshold in °C for hddtemp
# Default values if not defined: 45/52/60
temperature_hdd_careful=45
temperature_hdd_warning=52
temperature_hdd_critical=60
# Battery threshold in %
battery_careful=80
battery_warning=90
battery_critical=95
# Sensors alias
#temp1_alias=Motherboard 0
#temp2_alias=Motherboard 1
#core 0_alias=CPU Core 0
#core 1_alias=CPU Core 1

[processlist]
disable=False
# Define CPU/MEM (per process) thresholds in %
# Default values if not defined: 50/70/90
cpu_careful=50
cpu_warning=70
cpu_critical=90
mem_careful=50
mem_warning=70
mem_critical=90
#
# Nice priorities range from -20 to 19.
```

```

# Configure nice levels using a comma separated list.
#
# Nice: Example 1, non-zero is warning (default behavior)
nice_warning=-20,-19,-18,-17,-16,-15,-14,-13,-12,-11,-10,-9,-8,-7,-6,-5,-4,-3,-2,-
1,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19
#
# Nice: Example 2, low priority processes escalate from careful to critical
#nice_careful=1,2,3,4,5,6,7,8,9
#nice_warning=10,11,12,13,14
#nice_critical=15,16,17,18,19

[ports]
disable=False
# Interval in second between two scans
# Ports scanner plugin configuration
refresh=30
# Set the default timeout (in second) for a scan (can be overwritten in the scan list)
timeout=3
# If port_default_gateway is True, add the default gateway on top of the scan list
port_default_gateway=True
#
# Define the scan list (1 < x < 255)
# port_x_host (name or IP) is mandatory
# port_x_port (TCP port number) is optional (if not set, use ICMP)
# port_x_description is optional (if not set, define to host:port)
# port_x_timeout is optional and overwrite the default timeout value
# port_x_rtt_warning is optional and defines the warning threshold in ms
#
#port_1_host=192.168.0.1
#port_1_port=80
#port_1_description=Home Box
#port_1_timeout=1
#port_2_host=www.free.fr
#port_2_description=My ISP
#port_3_host=www.google.com
#port_3_description=Internet ICMP
#port_3_rtt_warning=1000
#port_4_description=Internet Web
#port_4_host=www.google.com
#port_4_port=80
#port_4_rtt_warning=1000
#
# Define Web (URL) monitoring list (1 < x < 255)
# web_x_url is the URL to monitor (example: http://my.site.com/folder)
# web_x_description is optional (if not set, define to URL)
# web_x_timeout is optional and overwrite the default timeout value
# web_x_rtt_warning is optional and defines the warning respond time in ms (approximatively)
#
#web_1_url=https://blog.nicolargo.com
#web_1_description=My Blog
#web_1_rtt_warning=3000
#web_2_url=https://github.com
#web_3_url=http://www.google.fr
#web_3_description=Google Fr
#web_4_url=https://blog.nicolargo.com/nonexist
#web_4_description=Intranet

[docker]
disable=False
#cpu_careful=50
# Thresholds for CPU and MEM (in %)
#cpu_warning=70
#cpu_critical=90
#mem_careful=20
#mem_warning=50
#mem_critical=70

```

```

#
# Per container thresholds
#containername_cpu_careful=10
#containername_cpu_warning=20
#containername_cpu_critical=30
#
# By default, Glances only display running containers
# Set the following key to True to display all containers
all=False

#####
# Client/server
#####
[serverlist]
# Define the static servers list
#server_1_name=localhost
#server_1_alias=My local PC
#server_1_port=61209
#server_2_name=localhost
#server_2_port=61235
#server_3_name=192.168.0.17
#server_3_alias=Another PC on my network
#server_3_port=61209
#server_4_name=pasbon
#server_4_port=61237

[passwords]
# Define the passwords list
# Syntax: host=password
# Where: host is the hostname
#     password is the clear password
# Additionally (and optionally) a default password could be defined
#localhost=abc
#default=defaultpassword

#####
# Exports
#####
[graph]
# Configuration for the --export graph option
# Set the path where the graph (.svg files) will be created
# Can be overwrite by the --graph-path command line option
path=/tmp
# It is possible to generate the graphs automatically by setting the
# generate_every to a non zero value corresponding to the seconds between
# two generation. Set it to 0 to disable graph auto generation.
generate_every=60
# See followings configuration keys definitions in the Pygal lib documentation
# http://pygal.org/en/stable/documentation/index.html
width=800
height=600
style=DarkStyle

[influxdb]
# Configuration for the --export influxdb option
# https://influxdb.com/
host=localhost
port=8086
protocol=http
user=root
password=root
db=glances
# Prefix will be added for all measurement name
# Ex: prefix=foo
#     => foo.cpu
#     => foo.mem
# You can also use dynamic values
#prefix=`hostname`
prefix=localhost
# Tags will be added for all measurements
#tags=foo:bar,spam:eggs
# You can also use dynamic values
#tags=system:`uname -s`
```

```
[cassandra]
# Configuration for the --export cassandra option
# Also works for the ScyllaDB
# https://influxdb.com/ or http://www.scylladb.com/
host=localhost
port=9042
protocol_version=3
keyspace=glances
replication_factor=2
# If not define, table name is set to host key
table=localhost
# If not define, username and password will not be used
#username=cassandra
#password=password

[opentsdb]
# Configuration for the --export opentsdb option
# http://opentsdb.net/
host=localhost
port=4242
#prefix=glances
#tags=foo:bar,spam,eggs

[statsd]
# Configuration for the --export statsd option
# https://github.com/etsy/statsd
host=localhost
port=8125
#prefix=glances

[elasticsearch]
# Configuration for the --export elasticsearch option
# Data are available via the ES RESTful API. ex: URL/<index>/cpu/system
# https://www.elastic.co
host=localhost
port=9200
index=glances

[riemann]
# Configuration for the --export riemann option
# http://riemann.io
host=localhost
port=5555

[rabbitmq]
# Configuration for the --export rabbitmq option
host=localhost
port=5672
user=guest
password=guest
queue=glances_queue

[mqtt]
# Configuration for the --export mqtt option
host=localhost
port=8883
user=guest
password=guest
topic=glances
tls=true

[couchdb]
# Configuration for the --export couchdb option
# https://www.couchdb.org
host=localhost
port=5984
db=glances
# user and password are optional (comment if not configured on the server side)
#user=root
#password=root
```

```
[kafka]
# Configuration for the --export kafka option
# http://kafka.apache.org/
host=localhost
port=9092
topic=glances
#compression=gzip

[zeromq]
# Configuration for the --export zeromq option
# http://www.zeromq.org
# Use * to bind on all interfaces
host=*
port=5678
# Glances envelopes the stats in a publish message with two frames:
# - First frame containing the following prefix (STRING)
# - Second frame with the Glances plugin name (STRING)
# - Third frame with the Glances plugin stats (JSON)
prefix=G

[prometheus]
# Configuration for the --export prometheus option
# https://prometheus.io
# Create a Prometheus exporter listening on localhost:9091 (default configuration)
# Metric are exporter using the following name:
# <prefix>_<plugin>_<stats>{labelkey:labelvalue}
# Note: You should add this exporter to your Prometheus server configuration:
# scrape_configs:
#   - job_name: 'glances_exporter'
#     scrape_interval: 5s
#     static_configs:
#       - targets: ['localhost:9091']
#
# Labels will be added for all measurements (default is src:glances)
# labels=foo:bar,spam:eggs
# You can also use dynamic values
# labels=system:'uname -s'
#
host=localhost
port=9091
#prefix=glances
labels=src:glances

[restful]
# Configuration for the --export RESTful option
# Example, export to http://localhost:6789/
host=localhost
port=6789
protocol=http
path=/

#####
# AMPS
# * enable: Enable (true) or disable (false) the AMP
# * regex: Regular expression to filter the process(es)
# * refresh: The AMP is executed every refresh seconds
# * one_line: (optional) Force (if true) the AMP to be displayed in one line
# * command: (optional) command to execute when the process is detected (thk to the regex)
# * countmin: (optional) minimal number of processes
#           A warning will be displayed if number of process < count
# * countmax: (optional) maximum number of processes
#           A warning will be displayed if number of process > count
# * <foo>: Others variables can be defined and used in the AMP script
#####
```

```
[amp_dropbox]
# Use the default AMP (no dedicated AMP Python script)
# Check if the Dropbox daemon is running
# Every 3 seconds, display the 'dropbox status' command line
enable=false
regex=.*dropbox.*
refresh=3
one_line=false
command=dropbox status
countmin=1

[amp_python]
# Use the default AMP (no dedicated AMP Python script)
# Monitor all the Python scripts
# Alert if more than 20 Python scripts are running
enable=false
regex=.*python.*
refresh=3
countmax=20

[amp_conntrack]
# Use comma separated for multiple commands (no space around the comma)
enable=false
regex=/sbin/init
refresh=30
one_line=false
command=sysctl net.netfilter.nf_conntrack_count;sysctl net.netfilter.nf_conntrack_max

[amp_nginx]
# Use the NGinx AMP
# Nginx status page should be enable (https://easyengine.io/tutorials/nginx/status-page/)
enable=false
regex=/usr/sbin/nginx
refresh=60
one_line=false
status_url=http://localhost/nginx_status

[amp_systemd]
# Use the Systemd AMP
enable=false
regex=/lib/systemd/systemd
refresh=30
one_line=true
systemctl_cmd=/bin/systemctl --plain

[amp_systemv]
# Use the Systemv AMP
enable=false
regex=/sbin/init
refresh=30
one_line=true
service_cmd=/usr/bin/service --status-all
```

Conclusión final

Glances nos ofrece más características y una mejor interfaz htop. Sin embargo, consume más recursos. Su punto fuerte es el montón de opciones que nos ofrece, y en excelente manual que puedes consultar [aquí](#).

Monitorizar el uso de la memoria y avisar por mail



Después de los anteriores artículos sobre el uso de la CPU y del disco, hoy toca la memoria ram; que al igual que en los anteriores, también crearemos un script que nos enviara un mail de aviso cuando nuestro sistema sobrepase el umbral definido de consumo.

Como podas comprobar es muy fácil de usar, tan solo debes definir el valor límite (umbral) que necesitas, e insertar tu email.

Monitorizar el uso de la memoria y avisar por mail

Creamos el script.

```
nano /opt/scripts/alerta-ram.sh
```

Copia y pega el siguiente script, pero atento a las opciones.

- **MIN=750** : El umbral de uso está establecido en el 750M, modifica según tus necesidades.
- **email@midominio.com** : inserta tu correo electrónico.

```
#!/bin/bash
#
#Aqui indicamos el minimo numero de MB libres para enviar un correo.
MIN=750

#Email de aviso.
MAIL='email@midominio.com'

# Fecha y hora para el log y para el email.
DATE=`date +%F`
TIME=`date +%-H:%M` 

# Ruta del registro real
LOG="/var/logs/ram/ram_$DATE.txt"

# Hostname o IP del servidor.
SERVER="127.0.0.1" #Modifica segun tus necesidades

echo "">>$LOG
echo "Valores del dia $DATE a las $TIME">>$LOG
echo "">>$LOG

FREE=`free -m | grep 'Mem:' | awk {'print $4'}`

echo "RAM disponible: `free -m | grep 'Mem:' | awk {'print $2'}` MB">>$LOG
echo "RAM libre: $FREE MB">>$LOG

if [ $FREE -lt $MIN ]; then
# Enviamos el mail si se sobrepasa el umbral.
mail $MAIL -s "ATENCION!! La memoria en $SERVER esta al limite." >>
$LOG
fi
```

Guarda el archivo y cierra el editor.

Ahora creamos una tarea cron que ejecute el script cada 10 minutos.

```
crontab -e
```

Añade la siguiente linea.

```
*/10 * * * * /bin/bash
/opt/scripts/alerta-ram.sh
```

Guarda el archivo y cierra el editor, solo nos falta reiniciar cron.

```
# Debian, Ubuntu, y derivados
sudo service cron restart

# CentOS, RHEL, y derivados
systemctl restart crond.service
```

Síguenos en las Redes:



Generar contraseñas en linux

Hoy en día es necesario tener **contraseñas seguras**, para ello tenemos a nuestra disposición herramientas generadoras, sitios web online, scripts, y un sinfín de formulas varias.

Pues a partir de ya, todas las opciones que conocías las debes desechar. Para que vas a perder el tiempo en esas aplicaciones, si con simples comandos en nuestra **terminal linux** puedes generar contraseñas seguras.

En el articulo de hoy vemos varios ejemplos, usa el que más se adapte a tus necesidades. En todos los ejemplos propuestos se generan passwords de 32 caracteres, puedes modificar ese valor si es necesario.

Generar contraseñas en linux

- **Ejemplo 1**

```
tr -dc [:graph:] < /dev/urandom | head -c 32| xargs echo;
```

Salida...

`YW2O!+RT={rFwlWk:AE9~R^mbY6v=`

- **Ejemplo 2**

```
tr -dc A-Za-z0-9_!@#$%^&*()+= < /dev/urandom | head -c 32 | xargs
```

Salida...

`Ry%$5ahalgtvM&WMq9w0@nOSSWg@XQI1`

- **Ejemplo 3**

```
LC_CTYPE=C tr -dc A-Za-z0-9_!@#$%^&*()+= < /dev/urandom | head -c 32 | xargs
```

Salida...

`$GVs(%24IPc$L6#(7ka2W4Y3R=58@SRW`

- **Ejemplo 4**

```
LC_ALL=C; cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 32 | head -n 1
```

Salida...

`5bRIMZrUHquyxRYr9PyJlqSP6yMMmneO`

- **Ejemplo 5**

```
tr -dc '[[:alnum:]]' < /dev/urandom | dd bs=4 count=8 2>/dev/null | xargs echo;
```

Salida...

`OybRMuktde4lWDztmjXrKSQ5Tt6Ga0j`

- **Ejemplo 6**

```
cat /dev/urandom | tr -dc 'a-zA-Z0-9' | head -c 32| xargs echo;
```

Salida...

`xvdo7i7gqwz2k7v91sd4ch4qai7futyo`



- **Ejemplo 7**

```
cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 32 | head -n 1
```

Salida...

gWJw6rSl01ypmnSO3kVtxluiSrwu7wH7

- **Ejemplo 8**

```
head -c 16 /dev/urandom | md5sum | head -c 32| xargs echo;
```

Salida...

a52278c2c4bff3d1444548711c1547d3

- **Ejemplo 9**

```
base64 /dev/urandom | tr -d '/+' | head -c 32 | tr '[:upper:]' '[:lower:]' | xargs echo;
```

Salida...

qpulavbzcnqlrg9pldqat3yjee0ywuh0

- **Ejemplo 10**

```
head -c24 < <(tr -dc '\041-\176' < /dev/urandom) | xargs echo;
```

Salida...

q8g_Y^&\$6NK:(snjo@l[al

- **Ejemplo 11**

```
head -c24 < /dev/random | base64
```

Salida...

IYqAv4WeoyyLseCGVMz6dR8hSU/Ba52W

Ejemplo 12

```
echo $(cat /dev/urandom | tr -dc [:print:] | tr -d [:space:]'\042\047\134' | fold -w 32 | head -n 1)
```

Salida...

KC5Z*3QwJls62d#*0h&^/GwmR%u \$ha.

Ejemplo 13

```
openssl rand -base64 32
```

Salida...

LLVGxBujrl2eO6likY94vAvwaPMxxOnEVXe2/PBF0vM=

Ejemplo 14

```
openssl rand -hex 32
```

Salida...

64b24882cb4e4d4833d0e97384666b02a39ad6296011f6701aa0fec9d61251ba

Ejemplo 15

```
openssl rand -base64 500 | tr -dc 'a-zA-Z0-9' | fold -w 32 | head -n 1
```

Salida...

EO3ZVawwc85lluzzDInh0ZcUvZse3I9j

- **Ejemplo 16**

```
xxd -l16 -ps /dev/urandom
```

Salida...

fce5d20dc09a028f9b1b4be63c65de0c

- **Ejemplo 17**

```
cat /proc/sys/kernel/random/uuid
```

Salida...

4f757af5-41f7-49a3-98a7-1afd9e12ed62

Como puedes comprobar linux nos ofrece muchas opciones para generar contraseñas, utiliza la que más se adapte a tus necesidades.

Síguenos en las Redes:



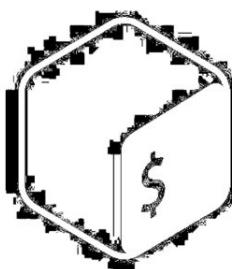
Canales de Telegram: [Canal SoloLinux](#) – [Canal SoloWordpress](#)

Esta revista es de **distribución gratuita**, si lo consideras oportuno puedes ponerle precio.
Tu también puedes ayudar, contamos con la posibilidad de hacer donaciones para la REVISTA, de manera muy simple a través de **PAYPAL**

AYUDANOS A SEGUIR CRECIENDO



Como escribir scripts bash correctamente



Escribir scripts bash

SoloLinux correctamente

Los **scripts bash** son la forma más sencilla de ejecutar y automatizar tareas en nuestro linux. Son rápidos, seguros y muy fiables.

En este articulo vemos las leyes de buenas prácticas a la hora de crear un script, o mejor dicho... Como escribir **scripts bash** correctamente para que sea compresible, y lo más importante, reutilizable.

Como escribir scripts bash correctamente

Aportamos los diez consejos que cualquier **desarrollador bash** debería seguir.

Siempre agregar comentarios

Los comentarios ayudan a comprender el desarrollo del script, y facilitan la tarea de otro desarrollador que quiera continuar o mejorar tu script. Recuerda que son a modo indicativo, para que no se intenten ejecutar deben estar comentados (carácter #).

```
# El script de sololinux.es, .....
```

Salir del script al encontrar un error

Bash es muy potente, tanto, que a veces al producirse un error es capaz de seguir ejecutándose. No debes permitirlo, para subsanar este problema usamos la opción abreviada de «errexit».

```
# El script de sololinux.es
```

```
set -e
```

Salir del script cuando bash use una variable no declarada

Algunas veces bash puede intentar ejecutar variables no declaradas, esto podría causar un desarrollo anormal del resultado final del script. Para salir del script utilizamos la opción abreviada de «nunset».

```
# El script de sololinux.es
```

```
set -u
```

Las referencias a variables deben llevar comillas dobles

Cuando usamos las comillas dobles en las referencias a variables, nos evitamos la división de palabras y el tener que utilizar más espacios de los debidos.

```
#!/bin/bash
# El script de sololinux.es
```

```
set -e
```

```
set -u
```

```
echo "Salida sin comillas dobles"
```

```
echo
```

```
names="YO USO LINUX"
```

```
for name in $names; do
    echo "$name"
```

```
done
```

```
echo
```

```
echo "Salida con comillas dobles"
```

```
echo
```

```
for name in "$names"; do
    echo "$name"
```

```
done
```

```
exit 0
```

Imagen de ejemplo...

```
sololinux sergio # bash ejemplo-comillas.sh
Salida sin comillas dobles
```

```
YO
USO
LINUX
```

```
Salida con comillas dobles
```

```
YO USO LINUX
```

```
sololinux sergio #
```

www.sololinux.es

Utiliza las funciones

Este es un gran defecto de los programadores noveles. Acostúmbrate a usar las funciones, se modulariza el código y es más legible.

```
function check_root(){
    comando-1;
    comando-2;
}
```

OR

```
check_root(){
    comando-1;
    comando-2;
}
```

Si generas el código en una linea, no te olvides de terminar.

```
check_root(){ comando-1; comando-2; }
```

No compares cadenas con ==

No es necesario introducir los dos caracteres de igual, con uno es suficiente. Ahorraras espacio y trabajo.

```
value1="megustalinux"
value2="megustasololinux"
if [ "$value1" = "$value2" ]
```

Usa \$ en vez de las comillas inversas

Para sustituir comandos usa el símbolo del dolar en vez de las comillas inversas.

```
user=`echo "$UID"`
user=$(echo "$UID")
```

Utiliza mayúsculas para las variables de entorno

Para las variables de entorno debes usar letras mayúsculas, en cambio si son variables personalizadas deberías utilizar letras minúsculas. Con esta practica evitaras muchos conflictos entre variables.

```
solo_linux="$HOME/Trabajos/scripts/demo.sh"
perl "$solo_linux" -h "$1"
```

Usa readonly en las variables estáticas

Una vez definidas en un script las variables estáticas no se deben modificar, así que las ponemos en modo solo-lectura.

```
readonly hostname_file="/etc/hostname"
readonly localtime_file="/etc/localtime"
readonly network_file="/etc/network"
```

Depurar el código

Si son scripts con pocas lineas no es necesario, pero si escribes scripts bash con cientos de lineas debes depurarlo antes de ponerlo en producción. En un próximo artículo veremos como depurar el código de bash.

Cómo desfragmentar en linux



Me sorprendo con las afirmaciones que veo en otros sitios, en ellas se asegura que en linux no es necesario desfragmentar.

Estas equivocado amigo, en linux tambien puede ser necesario desfragmentar un disco. Lo que pasa es que los **sistemas de archivos linux** periodizan mucho mejor los archivos que otros sistemas, por ejemplo que **windows**.

En sistemas Fat y Fat32 los archivos se van acoplando correlativamente (uno detrás de otro), el efecto negativo es, que cuando (ejemplo) subes un archivo a una carpeta ya creada, tambien lo inserta correlativamente sea cual sea el anterior archivo.

En los sistemas de archivos **NTFS** este efecto se intento corregir, logrando una pequena mejora aun insuficiente. Los **sistemas de archivos linux** como EXT2,3 y 4, JFS, ZFS, XFS, ReiserFS y BTRFS, tratan los archivos de otra forma, digamos que dejan mas espacio en blanco para que puedas subir archivos y no tengas problemas de fragmentacion.

El porqué y el cómo, es algo más complejo y largo de explicar pero con la siguiente imagen te puedes hacer una idea de como tratan los archivos los diferentes sistemas.

En discos hdd no muy grandes o en maquinas un poco más antiguas es altamente recomendable desfragmentar el disco, no con tanta frecuencia como en Windows pero si cuando notes que la carga de archivos demora demasiado. En este articulo vemos como **desfragmentar un disco en linux** de forma segura.

Cómo desfragmentar en linux

Existen herramientas como **e2fsprogs** o **fsck**, pero no las recomiendo (tienes que desmontar el disco). Yo te propongo un **script perl** que funciona realmente bien, ofreciendo unos resultados espectaculares.

Creamos el script.

```
sudo nano desfrag.pl
```

Copia y pega el script. ([Pagina siguiente..](#))



```

#!/bin/perl
# defrag hdd

trap 'abort' 1 2 15
renice 19 $$ > /dev/null

abort()
{
echo -e "\nAborting"
rm -f tmpfile dirlist
exit 1
}

fail()
{
echo -e "\nFailed"
abort
}

declare -i filesize=0
declare -i numfiles=0

#The maximum size of a file we can easily cache in ram
declare -i maxsize=$((`awk '/MemTotal/ {print $2}' \
/proc/meminfo`*1024))
(( maxsize-= `awk '/Mapped/ {print $2}' \
/proc/meminfo` ))
(( maxsize/= 2))

if [[ -a tmpfile || -a dirlist ]] ; then
echo dirlist or tmpfile exists
exit 1
fi

# Sort in the following order:
# 1) Depth of directory
# 2) Size of directory descending
# 3) Filesize descending
# I made this crap up. It's completely unvalidated.

echo "Generando la lista de archivos..."

#stupid script to find max directory depth
find -xdev -type d -printf "%d\n" | sort -n | uniq >
dirlist

#sort directories in descending size order
cat dirlist | while read d;
do
find -xdev -type d -mindepth $d -maxdepth $d -printf
"\%p\n" |
xargs du -bS --max-depth=0 | \
sort -k 1,1nr -k 2 | \
cut -f2 >> tmpfile
if (( $? )) ; then
fail
fi

done

rm -f dirlist

#sort files in descending size order
cat tmpfile | while read d;
do
find "$d" -xdev -type f -maxdepth 1 -printf "%s\t%p\n" |
sort -k 1,1nr | \
cut -f2 >> dirlist
if (( $? )) ; then
fail
fi
done

rm -f tmpfile

numfiles=`wc -l dirlist | awk '{print $1}'`
echo -e "$numfiles files will be reordered\n"

#copy to temp file, check the file hasn't changed and
then overwrite original
cat dirlist | while read i;
do
(( --numfiles ))
if [[ ! -f $i ]]; then
continue
fi

```

```

#We could be this paranoid but it would slow it down 1000
times
#if [[ `lsof -f -- "$i"` ]]; then
#echo -e "\n File $i open! Skipping"
#continue
#fi

filesize=`find "$i" -printf "%s"`
# read the file first to cache it in ram if possible
if (( filesize < maxsize ))
then
echo -e "\r $numfiles files left \c"
cat "$i" > /dev/null
else
echo -e "\r $numfiles files left - Reordenando archivos
con tamaño de $filesize ... \c"
fi

datestamp=`find "$i" -printf "%s"`
cp -a -f "$i" tmpfile
if (( $? )) ; then
fail
fi
# check the file hasn't been altered since we copied it
if [[ `find "$i" -printf "%s"` != $datestamp ]] ; then
continue
fi

mv -f tmpfile "$i"
if (( $? )) ; then
fail
fi
done

echo -e "\nDesfragmentacion concluida"
rm -f dirlist

```

Guarda el archivo y cierra el **editor nano**.

El script necesita permisos para ejecutarse, se los concedemos.

```
chmod u+x desfrag.pl
```

Llego el momento de ejecutarlo. El **script desfragmentador** permite desfragmentar todo el disco sin tener que desmontarlo, pero no es necesario. Cuando trabajamos en linux lo hacemos sobre /HOME, y eso es lo que debes desfragmentar.

Ejecuta el siguiente comando.

```
sudo ./desfrag.pl /home
```

El proceso comenzara a listar los archivos a reordenar, se paciente.

```
Creating list of files...
41178 files will be reordered                               www.sololinux.es
| 40718 files left █ Reordering large file sized 2027891965 ...
```

Dependiendo del tamaño del disco y de tu sistema, el tiempo del proceso puede variar.

```
Creating list of files...
41178 files will be reordered                               www.sololinux.es
0 files left      - Reordering large file sized 2027891965 ...
Succeeded
```

Nota final:

Es importante que una vez comience la desfragmentación cierres todas las ventanas y aplicaciones del sistema, no lo toques hasta que termine el proceso, déjalo trabajar.

Este script es super recomendado para los que hacen un uso intensivo de las descargas (por ejemplo torrent), bajan de la red una gran cantidad de datos y también los borran.

Una vez termine el proceso de desfragmentación debes reiniciar el sistema, no te preocupes el script no a borrado nada, solo lo cambio de sitio (aun así, debes tener una copia de seguridad). Al reiniciar el sistema notaras enseguida que tu linux corre más ligero.

En algunos sistemas puede aparecer un aviso de error en la linea 26 y 27, no te preocupes no pasa nada, es porque no hemos definido el tamaño de los archivos. El script continuara su proceso sin la intervención del usuario.



Esta revista es de **distribución gratuita**, si lo consideras oportuno puedes ponerle precio.
Tu también puedes ayudar, contamos con la posibilidad de hacer donaciones para la REVISTA, de manera muy simple a través de **PAYPAL**

AYUDANOS A SEGUIR CRECIENDO



Síguenos en las Redes:



Conocer el estado de fragmentación de un disco

Después del artículo anterior he llegado a la conclusión de que existe una gran controversia sobre como **conocer el estado de fragmentación de un disco**.

Al igual que en el artículo anterior, vamos a crear un **script perl** que al ejecutarlo nos imprimirá en pantalla la fragmentación de nuestro disco.

Conocer el estado de fragmentación de un disco

Creamos el script perl.

```
sudo nano frag.pl
```

Copia y pega el script.

```
#!/usr/bin/perl -w

#this script search for frag on a fs
use strict;

#number of files
my $files = 0;
#number of fragment
my $fragments = 0;
#number of fragmented files
my $fragfiles = 0;

#search fs for all file
open (FILES, "find " . $ARGV[0] . " -xdev -type f -print0 |");

$/ = "\0";

while (defined (my $file = <FILES>)) {
    open (FRAG, "-|", "filefrag", $file);
    my $res = <FRAG>;
    if ($res =~ m/.*: \s+ (\d+) extents? found/) {
        my $fragment = $1;
        $fragments += $fragment;
        if ($fragment > 1) {
            $fragfiles++;
        }
        $files++;
    } else {
        print ("$res : not understand for
$file.\n");
    }
    close (FRAG);
}
close (FILES);

print ( $fragfiles / $files * 100 . "% non contiguous files, " . $fragments /
$files . " average fragments.\n");
```



www.sololinux.es

Guarda el archivo y cierra el editor nano.

El script necesita que le concedamos permisos.

```
chmod u+x frag.pl
```

Como por defecto siempre trabajamos sobre **/HOME**, solo necesitamos conocer ese resultado.

```
sudo ./frag.pl /home
```

El resultado puede demorar un poco, todo depende del tamaño del disco y del sistema. Al concluir, verás algo similar a...

```
soloLinux sergio # chmod u+x frag.pl
soloLinux sergio # sudo ./frag.pl /home
0.79456587126529% non contiguous files, 0.993282534589934 average fragments.
soloLinux sergio #
```

Nota final: Si descargas y borras archivos de gran tamaño muy a menudo, es interesante que ejecutes tanto este script, como el que **desfragmenta tu hdd**.

Tip: La forma natural de desfragmentar un disco en linux es hacer una copia de tu **/home** en un dispositivo externo, ahora borras todo el contenido de **/home** y restauras la copia de seguridad. Linux colocará cada archivo en su sitio, pero el problema de este método es el tiempo (tarda mucho), por eso recomiendo el uso de los scripts perl.

Ubicación geográfica de una ip en terminal

En este mini tuto, vamos a ver como identificar la ubicación geográfica de cualquier ip desde nuestra **terminal linux**.

Su uso está bastante extendido sobre todo en formularios, y sitios web que te dicen tu ip y ubicación. Ya veras que simple.

Ubicación geográfica de una ip en terminal

El único requisito es tener instalado curl y jq (necesario para procesar los datos json).

En Debian, Ubuntu, Linux Mint y derivados:

```
sudo apt install curl jq
```

En CentOS, RHEL y derivados:

```
sudo yum install curl jq
```

Es evidente que para poder ubicar un ip, primero debemos capturarla. Podemos lanzar una solicitud a la API a ipinfo.io con el siguiente comando.

```
curl https://ipinfo.io/ip
```

Una vez tenemos la ip, realizamos otra solicitud a la API de [ipvigilante](https://ipvigilante.com) (incluyendo la ip capturada en el paso anterior).

```
curl https://ipvigilante.com/<ip-capturada>
```

Ejemplo...

```
curl https://ipvigilante.com/90.94.251.85
```

Bueno... ahora ya sabemos como opera la ubicación geográfica, así que si la quieres integrar en alguna aplicación lo mejor es ejecutar las dos solicitudes en una sola linea.

```
curl -s https://ipvigilante.com/$(curl -s https://ipinfo.io/ip) | jq  
.data.latitude, .data.longitude, .data.city_name, .data.country_name'
```

Nota final: ipvigilante.com es un proyecto en fase beta, por tanto es posible que alguna vez se produzca un error.



Las mejores distribuciones linux del 2020



Como todos sabemos el **soporte** de **Windows 7** se ha terminado, llego el momento del cambio. Tienes dos opciones, o actualizas a otro sistema peor que el que tenías (a **windows 10**), o bien optas por la opción más inteligente y te pasas a linux.

En este articulo, veremos las distribuciones que más se adaptan a los usuarios que se quieren pasar de **Windows a Linux** (según mi punto de vista), independientemente de sus conocimientos.

Los entornos de escritorio de linux son excelentes, la experiencia de usuario también. Hoy vamos a ver las **distribuciones linux del 2020** con las que un usuario que viene de Windows se pueda sentir más familiarizado. Una buena interfaz de usuario, un gestor de aplicaciones comprensible, ademas de ayuda en la migración es lo que busca el nuevo usuario de linux.

Indice de contenido

- [Linux Mint](#)
- [Feren OS](#)
- [Zorin OS](#)
- [Deepin](#)
- [Linux Lite](#)
- [Otras distribuciones](#)

Las mejores distribuciones linux del 2020

Muchos os vais a sorprender al no a incluir Ubuntu en este articulo, pero creo que al no incluir software privativo de manera predeterminada puede causar algún que otro problema en algunos usuarios. Si están varios de sus derivados, alguno sorprendente.

Las mejores distribuciones linux del 2020

Muchos os vais a sorprender al no incluir Ubuntu en este artículo, pero creo que al no incluir software privativo de manera predeterminada puede causar algún otro problema en algunos usuarios. Si están varios de sus derivados, alguno sorprendente.

Linux Mint



Linux Mint se basa en **Ubuntu**, por tanto se beneficia de una de las distribuciones que revolucionó el mundo linux, y de su tremenda gama de aplicaciones y herramientas.

Con su administrador de software podrás instalar todo tipo de utilidades, si le sumamos una interfaz con cierto parecido a Windows7 y una excelente compatibilidad con hardware nuevo, o que ya tenga unos años (mejor que Ubuntu), Linux Mint debe estar en la lista de sustitutos de Windows. Sus requisitos mínimos son bastante simples.

- 1 GB de RAM (se recomiendan 2 GB).
- 15 GB de espacio en disco (se recomiendan 20 GB).
- 800 x 600 de resolución (para una optima experiencia de usuario, 1024 x 768 o más).

[Pagina oficial de descargas.](#)

Zorin OS

Zorin OS también se basa en **Ubuntu**, pero tiene como valor añadido una interfaz muy parecida a **Windows 7**.

Los que la conocen saben lo bien que puedes trabajar con ella, ahora... en la ultima versión aun se ha optimizado más su facilidad de uso. Cambios como la instalación de aplicaciones o la interacción con dispositivos móviles, mejoran la adaptación de los nuevos usuarios.

Requisitos mínimos de Zorin OS:

- **Zorin OS Ultimate, Educación y Core**
CPU de 1 GHz de doble núcleo – 64 bits
RAM 2 GB
Espacio en disco de 10 GB (Core y educación) o 20 GB (Ultimate)
Resolución 800 x 600

Feren OS



Vaya sorpresa me he llevado con esta **distribución Linux**, si fuera por la solidez y veteranía de **Linux Mint** tal vez ocuparía el primer puesto.

Con solo tres años entre nosotros, **Feren OS** coge lo mejor de **Ubuntu** y de **Linux Mint**, el resultado es un remix espectacular que sorprenderá a propios y extraños. Tiene soporte nativo para aplicaciones de windows (no tienes que preocuparte de instalar Wine), incluso un importador de datos y algunas configuraciones (desde windows).

En breve haremos un artículo especial sobre ella, se lo merece; seguro que en breve será de las más instaladas. Sus requisitos mínimos son similares a los de Linux Mint.

- 1,5 GB de RAM (se recomiendan 2 GB).
- 15 GB de espacio en disco (se recomiendan 20 GB).
- 800 x 600 de resolución (para una optima experiencia de usuario, 1024 x 768 o más).

[Pagina oficial de descargas de Feren OS](#)



• **Zorin OS Lite:**

CPU de un solo núcleo a 700 MHz: 64 bits o 32 bits
RAM 512 MB
Espacio en disco de 8 GB
Resolución 640 x 480

[Pagina oficial de descargas de Zorin OS](#)

Deepin



Linux Deepin es la alternativa perfecta a **Windows 10**. Basada en Debian y Ubuntu, esta distribución viene con una gran cantidad de software preinstalado.

Fue desarrollada por una empresa de China con el fin de no depender del monopolio Windows. Su interfaz y diseño están muy cuidados, a la vez que intuitivos. Deepin nos ofrece su propia tienda de aplicaciones en la que seguro encontrarás lo que necesitas.

Sus requisitos mínimos son los siguientes:

- CPU Pentium IV 2GHz o superior
- Memoria de 2 GB (se recomiendan 4 GB)
- Como mínimo 25 GB libres en el disco

[Pagina oficial para descargar Deepin](#)

Otras distribuciones

Soy consciente que nos dejamos grandes distribuciones en el tintero, pero estas son las que a mi parecer son las más indicadas para los que vienen de Windows 7, y nunca manejaron linux.

Aun así, también podemos recomendar...

- MX Linux
- Manjaro Linux
- Ubuntu Desktop
- elementary OS
- Solus
- Open Suse
- Fedora
- y alguna más

Linux Lite



Linux Lite es de esas distribuciones linux que todos hemos probado alguna vez. Es ligera, bonita, rápida, y con un buen paquete de software instalado por defecto.

Siendo conscientes que se terminaba el soporte de Windows 7, lanzaron su ultima versión junto a un llamamiento (un tanto peculiar), a los usuarios de Windows 7. Prefiero que lo veas tu mismo...

Nos gustaría aprovechar esta oportunidad para dar la bienvenida a todas las personas con Windows 7 que han venido aquí para encontrar una alternativa simple, rápida y gratuita a Windows 7 que ha llegado al final de su vida útil y ya no proporciona actualizaciones de seguridad. Linux Lite hace la transición a un sistema operativo basado en Linux al ofrecer un paquete completo de Office compatible con Microsoft, un software familiar como Firefox, Chrome, Teamviewer, VLC, así como herramientas completas de copia de seguridad del sistema, un manual de ayuda completo y fácil de seguir para guiar usted en su viaje, Steam para que pueda seguir jugando sus juegos de Windows y un software mucho más familiar. Nuestro escritorio se presenta igual que en Windows con un menú de inicio a la izquierda y una bandeja a la derecha con opciones de volumen, red y calendario, con iconos familiares de escritorio que lo llevan exactamente a donde desea ir en su sistema.

A pesar que es un distribución que me gusta, tiene un problema con nuestro idioma; nunca llega a estar al 100% en español (castellano). Sus requerimientos a nivel de hardware son muy básicos.

- Cpu de 1Ghz (recomendado 1,5Ghz)
- Memoria de 768Mb (recomendado 1GB)
- Espacio en disco 8 Gb (recomendado 20Gb)
- Una resolucion de 1024 x 768 (recomendado 1366 x 768 o superior)

[Descargar Linux Lite desde su pagina oficial](#)

Colorear la salida de comandos con el arcoíris

Navegando por la red encontré una aplicación realmente curiosa, se llama **lolcat**. Esta pequeña herramienta concatena una amplia gama de colores sobre la salida de **comandos**.

No aporta ninguna función especial ni mejora el rendimiento de la terminal, simplemente nos ofrece unos alegres tonos de colores en la salida de los comandos o scripts que ejecutemos en la terminal. En este artículo vemos como **instalar lolcat** y sus diversos modos de uso.

Colorear la salida de comandos con el arcoíris

Instalar lolcat

El único requisito de **lolcat** es que tengas instalado el lenguaje de programación **Ruby**, normalmente ya viene con la mayoría de **distribuciones linux**, por si acaso...

Debian , Ubuntu, Linux Mint, y derivados:

```
sudo apt install -y ruby
```

CentOS, RHEL, y derivados:

```
sudo yum install -y ruby
```

Fedora, y otras distros con paquetes dnf:

```
sudo dnf install -y ruby
```

Ahora vamos a **instalar lolcat**.

En Ubuntu, Linux Mint y derivados, **lolcat** viene en los repositorios oficiales. Ejecuta el siguiente comando:

```
sudo apt install lolcat
```

En otras distribuciones (o si quieres tener la ultima versión), lo puedes descargar y descomprimir desde su fuente original.

```
wget https://github.com/busyloop/lolcat/archive/master.zip && unzip master.zip
```

Instalamos **lolcat** desde la carpeta que hemos descomprimido.

```
cd lolcat-master/bin && gem install lolcat
```

Verificamos la versión instalada de **lolcat** (si instalaste desde los repositorios oficiales de Ubuntu será la versión 42).

```
lolcat --version
```



Ejemplo de salida...

```
sololinux ~ # lolcat --version
lolcat 100.0.0 (c)2011 moe@busyloop.net
```

Cómo usar lolcat

La ayuda de la herramienta es muy simple pero concreta.

```
lolcat -h
```

Ejemplo de salida...

```
Usage: lolcat [OPTION]... [FILE]...
Concatenate FILE(s), or standard input, to standard output.
With no FILE, or when FILE is -, read standard input.

--spread, -p <f>: Rainbow spread (default: 3.0)
--freq, -F <f>: Rainbow frequency (default: 0.1)
--seed, -S <i>: Rainbow seed, 0 = random (default: 0)
--animate, -a: Enable psychedelics
--duration, -d <i>: Animation duration (default: 12)
--speed, -s <f>: Animation speed (default: 20.0)
--force, -f: Force color even when stdout is not a tty
--version, -v: Print version and exit
--help, -h: Show this message
www.sololinux.es

Examples:
lolcat f - g      Output f's contents, then stdin, then g's contents.
lolcat             Copy standard input to standard output.
fortune | lolcat Display a rainbow cookie.

Report lolcat bugs to <http://www.github.org/busyloop/lolcat/issues>
lolcat home page: <http://www.github.org/busyloop/lolcat/>
Report lolcat translation bugs to <http://speaklolcat.com/>
```

La sintaxis es simple.

```
sudo [comando/script] | lolcat
```

Como ejemplo listamos los dispositivos de almacenamiento de nuestro sistema.

```
sudo fdisk -l | lolcat
```

Un ejemplo con el comando ping...

```
sudo ping google.com -c10 | lolcat
```

Con un script bash.

```
sudo bash libera.sh| lolcat
```

Lolcat también nos permite definir el tiempo de concatenación de cada linea, este efecto es muy vistoso dado que mientras espera a la siguiente linea, los colores van moviéndose de lado a lado.

En este caso agregamos unos valores a la sintaxis.

```
sudo [comando/script] | lolcat -a -d [Valor]
```

El valor debe ser numero (a más valor, más lento), y lo podemos aplicar a cualquiera de los ejemplo anteriores. Nosotros usamos fdisk.

```
sudo fdisk -l | lolcat -a -d 20
```

Si necesitas más información puedes visitar el [Github oficial del proyecto](#).



Esta revista es de **distribución gratuita**, si lo consideras oportuno puedes ponerle precio.
Tu también puedes ayudar, contamos con la posibilidad de hacer donaciones para la REVISTA, de manera muy simple a través de **PAYPAL**

AYUDANOS A SEGUIR CRECIENDO



Síguenos en las Redes:



Instalar Ganglia Monitoring Server en Ubuntu



Ganglia, es un sistema de monitoreo para sistemas y redes de alto rendimiento. Creado en 2002 bajo **licencia BSD** por la **Universidad de Berkeley**, es la base de muchos de los sistemas de monitorización que todos conocemos.

Ganglia utiliza estructuras de datos y algoritmos especialmente diseñados para lograr un consumo mínimo del sistema, independientemente de los datos masivos a recolectar. Dado su carácter poderoso y robusto, actualmente se usa en miles de clusters de todo el mundo, también se admite en la mayoría de sistemas basados en **Unix**, y varias arquitecturas.

En este artículo veremos como **instalar y configurar un servidor Ganglia** (específico), para monitorizar varios servidores, vps o clusters.

Instalar Ganglia Monitoring Server en Ubuntu

Nosotros instalamos **Ganglia Monitoring en Ubuntu 18.04 LTS**, pero también es válido para otros sistemas similares. Comenzamos.

Instalar Apache

Instalamos el servidor **Apache**.

```
sudo apt update
```

```
sudo apt install apache2
```

Iniciamos y habilitamos Apache con el inicio del sistema.

```
sudo systemctl start apache2.service
```

```
sudo systemctl enable apache2.service
```

Ya tenemos Apache instalado y corriendo en nuestro servidor.

Instalar PHP 7.2

La versión 7.2 de php viene de manera predeterminada en los repositorios de **Ubuntu 18.04 LTS**, si instalas **Ganglia** en otras versiones como por ejemplo **Ubuntu 16.04 LTS** debes agregar el siguiente repositorio (si no es tu caso, omite).

```
sudo apt-get install software-properties-common  
sudo add-apt-repository ppa:ondrej/php
```

Actualizamos e instalamos PHP 7.2 con los módulos necesarios para el correcto funcionamiento de Ganglia.

```
sudo apt update
```

```
sudo apt install php7.2 libapache2-mod-php7.2 php7.2-common php7.2-gmp php7.2-curl php7.2-intl php7.2-mbstring php7.2-xmlrpc php7.2-mysql php7.2-gd php7.2-xml php7.2-cli php7.2-zip
```

Ahora vamos a **configurar php.ini**.

```
sudo nano /etc/php/7.2/apache2/php.ini
```

La configuración que te propongo es correcta para Ganglia, aun así, puedes modificarla dependiendo de tus necesidades.

```
file_uploads = On  
allow_url_fopen = On  
short_open_tag = On  
memory_limit = 256M  
upload_max_filesize = 100M  
max_execution_time = 360  
max_input_vars = 1500  
date.timezone = Europe/Madrid
```

Reiniciamos Apache.

```
sudo systemctl restart apache2.service
```

La instalación de php a terminado, vamos a instalar el **monitoring server**.

Instalamos la herramienta Ganglia

Nuestro servidor web ya está en marcha, continuamos con la instalación de Ganglia desde los repositorios oficiales predeterminados.

```
sudo apt update
```

```
sudo apt install ganglia-monitor rrdtool  
gmetad ganglia-webfrontend
```

Una vez termine la instalación, iniciamos y habilitamos la herramienta para que arranque con el sistema.

```
sudo systemctl start ganglia-
monitor.service
```

```
sudo systemctl enable ganglia-
monitor.service
```

Puedes verificar que se instaló correctamente y que está corriendo, con el comando...

```
systemctl status ganglia-monitor.service
```

Ejemplo de salida válida...

```
● ganglia-monitor.service
    Loaded: loaded (/etc/init.d/ganglia-monitor; generated)
      Active: active (running) since Wed 2020-01-23
                 07:12:46 CST; 38s ago
        Docs: man:systemd-sysv-generator(8)
       Tasks: 2 (limit: 3927)
      CGroup: /system.slice/ganglia-monitor.service
              └─20853 /usr/sbin/gmond --pid-file
/var/run/gmond.pid
```

```
Jan 23 07:12:46 ubuntu1804 systemd[1]: Starting
ganglia-monitor.service...
```

```
Jan 23 07:12:46 ubuntu1804 ganglia-monitor[20853]:
Starting Ganglia Monitor Daemon: gmond.
```

```
Jan 23 07:12:46 ubuntu1804 systemd[1]: Started
ganglia-monitor.service.
```

Perfecto... Ganglia ya funciona.

Configurar Ganglia

El archivo de configuración predeterminado es «gmetad.conf», para que Ganglia nos funcione bien debemos hacer algunas modificaciones.

```
sudo nano /etc/ganglia/gmetad.conf
```

Edita la ip de «my cluster». Esta IP debe ser del servidor que trabajara como nodo principal.

```
# A list of machines which service the
# data source follows, in the
# format ip:port, or name:port. If a
# port is not specified then 8649
# (the default gmond port) is assumed.
# default: There is no default value
#
# data_source "my cluster" 10 localhost
my.machine.edu:8649 1.2.3.5:8655
# data_source "my grid" 50 1.3.4.7:8655
grid.org:8651 grid-backup.org:8651
# data_source "another source"
1.3.4.7:8655 1.3.4.8

data_source "my cluster" 50
192.168.0.100:8655 <<<-----
#
```

Guarda el archivo y cierra el editor.

También debemos insertar la ip principal en el archivo «gmond.conf».

```
sudo nano /etc/ganglia/gmond.conf
```

Cambia las IP.

```
/* Feel free to specify as many
udp_send_channels as you like. Gmond
used to only support having a single
channel */
udp_send_channel {
  mcast_join = 192.168.101.101
<<<-----
  port = 8649
  ttl = 1
}

/* You can specify as many
udp_recv_channels as you like as well.
*/
udp_recv_channel {
  mcast_join = 192.168.101.101
<<<-----
  port = 8649
  bind = 192.168.101.101 <<<-----
}

/* You can specify as many
tcp_accept_channels as you like to share
an xml description of the state of
the cluster */
tcp_accept_channel {
  port = 8649
}
```

Guarda el archivo y cierra el editor.

Copiamos el archivo de configuración que hemos editado, al virtual host de Apache.

```
sudo cp
/etc/ganglia-webfrontend/apache.conf
/etc/apache2/sites-enabled/ganglia.conf
```

La instalación de Ganglia a concluido, solo nos falta reiniciar los servicios requeridos.

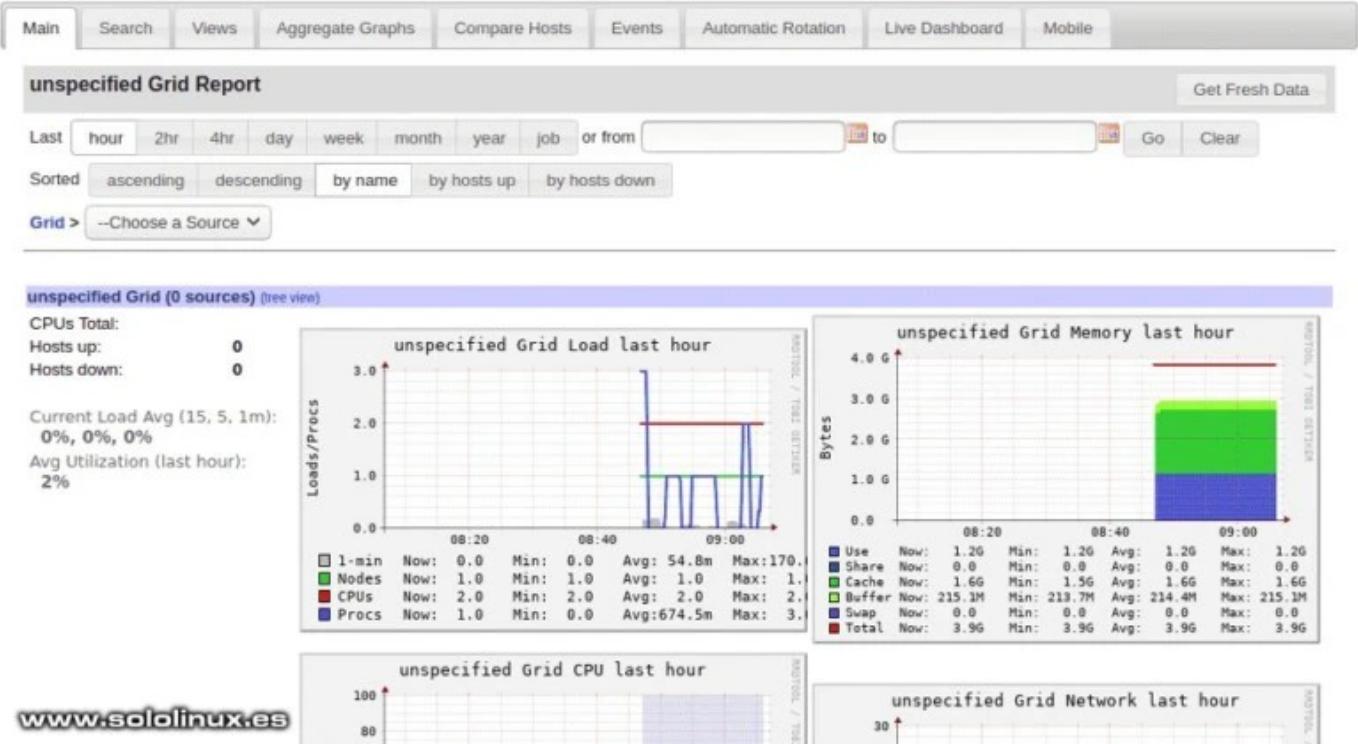
```
sudo systemctl restart ganglia-monitor
sudo systemctl restart gmetad
sudo systemctl restart apache2
```

Acceder a Ganglia Monitoring

Para acceder a Ganglia inserta la siguiente url (con tu ip), en tu navegador web favorito.

<http://ip-servidor-primario/ganglia>

Aparece una pantalla similar a la de la siguiente imagen:



Revisa todas las opciones disponibles y configura lo que quieras.

Instalar clientes Ganglia

La instalación de los **clientes de Ganglia** es muy sencilla, tan solo debemos instalar la aplicación **ganglia-monitor**, e insertar la ip del servidor primario en su archivo de configuración.

```
sudo apt install ganglia-monitor
```

Abrimos el archivo de configuración.

```
sudo nano /etc/ganglia/gmond.conf
```

Inserta la ip del servidor primario tal como te indico a continuación, recuerda que estamos en el cliente.

```
udp_send_channel {
    host = 192.168.0.100 <<<-----
    port = 8649
    ttl = 1
}
```

Guarda el archivo y cierra el editor.

Iniciamos y habilitamos la herramienta cliente para que arranque con el sistema.

```
systemctl start ganglia-monitor
systemctl enable ganglia-monitor
```

En pocos minutos los sistemas cliente comenzaran a enviar datos al servidor primario.

Ha concluido la instalación y configuración de Ganglia Monitoring Server.

Cómo hacer una prueba de carga en un servidor con Siege



Prueba de carga de un servidor

Quieres saber la carga que soporta tu servidor?, es la pregunta que todos nos hacemos alguna vez. Hoy vemos como instalar y utilizar la veterana herramienta **Siege HTTP Load Testing**.

Siege es una utilidad capaz de lanzar pruebas de carga http y benchmarking, a cualquier dominio que tu le indiques. Con Siege podrás verificar como reacciona tu sitio web o servidor, bajo una presión constante de visitas.

Esta fabulosa aplicación admite la autenticación básica, cookies, y los protocolos HTTP, HTTPS y FTP. Nos permite configurar el numero de clientes simulados, estos clientes pondrán al servidor en una situación límite, más conocida como «servidor bajo asedio».

Cómo hacer una prueba de carga en un servidor con Siege

Siege viene en la mayoría de repositorios oficiales de las diversas **distribuciones linux**, pero omitimos este tipo de instalación ya que no siempre traen la ultima versión. Mejor desde su código fuente.

Sigue los pasos que te indicamos.

```
wget
http://download.joedog.org/siege/siege-
latest.tar.gz

tar -zvxf siege-latest.tar.gz

cd asedio - * /

sudo ./configure --prefix = / usr /
local --with-ssl = / usr / bin / openssl

sudo make && make install
```

Ahora generamos el archivo de configuración.

```
sudo siege.config
```

Ejemplo de salida...

```
sololinux siege-4.0.4 # sudo siege.config
/usr/local/bin/siege.config: 32:
/usr/local/bin/siege.config: cannot create
/root/.siege/siege.conf: Directory nonexistent
New configuration template added to /root/.siege
Run siege -C to view the current settings in that file
```

El archivo de configuración se generó en `~/.siege/siege.conf`.

Ejemplo...

```
verbose = false
color = on
quiet = false
show-logfile = true
logging = false
gmethod = HEAD
parser = true
nofollow = ad.doubleclick.net
nofollow = pagead2.googlesyndication.com
nofollow = ads.pubsqrd.com
nofollow = ib.adnxs.com
limit = 255
protocol = HTTP/1.1
chunked = true
cache = false
connection = close
concurrent = 25
time = 1M
delay = 0.0
internet = false
benchmark = false
accept-encoding = gzip, deflate
url-escaping = true
unique = true
```

Puedes modificar el archivo según tus necesidades.

Para ejecutar la prueba de carga tan solo tienes que ejecutar el siguiente comando.

```
siege misitioweb.com
```

Puedes visitar su [Github oficial](#) si necesitas más información.

Minetest: la mejor alternativa a Minecraft

Según los jugones, **Minetest** es la **mejor alternativa** a **Minecraft**. Hablamos de un juego gratuito y de código abierto compatible con **Linux**, **Windows**, **macOS**, **BSD** y **Android**.

Este juego está inspirado en el popular **Minecraft**. Como detalle importante destacamos que viene con el motor del juego al completo, esto quiere decir que puedes crear tus propios juegos y mods.

Soporte multijugador, paquetes de texturas, generadores de mapas, modificaciones y bloques de $62,000 \times 62,000 \times 62,000$. En fin, tiene de todo y por eso se publica bajo la licencia **GNU Lesser General Public License versión 2.1+**.

Minetest: la mejor alternativa a Minecraft

Vemos como instalar Minetest en varias distribuciones linux.

Ubuntu, Linux Mint, y derivados:

```
sudo add-apt-repository  
ppa:minetestdevs/stable  
  
sudo apt-get update  
  
sudo apt install minetest
```



Debian:

```
sudo apt-get update  
  
sudo apt-get install minetest
```

OpenSuse:

```
zypper refresh  
  
zypper install minetest
```

Fedora y derivados:

```
sudo dnf install minetest
```

Arch Linux, Manjaro, y derivados:

Descomenta el repositorio «Community».

Guarda y cierra el editor. Instalamos Minetest.

```
sudo pacman -Syy minetest
```

FreeBSD:

```
install minetest minetest_game
```

Instalar Minetest con Flatpak:

```
flatpak install flathub net.minetest.Minetest
```

Instalar el paquete Snap:

```
sudo snap install minetest
```

Ejecutas el juego con...

```
minetest
```

AppArmor vs SELinux

www.sololinux.es



Para aumentar los mecanismos de seguridad que ofrecen los permisos y las listas de control de acceso «ugo/rwx», la **Agencia de Seguridad Nacional de los Estados Unidos (NSA)** desarrollo un control de acceso obligatorio que todos conocemos como **SELinux (Security Enhanced Linux)**.

Por otro lado, y a modo privativo la empresa **Immunix** creo **AppArmor**. **Novell** adquirió AppArmor, y puso la herramienta en manos de la comunidad en formato open source. Actualmente es **Canonical** quien maneja su desarrollo y mantenimiento.

RHEL, CentOS y Fedora, son las distribuciones linux más conocidas que usan SELinux por defecto. Por otra parte Ubuntu, Linux Mint y Open Suse, están entre las destacadas que utilizan AppArmor (Open Suse desarrollo su propia GUI para manejar fácilmente las reglas de la herramienta).

Sería muy difícil hacer una comparativa realista de las dos aplicaciones. El fin de las dos es el mismo, pero su forma de operar difiere en gran medida, por eso vamos a explicar un poco cada herramienta, y tu decides cual te conviene dado que ambas son compatibles con cualquier **distribución linux**.

AppArmor vs SELinux

AppArmor

Si hay algo que me gusta de AppArmor es su modo de autoaprendizaje, es capaz de detectar cómo debe funcionar nuestro sistema de forma automática.

En vez de políticas administradas por comandos, AppArmor utiliza perfiles que se definen en archivos de texto que pueden ser modificados de forma muy sencilla. Puedes encontrar los perfiles predefinidos (se permite agregar más), con el siguiente comando:

```
cd /etc/apparmor.d
```

```
dir
```

Ejemplo de edición...

```
nano usr.bin.firefox
```

Profundizar en la configuración de SELinux es complejo, se requieren unos conocimientos nivel medio/alto. Por el motivo mencionado se creo AppArmor, digamos que puede hacer lo mismo pero de una forma mucho mas sencilla y con menos peligro (si te equivocas editas de nuevo el archivo o desactivas el perfil).

Para desactivar un perfil tan solo tienes que ejecutar lo siguiente (en el ejemplo el de firefox):

```
sudo ln -s /etc/apparmor.d/usr.sbin.firefox /etc/apparmor.d/disable/
```

AppArmor ofrece dos modos predefinidos de seguridad, complain y enforce. Modificar un perfil con un modo de seguridad predeterminado también es una operación simple. En el ejemplo modificamos el perfil «usr.sbin.ntpd».

```
# complain
aa-complain /etc/apparmor.d/usr.sbin.ntpd

# enforce
aa-enforce /etc/apparmor.d/usr.sbin.ntpd
```

También podemos verificar el estado actual de AppArmor y visualizar los perfiles habilitados.

```
apparmor_status
```

```
sololinux ~ # apparmor_status
apparmor module is loaded.
13 profiles are loaded.
13 profiles are in enforce mode.
/sbin/dhclient
/snap/snapd/5754/usr/lib/snapd/snap-confine
/snap/snapd/5754/usr/lib/snapd/snap-confine//mount-namespace-capture-helper
/usr/lib/NetworkManager/nm-dhcp-client.action
/usr/lib/NetworkManager/nm-dhcp-helper
/usr/lib/connman/scripts/dhclient-script
/usr/lib/lightdm/lightdm-guest-session
/usr/lib/lightdm/lightdm-guest-session//chromium
/usr/lib/snapd/snap-confine
/usr/lib/snapd/snap-confine//mount-namespace-capture-helper
/usr/sbin/ippusbxd
/usr/sbin/ntpd
/usr/sbin/tcpdump
0 profiles are in complain mode.
1 processes have profiles defined.
1 processes are in enforce mode.
  /usr/sbin/ntpd (1283)
0 processes are in complain mode.
0 processes are unconfined but have a profile defined.
sololinux ~ #
```

www.sololinux.es

Para más información visita la [Ubuntu wiki](#).

SELinux

Por su forma de operar **SELinux** puede ser mucho más estricto que **AppArmor**, incluso muchas herramientas y **paneles de control web** recomiendan desactivarlo para no tener problemas.

Al igual que **AppArmor**, **SELinux** (Security Enhanced Linux) también tiene dos modos de protección:

- enforcing: SELinux niega accesos en función de la política de las reglas establecidas.
- permissive: SELinux no bloquea el acceso, pero se registrarán para un posterior análisis.

Puedes verificar el modo que estas utilizando con el comando...

```
getenforce
```

Podemos modificar el modo (incluso deshabilitar SELinux) en su archivo de configuración.

```
nano /etc/selinux/config
```

Cambias la linea:

```
SELINUX= xxx
```

Por alguna de las siguientes opciones:

- enforcing
- permissive
- Disabled

Guarda el archivo, cierra el editor, y reinicia el sistema.

```
reboot
```

Verificamos el estado de SELinux.

```
[root@...]# cat /etc/selinux/config
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of three two values:
#       targeted - Targeted processes are protected.
#       minimum - Modification of targeted policy. Only selected processes are protected.
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted

[root@...]# sestatus
```



www.sololinux.es

Ahora vemos unos ejemplos de uso. Una de las operaciones más comunes es al **modificar el puerto ssh** (por defecto 22), es evidente que debemos decirle a SELinux que ssh cambia de puerto, por ejemplo el 123.

```
semanage port -a -t ssh_port_t -p tcp
123
```

Otro ejemplo es cambiar la carpeta permitida como servidor web.

```
semanage fcontext -a -t
httpd_sys_content_t "/srv/www(/.*)?"
```

```
restorecon -R -v /srv/www
```

Puedes obtener más información en su [wiki oficial](#).

Conclusión final

Los dos sistemas de seguridad tratados en este artículo nos ofrecen herramientas para aislar aplicaciones, y a un posible atacante del resto del sistema (cuando una aplicación es comprometida).

Los conjuntos de reglas de **SELinux** son excesivamente complejos, por otro lado nos permite tener más control sobre cómo se aíslan los procesos. La generación de las políticas puede ser automática, aun así su manejo es complicado si no eres experto.

AppArmor es más sencillo de usar. Los perfiles se pueden escribir a mano con tu editor preferido, o generarlos con «**aa-logprof**». AppArmor utiliza un control basado en rutas, por tanto el sistema es más transparente y se puede verificar de forma independiente.

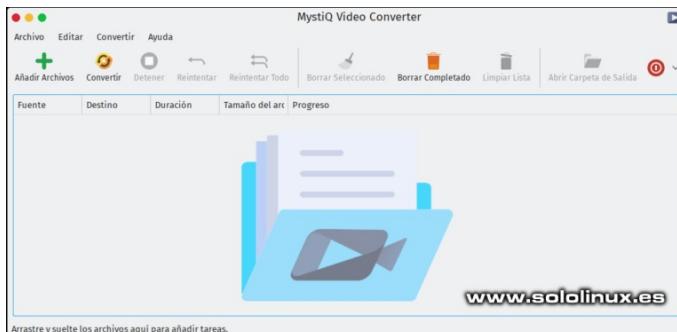
La elección es tuya.

Instalar MystiQ Video Converter en Ubuntu y derivados



MystiQ Video Converter es una interfaz gráfica de **FFmpeg**, gracias a su facilidad de uso nos permite convertir vídeos de un formato a otro con unos pocos clicks.

Basado en **Qt5** y **SoX** es de **código abierto**. Destacamos que viene con un conjunto avanzado de ajustes predeterminados, que nos ayudan a convertir archivos multimedia de manera muy simple. También nos permite ajustar los parámetros de conversión según nuestras necesidades.



Instalar MystiQ Video Converter

MystiQ necesita **Qt 5.10** o superior, por tanto la instalación puede variar dependiendo de tu **versión de Ubuntu** o derivados equivalentes. En este artículo tratamos la instalación en **Ubuntu 18.04 LTS Bionic Beaver**, y en **Ubuntu 19.10 Eoan Ermine**.

Instalar MystiQ en Ubuntu 18.04 LTS

Ubuntu 18.04 viene por defecto con **Qt 5.9**, así que ademas de instalar las herramientas necesarias para el correcto funcionamiento de **MystiQ Video Converter**, también debemos actualizar Qt.

```
sudo add-apt-repository ppa:beineri/opt-qt-5.12.6-bionic
sudo apt update
sudo apt install build-essential g++
ffmpeg sox mplayer libnotify-dev mesa-common-dev libglvnd-dev
sudo apt install qt-latest tree
```

```
echo "/opt/qt512/bin" | sudo tee /etc/xdg/qtchooser/default.conf
```

```
echo "/opt/qt512/lib" | sudo tee -a /etc/xdg/qtchooser/default.conf
```

```
qtchooser -print-env
```

Nos falta descargar e instalar **MystiQ Video Converter**.

```
wget https://github.com/llamaret/MystiQ/releases/download/v20.01.22/mystiq_20.02.15-1_amd64.deb
```

```
sudo dpkg -i mystiq_20.02.15-1_amd64.deb
```

```
sudo apt-get install -f
```

Instalar MystiQ en Ubuntu 19.10

Instalamos las herramientas necesarias.

```
sudo apt install ffmpeg sox mplayer libnotify-dev qt5-default qt5-qmake g++
```

Descargamos la ultima versión de **MystiQ Video Converter** (en este caso «**mystiq_20.02.15-1_amd64.deb**»).

```
wget https://github.com/llamaret/MystiQ/releases/download/v20.01.22/mystiq_20.02.15-1_amd64.deb
```

Instalamos...

```
sudo dpkg -i mystiq_20.02.15-1_amd64.deb
```

```
sudo apt-get install -f
```

Nota final: Antes de instalar la versión propuesta puedes verificar «[aquí](#)» si existe alguna más actualizada.

Uso del comando seq con ejemplos

El **comando seq** es una herramienta capaz de imprimir una secuencia de números naturales, que pueden ser transmitidos a otros **scripts** o aplicaciones.

Al combinar el **comando seq** con otros **comandos**, podemos generar secuencias numéricas, bucles, y varias opciones más. En este artículo conocemos este comando, muy fácil de usar y de entender.

Uso del comando seq con ejemplos

Seq viene de forma predeterminada en todas las **distribuciones linux**, puedes verificar la versión que tienes instalada ejecutando lo siguiente.

```
seq --version
```

Ejemplo de salida...

```
sololinux ~ # seq --version
seq (GNU coreutils) 8.25
Copyright © 2016 Free Software Foundation, Inc.
Licencia GPLv3+: GPL de GNU versión 3 o posterior
<http://gnu.org/licenses/gpl.html>.
Esto es software libre: usted es libre de cambiarlo y
redistribuirlo.
No hay NINGUNA GARANTÍA, hasta donde permite la
ley.
```

Escrito por Ulrich Drepper.

```
sololinux ~ #
```

Como puedes comprobar en la siguiente línea la sintaxis de **seq** es muy simple.

```
seq [options] specification
```

Ejemplos básicos de seq

1- Imprimir una secuencia que empiece en 1 hasta un numero definido, por ejemplo el 7.

```
seq 7
```

Ejemplo de salida...

```
sololinux ~ # seq 7
1
2
3
4
5
6
7
sololinux ~ #
```

2- Definir el primer y ultimo numero de la secuencia. En el ejemplo comenzamos por 3, y concluimos con 8.



```
seq 3 8
```

Ejemplo de salida...

```
sololinux ~ # seq 3 8
3
4
5
6
7
8
Sololinux ~ #
```

3- Para generar una secuencia aritmética definida, usamos la siguiente sintaxis.

```
seq [start] [incr] [stop]
```

En el siguiente ejemplo comenzamos la secuencia en 5, e incrementamos de tres en tres hasta lograr el numero 15. Como el numero 15 no hace la suma, se imprimirá el más cercano (en este caso el 14).

```
seq 5 3 15
```

Ejemplo de salida...

```
sololinux ~ # seq 5 3 15
5
8
11
14
Sololinux ~ #
```

4- Los valores también pueden ser negativos, incluso decimales con coma flotante.

```
seq 14 -3 5
```

Ejemplo de salida...

```
sololinux ~ # seq 14 -3 5
14
11
8
5
sololinux ~ #
```

Formatear la salida del comando seq

Como has podido comprobar, en los ejemplos anteriores el resultado se imprime con saltos de linea. Ahora vemos como formatear la salida para que nos ofrezca otras opciones de impresión.

1- Impresión de secuencia numérica separada por barra vertical. Utilizamos la opción **-s**.

```
seq -s "|" 12
```

Ejemplo de salida...

```
sololinux ~ # seq -s "|" 12
1|2|3|4|5|6|7|8|9|10|11|12
sololinux ~ #
```

2- La opción **-f** nos permite utilizar el estilo **printf** con los caracteres **E, e, f, G, g, %**.

```
seq -f '##%g##' 1 7
```

Ejemplo de salida...

```
sololinux ~ # seq -f '##%g##' 1 7
##1##
##2##
##3##
##4##
##5##
##6##
##7##
sololinux ~ #
```

Otro ejemplo de salida...

```
sololinux ~ # seq -f %f 1 7
1,000000
2,000000
3,000000
4,000000
5,000000
6,000000
7,000000
sololinux ~ #
```

3- Con **-w** podemos alinear la secuencia numérica agregando ceros.

```
seq -w 5 12
```

Ejemplo de salida...

```
sololinux ~ # seq -w 5 12
05
06
07
08
09
10
11
12
sololinux ~ #
```

Integrar seq en un bash loop

El comando **seq** también se puede utilizar para hacer bucles en scripts bash. Creamos un sencillo script a modo de ejemplo.

```
nano demo-seq.sh
```

Insertamos lo siguiente...

```
#!/bin/bash
for i in $(seq 4 0.5 8)
do
echo "El numero es $i"
done
```

Guarda el archivo, cierra el editor, y lo ejecutas.

Ejemplo de salida...

```
sololinux # bash demo-seq.sh
El numero es 4,0
El numero es 4,5
El numero es 5,0
El numero es 5,5
El numero es 6,0
El numero es 6,5
El numero es 7,0
El numero es 7,5
El numero es 8,0
sololinux #
```

Ayuda del comando seq

Para imprimir la ayuda en el idioma predefinido de tu sistema, ejecutamos **-help**.

```
seq --help
```

```
sololinux ~ # seq --help
Modo de empleo: seq [OPCIÓN]... ÚLTIMO
o bien: seq [OPCIÓN]... PRIMERO ÚLTIMO
o bien: seq [OPCIÓN]... PRIMERO INCREMENTO ÚLTIMO
Muestra los números desde PRIMERO hasta ÚLTIMO, en incrementos de INCREMENTO.

Los argumentos obligatorios para las opciones largas son también obligatorios
para las opciones cortas.
-f, --format=FORMATO    utiliza un FORMATO de coma flotante en estilo printf
-s, --separador=CADENA  utiliza CADENA para separar los números
                        (por omisión: '\n')
-w, --equal-width        iguala el ancho rellenando con ceros
--help                  muestra esta ayuda y finaliza
--version               informa de la versión y finaliza
```

Si se omiten PRIMERO o INCREMENTO, el valor predeterminado es 1. Es decir, un INCREMENTO omitido tiene un valor predeterminado de 1 incluso cuando ÚLTIMO es más pequeño que PRIMERO. La sucesión de números termina cuando la suma del número actual y el incremento sería mayor que ÚLTIMO.

PRIMERO, INCREMENTO y ÚLTIMO se interpretan como valores de coma flotante.

INCREMENTO es normalmente positivo si PRIMERO es menor que ÚLTIMO, e

INCREMENTO es normalmente negativo si PRIMERO es mayor que ÚLTIMO.

FORMATO debe ser apropiado para imprimir un argumento de tipo 'double'; de forma predeterminada es %.PRECf si PRIMERO, INCREMENTO, y Último

son todos números decimales de coma fija con una precisión máxima de PREC,

y %g en caso contrario.

ayuda en línea sobre GNU coreutils: <<http://www.gnu.org/software/coreutils/>>

Informe de errores de traducción en seq a <<http://translationproject.org/team/es.html>>

Full documentation at: <<http://www.gnu.org/software/coreutils/seq>>

or available locally via: info '(coreutils) seq invocation'

Navegar de forma anónima con Tor y ProxyChains



Hoy en día la privacidad es fundamental, y en muchas ocasiones nos vemos obligados a usar la **red Tor**. Es evidente que nada es infalible, así que nosotros vamos a navegar por Internet agregando a Tor, la herramienta **ProxyChains**.

¿Qué es **ProxyChains**?

ProxyChains es una herramienta con la capacidad de conectarse a varios proxy a través de los protocolos HTTP(S), SOCKS4 y SOCKS5. Al usar la aplicación con Tor se dificulta enormemente la detección de nuestra ip real. Sus características principales, son:

- Permite encadenar diferentes proxies a la vez HTTP/S- SOCKS4 – SOCKS5.
- Puedes elegir la forma de encadenar los proxy: aleatorio (random), estricto (strict), dinámico (dynamic), round Robind.
- Tiene la capacidad de resolver las DNS (UDP) a través del proxy.
- Posibilidad de integrar el número de proxies encadenados que necesites (los que estén caídos se los salta).
- Proxychains trabaja con cualquier aplicación de cliente TCP, por ejemplo, nmap, telnet, etc.
- Admite casi todos los tipos de servidores, como: like squid, sendmail, etc.

Las funciones de **ProxyChain** son variadas, puedes escanear de red, conectar por telnet o ssh para tareas oscuras, o simplemente puedes **navegar por internet** con tu navegador favorito. Nosotros no vamos a fomentar el **black hack**, así que en este artículo te voy a enseñar como navegar por la web con tu privacidad asegurada (o casi...).

Navegar de forma anónima con Tor y ProxyChains

Lo primero que debemos hacer es **instalar Tor y ProxyChains**. Las dos aplicaciones vienen de forma predeterminada en los repositorios casi todas las distribuciones linux.

Instalar Tor y ProxyChains en Debian, Ubuntu, Linux Mint, y derivados

```
sudo apt install tor proxychains
```

Instalar Tor y ProxyChains en CentOS, RHEL, Fedora, y derivados

```
yum install epel-release  
yum install tor proxychains
```

```
-----  
dnf install epel-release  
dnf install tor proxychains
```

Instalar Tor y ProxyChains en Arch Linux, Manjaro, y derivados

```
pacman -S tor proxychains
```

Instalar Tor y ProxyChains en OpenSuse

```
zypper install tor proxychains
```

Instalar Tor y ProxyChains en Gentoo

```
emerge tor proxychains
```

Instalar Tor y ProxyChains en FreeBSD

```
pkg install tor proxychains
```

No te olvides de **iniciar Tor**.

```
service tor start
```

Configurar ProxyChains

Una vez instaladas las herramientas necesitamos configurar ProxyChains, y para ello vamos a editar el archivo «proxychains.conf».

```
sudo nano /etc/proxychains.conf
```

Modo de encadenar los proxy

Ejecuta la acción que te indico en las flechas del archivo de configuración.

```
# proxychains.conf VER 3.1
#
#      HTTP, SOCKS4, SOCKS5 tunneling proxifier with DNS.
#
# The option below identifies how the ProxyList is treated.
# only one option should be uncommented at time,
# otherwise the last appearing option will be accepted
#
#dynamic_chain <<<---descomentar
#
# Dynamic - Each connection will be done via chained proxies
# all proxies chained in the order as they appear in the list
# at least one proxy must be online to play in chain
# (dead proxies are skipped)
# otherwise EINTR is returned to the app
#
#strict_chain <<<---comentar
#
# Strict - Each connection will be done via chained proxies
# all proxies chained in the order as they appear in the list
# all proxies must be online to play in chain
# otherwise EINTR is returned to the app
```

Debe quedar así...

```
# proxychains.conf VER 3.1
#
#      HTTP, SOCKS4, SOCKS5 tunneling proxifier with
DNS.
#
#
# The option below identifies how the ProxyList is treated.
# only one option should be uncommented at time,
# otherwise the last appearing option will be accepted
#
dynamic_chain
#
# Dynamic - Each connection will be done via chained
proxies
# all proxies chained in the order as they appear in the list
# at least one proxy must be online to play in chain
# (dead proxies are skipped)
# otherwise EINTR is returned to the app
#
#strict_chain
#
# Strict - Each connection will be done via chained proxies
# all proxies chained in the order as they appear in the list
# all proxies must be online to play in chain
# otherwise EINTR is returned to the app
```

Guarda los cambios pero **NO CIERRES EL ARCHIVO**, aun no hemos terminado.

Buscar ip de proxy gratuitos

Existen muchos sitios donde podemos encontrar las ip de proxy (gratuitas), pero sin dudarlo uno de los mejores es:

<https://hidemy.name/es/proxy-list/>

Puedes definir las opciones del proxy, pero si es solo para navegar por la web puedes seguir mis recomendaciones.

- 1) Selecciona tu país preferido.
- 2) Tipos de Proxy.
- 3) Apunta las cinco primeras ip que aparecen (como mínimo), suelen ser las más rápidas (puede variar dependiendo de la hora o el día).

The screenshot shows a search interface for proxy servers. Step 1 highlights the 'Country' dropdown set to 'All countries (99)'. Step 2 highlights the 'Tipo de Proxy' checkboxes for 'HTTPS', 'Socks 4', and 'Socks 5'. Step 3 highlights the table of results, which lists five proxy entries from Spain, each with its IP address, port, location, speed, type, anonymity level (Alto), and last check time.

| Dirección IP | Puerto | País, Ciudad | Velocidad | Tipo | Anonimato | Cuando |
|-----------------|--------|-------------------|-----------|--------|-----------|--------------|
| 109.167.38.1 | 4145 | Spain Madrid | 460 ms | SOCKS4 | Alto | 6 minutes |
| 2.137.25.193 | 4145 | Spain Getafe | 460 ms | SOCKS4 | Alto | 6 minutes |
| 137.101.138.173 | 4145 | Spain Cordoba | 460 ms | SOCKS4 | Alto | 6 minutes |
| 79.156.102.136 | 4145 | Spain "el Morche" | 460 ms | SOCKS4 | Alto | 28 minutes |
| 88.2.24.50 | 40246 | Spain Barcelona | 1480 ms | SOCKS4 | Alto | 34 minutes |
| 45.65.65.18 | 4145 | Spain | 460 ms | SOCKS4 | Alto | 34 minutes |
| 80.24.119.213 | 4145 | Spain Algete | 1420 ms | SOCKS4 | Alto | 1 h. 19 min. |
| 45.129.166.40 | 4145 | Spain | 4040 ms | SOCKS4 | Alto | 2 h. 16 min. |
| 5.152.174.89 | 4145 | Spain Madrid | 2800 ms | SOCKS4 | Alto | 2 h. 51 min. |

Agregar las ip proxy a ProxyChains

Bien... ahora continuamos con el archivo «proxychains.conf» que teníamos abierto. Al final del archivo vemos algo parecido a...

```
# proxy types: http, socks4, socks5
# ( auth types supported: "basic"-http "user/pass"-socks )
#
[ProxyList]
# add proxy here ...
# meanwhile
# defaults set to "tor"
Socks4 127.0.0.1 9050
```

NO borres ni comentes el socks4 que tenemos habilitado, simplemente agrega debajo (linea por linea) las ip de los proxy elegidos.

Por ejemplo:

```
# proxy types: http, socks4, socks5
# ( auth types supported: "basic"-http "user/pass"-socks )
#
[ProxyList]
# add proxy here ...
# meanwhile
# defaults set to "tor"
socks4 127.0.0.1 9050
https 109.167.38.1
https 2.137.25.193
https 137.101.138.173
https 79.156.102.136
Https 88.2.24.50
```

Guardamos el archivo y cerramos el editor.

Necesitamos reiniciar Tor.

```
service tor restart
```

Navegar por internet con ProxyChains

La conexión a ProxyChains la establecemos desde la consola / terminal, pero ojo... sin ser root, como tu usuario normal.

La sintaxis es la siguiente:

```
proxychains <aplicacion> <argumento-o- url>
```

En nuestro ejemplo nos conectamos a «Google.com» con el navegador Firefox. Mira que fácil...

```
proxychains firefox www.google.com
```

No cierras la terminal, si lo haces se cortara la conexión.

Veras algo similar a...

```
sergio@sololinux ~ $ proxychains firefox www.google.com
ProxyChains-3.1 (http://proxychains.sf.net)

(firefox:21320): Gtk-WARNING **: Theme directory  of theme oxygen has no size field

|DNS-request| detectportal.firefox.com
|DNS-request| www.google.com
|S-chain|->-127.0.0.1:9050-<>-4.2.2.2:53-[S-chain]->-127.0.0.1:9050-<>-4.2.2.2:53-<>-OK
-><>-OK
|DNS-response| www.google.com is 216.58.212.4
|S-chain|->-127.0.0.1:9050-<>-216.58.212.4:443-[DNS-response| detectportal.firefox.com is 2.21.77.145
-><>-OK
|S-chain|->-127.0.0.1:9050-<>-2.21.77.145:80-[DNS-request| content-signature-2.cdn.mozilla.net
-><>-OK
|S-chain|->-127.0.0.1:9050-<>-4.2.2.2:53-[DNS-request| ocsp.pki.goog
-><>-OK
|S-chain|->-127.0.0.1:9050-<>-4.2.2.2:53-<>-OK
|DNS-response| content-signature-2.cdn.mozilla.net is 13.35.253.70
|S-chain|->-127.0.0.1:9050-<>-13.35.253.70:443-[DNS-response| ocsp.pki.goog is 172.217.17.195
-><>-OK
|S-chain|->-127.0.0.1:9050-<>-172.217.17.195:80-[DNS-request| mozilla.org
|DNS-request| mozilla.org
-><>-OK
|DNS-request| detectportal.firefox.com
|S-chain|->-127.0.0.1:9050-<>-2.21.77.145:80-[S-chain]->-127.0.0.1:9050-<>-4.2.2.2:53-[S-chain]->-127.0.0.1:9050-<>-4.2.2.2:53-[S-chain]->-127.0.0.1:9050-<>-4.2.2.2:53-[S-chain]->-127.0.0.1:9050-<>-4.2.2.2:53-[S-chain]->-127.0.0.1:9050-<>-4.2.2.2:53-[S-chain]->-127.0.0.1:9050-<>-4.2.2.2:53-<>-OK
-><>-OK
-><>-OK
-><>-OK
-><>-OK
|DNS-response| detectportal.firefox.com is 2.21.77.210
|DNS-response| mozilla.org is 63.245.208.195
|DNS-response| mozilla.org is 63.245.208.195
|DNS-request| mozilla.org
|S-chain|->-127.0.0.1:9050-<>-4.2.2.2:53-[DNS-response| search.services.mozilla.com is 34.212.103.110
|S-chain|->-127.0.0.1:9050-<>-34.212.103.110:443-<>-OK
|DNS-response| mozilla.org is 63.245.208.195
-><>-OK
|DNS-request| ocsp.digicert.com
|S-chain|->-127.0.0.1:9050-<>-4.2.2.2:53-[DNS-request| firefox.settings.services.mozilla.com
|S-chain|->-127.0.0.1:9050-<>-4.2.2.2:53-<>-OK
|DNS-request| ssl.gstatic.com
-><>-OK
|DNS-request| lh3.googleusercontent.com
|DNS-request| id.google.com
|DNS-response| ocsp.digicert.com is 93.184.220.29
|S-chain|->-127.0.0.1:9050-<>-93.184.220.29:80-[S-chain]->-127.0.0.1:9050-<>-4.2.2.2:53-[DNS-response| firefox.settings.
```

www.sololinux.es

Automáticamente se abrirá el **navegador firefox** a través de Tor y de ProxyChains.

Nota final: Si quieres **desinstalar Tor y ProxyChains** debes hacerlo de la forma habitual, por ejemplo...

```
sudo apt remove --purge tor proxychains
```

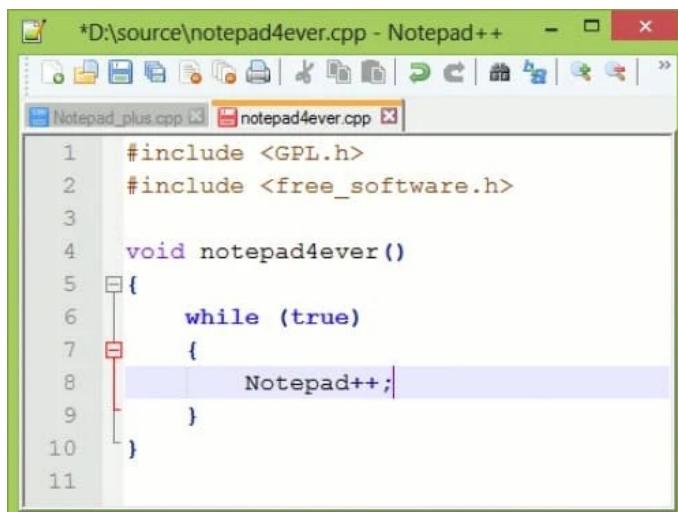
Alternativas a Notepad++ en Linux



Me comentan los que vienen de **Windows** que echan de menos a **Notepad++**, hablan de un editor muy rápido, funcional, y ademas **open source**. Hablamos del editor **Notepad++**.

Notepad++ destaca por su amplia gama de lenguajes, como PHP, HTML, CSS, Pascal, Perl, C++, Assembly, y muchos más. También permite abrir y editar varios archivos a la vez, ademas de la opción de buscar y reemplazar.

De forma predeterminada viene con complementos básicos, pero muy útiles; convertir, exportar, copiar, pegar, país, etc..., en fin, podemos decir que **Notepad++** es un editor decente. Aun así considero que los editores de linux no tienen mucho que envidiar a Notepad++, y eso que en este articulo no vamos a tratar editores de categoría PRO, tan solo vamos a ver los más similares tanto a nivel visual como productivo.



Alternativas a Notepad++ en Linux Notepadqq

```
libera.sh
35 sleep 3s; echo "Habilitar HTTP y reiniciar MariaDB"
36 systemctl restart mariadb
37 sleep 1s; service nginx restart
38 service memcached start
39 echo "OK - Http habilitado"
40 echo "<----->" 
41
42 sleep 3s; echo "Habilitando Memcached"
43 chkconfig memcached on
44 service memcached start
45 echo "OK - Memcached habilitado"
46 echo "<----->" 
47 service dosd restart
48 sleep 2s; echo "Comprobacion rutinaria"
49 echo "Script liberador sh"
50 echo "Created by SergioG.B."
51 echo "https://www.sololinux.es"
52 echo "<----->" 
53
54 sleep 2s; free
55 echo "...TODO CORRECTO...."
56 echo "<----->" 
57 echo "<----->"
```

Notepadqq es visualmente igual que Notepad++, podríamos decir que es un clon del original. También es de código abierto y nos ofrece todo lo que esperamos de un editor de estas características. Esquemas de color, código, soporte para complementos, y el resaltado de sintaxis para más de 100 lenguajes de programación.

Instalar Notepadqq es tarea sencilla.

Debian, Ubuntu, Linux Mint, y derivados:
`sudo add-apt-repository ppa:notepadqq-team/notepadqq`
`sudo apt-get update`
`sudo apt-get install notepadqq`

Fedora y derivados:
`sudo dnf install notepadqq`

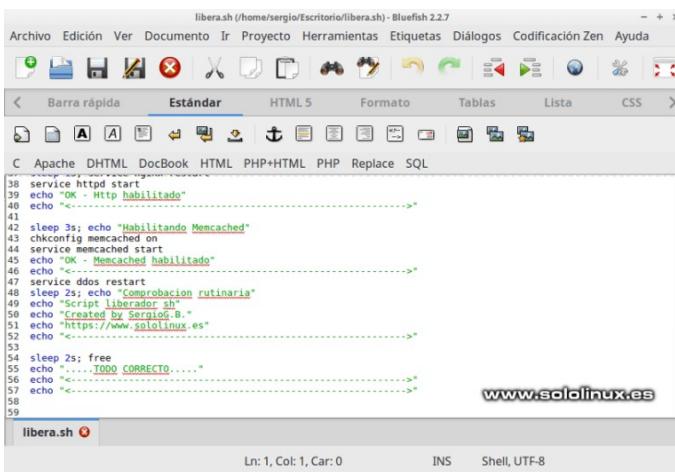
Arch Linux, Manjaro, y derivados:
`sudo pacman -S notepadqq`

Open Suse, Suse, y derivados:
`sudo zypper in notepadqq`

Paquete snap:
`sudo snap install --classic notepadqq`

Puedes unirte a su [comunidad en Facebook](#), o revisar su [Github](#).

Bluefish



Compatible con la mayoría de lenguajes de programación, bluefish es una aplicación ligera pero muy potente. Creada para programadores y diseñadores web experimentados, es muy fácil de usar y su curva de aprendizaje es prácticamente nula.

Altamente recomendada para desarrolladores web, Bluefish viene por defecto en los repositorios de la mayoría de distribuciones linux.

Por ejemplo en Debian, Ubuntu, Linux Mint y derivados, lo instalamos así...

[sudo apt install bluefish](#)

También puedes agregar el repositorio.

[sudo add-apt-repository ppa:klaus-vormweg/bluefish](#)

[sudo apt update](#)

[sudo apt install bluefish](#)

Fedora y derivados:

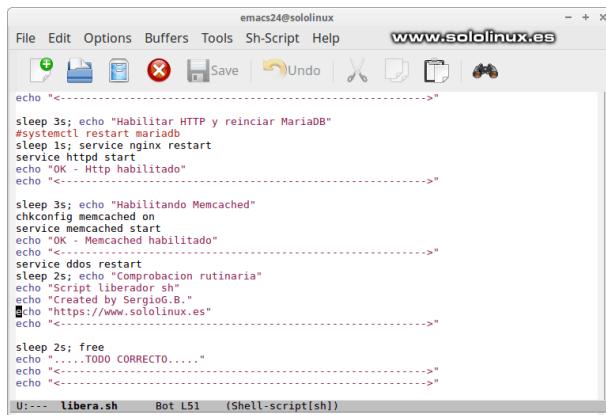
[sudo dnf install bluefish](#)

Arch Linux, Manjaro, y derivados:

[sudo pacman -S bluefish](#)

Visita su [pagina oficial](#)

Emacs



El **editor Emacs** nos ofrece una características impresionantes, como el modo de edición con reconocimiento de contenido, documentación completa incluida en la herramienta, soporte Unicode para casi todos los scripts que seas capaz de programar, código coloreado, y un sistema de empaque para descargar e instalar más extensiones al editor.

Planificador de proyectos, lector de correo y noticias, interfaz de depuración, calendario, etc. Sus funciones y herramientas son interminables. Es considerado por los puristas como el editor más poderoso de los entornos **Unix**.

Este editor lo podemos encontrar en los repositorios oficiales de **cualquier distribución linux**.

Debian, Ubuntu, Linux Mint, y derivados:

[sudo apt install emacs](#)

Fedora y derivados:

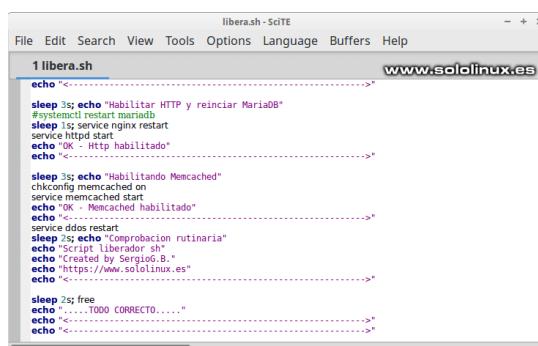
[sudo dnf install emacs](#)

Arch Linux, Manjaro, y derivados:

[sudo pacman -S emacs](#)

Aquí los enlaces a su [pagina oficial](#), y a su documentación.

SciTE



SciTE es un editor multiplataforma, ligero, rápido y altamente configurable. Permite personalizar los lenguajes de programación y los proyectos de forma individual. Soporta multitud de lenguajes de programación.

Instalar SciTE en Debian, Ubuntu, Linux Mint, y derivados:

[sudo apt-get install scite](#)

Arch Linux, Manjaro, y derivados:

[sudo pacman -S scite](#)

Para otras distribuciones puedes descargar el código fuente desde su [zona de descargas](#).

Atom



Atom es un editor moderno y flexible. Permite personalizar cualquier cosa, aunque tal como esta originalmente ya es productivo.

Cuenta con unas excelentes características, como un administrador de paquetes incorporado, autocompletado inteligente, navegador por el sistema de archivos, múltiples paneles, temas y otros ajustes de personalización.

Ahora vemos como instalar Atom en...

Ubuntu, Linux Mint, y derivados:

```
sudo add-apt-repository ppa:webupd8team/atom
sudo apt-get update
sudo apt-get install atom
```

Fedora y derivados:

```
wget https://atom.io/download/rpm -O atom.rpm
sudo dnf localinstall atom.rpm
```

OpenSuse, Suse y derivados:

```
wget https://atom.io/download/rpm -O atom.rpm
sudo zypper localinstall atom.rpm
```

Arch Linux, Manjaro, y derivados:

```
sudo pacman -S atom
```

Paquete snap:

```
sudo snap install atom --classic
```

Visitar la [pagina oficial de Atom](#).

Gedit

```
#!/bin/sh
#OK - Swap habilitado
#systemctl restart mariadb
sleep 1s; service nginx restart
service httpd start
echo "OK - Http habilitado"
echo "<----->"

sleep 3s; echo "Habilitando Memcached"
chkconfig memcached on
service memcached start
echo "OK - Memcached habilitado"
echo "<----->"

service ddos restart
sleep 2s; echo "Comprobacion rutinaria"
echo "script liberador.sh"
echo "Creado by Sergio.B."
echo "https://www.sololinux.es"
echo "<----->"

sleep 2s; free
echo ".....TODO CORRECTO...."
echo "<----->
echo "<----->

sh - Anchura de la pestaña: 8 - Ln 1, Col 1 - INS
```

Gedit es un editor de propósito general, simple y fácil de usar que viene preinstalado en Ubuntu. Pero ojo!!!, es muy potente, ademas incluye herramientas de edición de código fuente incluyendo el texto estructurado, como los lenguajes de marcado.

Incluye resaltado de sintaxis, sangría automática, ajuste de texto, números de línea, resaltado de la línea de trabajo actual, excelente soporte, corrección ortográfica, fuentes y colores que puedes configurar. Si a lo dicho le añadimos su soporte para plugins, hablamos de una buena elección como tu editor favorito.

Gedit viene instalado por defecto en las distribuciones que usan el entorno de escritorio Gnome, por si acaso ponemos también los comandos de instalación.

Ubuntu, Linux Mint, y derivados:

```
sudo apt install gedit
```

Fedora, y derivados:

```
sudo dnf install gedit
```

Arch Linux, Manjaro, y derivados:

```
sudo pacman -S gedit
```

Open Suse, Suse, y derivados:

```
sudo zypper install gedit
```

Página oficial de Gedit Gnome.

Muchos nos dejamos en el tintero, algunos muy buenos como, Geany, Vim, Sublime Text, Kate, y muchos más. Los editores tratados en este artículo quizás son los similares a Notepad++, pero eso no quiere decir que sean los mejores. Todo depende de ti, y de como te amoldes a cada uno de ellos.

Como instalar los plugins de Gedit

Gedit es el editor de texto predeterminado del escritorio GNOME, y no es por casualidad. Ligero, limpio y bonito, hacen de este editor uno de los preferidos de los usuarios de linux. Las características principales de Gedit son:

- Resaltado de código para muchos lenguajes de programación.
- Permite personalizar la paleta de colores del código y las fuentes.
- Resaltado de corchetes, búsqueda y sustitución de expresiones regulares, sangría automática, y mucho más.
- La interfaz de usuario es sencilla y de fácil de usar.
- Compatible con todas las distribuciones linux.

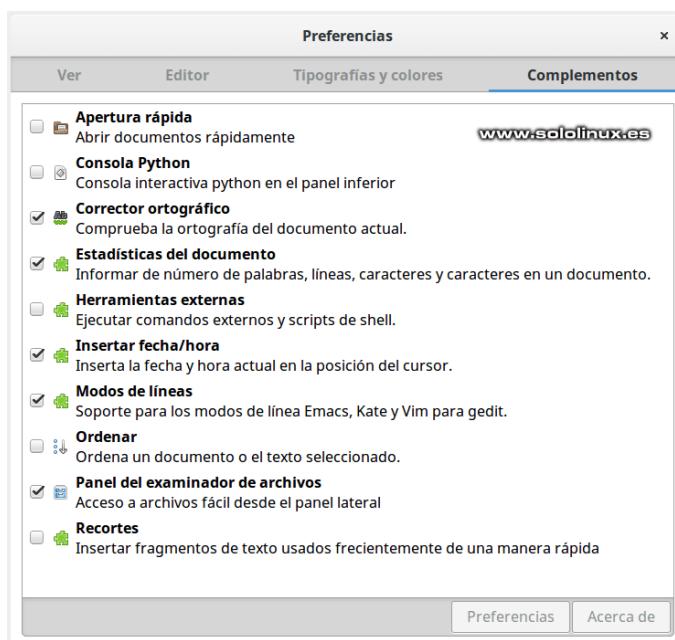
Muchos piensan que **Gedit** no es más que un simple **editor** o visor de texto, bueno, bueno... la verdad es que están equivocados, pues con unos simples pasos podemos convertir **Gedit** en un **editor de programación** muy avanzado.

En este artículo vemos como instalar y habilitar los plugins o complementos de Gedit, para dar un salto de calidad enorme.

Como instalar los plugins de Gedit

Gedit nos permite instalar un montón de complementos y utilidades, que facilitaran la producción de nuestros desarrollos. De manera predeterminada el editor trae muy pocos, para verlos abrimos el **editor Gedit**, pulsa en **Editar y Preferencias**.

En la ventana de preferencias, haces clic en Complementos. Veras algo similar a...



Como puedes ver en la anterior imagen viene con los plugins o complementos indispensables, pero nosotros queremos más, queremos un Gedit poderoso. No te preocupes, ahora instalamos muchos más de una forma simple.

Antes de instalar los plugins debes cerrar el editor Gedit.

Debian, Ubuntu, Linux Mint, y derivados:
`sudo apt install gedit-plugins`

Fedora y derivados:
`sudo dnf install gedit-plugins`

Arch Linux, Manjaro, y derivados:
`sudo pacman -S gedit-plugins`

Open Suse, Suse, y derivados:
`sudo zypper install gedit-plugins`

Ejemplo...

Leyendo lista de paquetes... Hecho

Creando árbol de dependencias

Leyendo la información de estado... Hecho

Se instalarán los siguientes paquetes adicionales:

`gir1.2-glib2-glib-1.0 gir1.2-gucharmap-2.90 gir1.2-zeitgeist-2.0 libgit2-24`

`libgit2-glib-1.0-0 libhttp-parser2.1 libssh2-1`

Paquetes sugeridos:

`zeitgeist-databud`

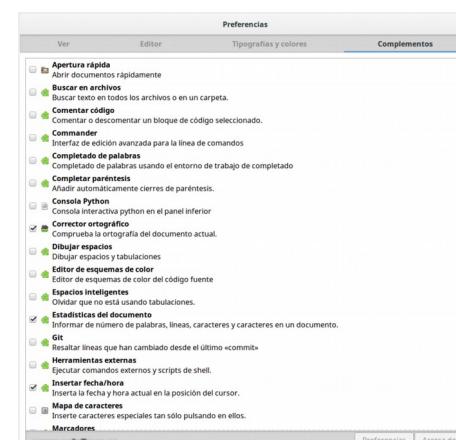
Se instalarán los siguientes paquetes NUEVOS:

`gedit-plugins gir1.2-glib2-glib-1.0 gir1.2-gucharmap-`

`2.90`

`gir1.2-zeitgeist-2.0 libgit2-24 libgit2-glib-1.0-0 libhttp-parser2.1 libssh2-1`

Al concluir la instalación abrimos de nuevo Gedit, pulsamos en Editar y Preferencias, ahora en Complementos. Observa la diferencia en la imagen de ejemplo...



Revisa bien todos los complementos y habilita los que necesites. Al terminar cierras Gedit, y lo abrimos otra vez, los complementos ya estás listos para trabajar.

Por qué Manjaro es tan buena distribución



Manjaro es considerada por la comunidad como la mejor **distribución linux**, para los que se adentran en el mundo de **Arch Linux** (y tienen poco experiencia).

En el artículo de hoy, intentamos explicar los motivos por los cuales **Manjaro** tiene tanto éxito. Aunque todos sabemos que detrás de esta distro hay mucho trabajo para poder combinar el poderoso **Arch Linux**, con una experiencia de usuario increíble.

Por qué Manjaro es tan buena distribución

Puede parecer una broma para los usuarios avanzados, pero el principal motivo del **éxito de Manjaro** es que los noveles tenían problemas para **instalar Arch**, normalmente por desconocimiento.

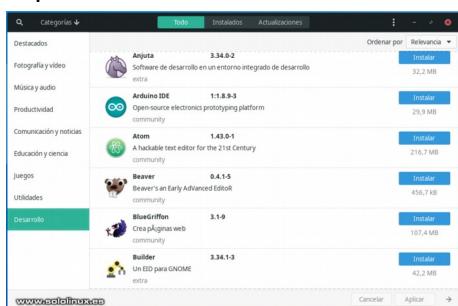
Arch Linux no tiene un instalador gráfico, tampoco viene con ningún **entorno de escritorio** por defecto; Todo debes instalarlo a través de la línea de comandos, y claro... esto provoca que los menos experimentados huyan de **Arch Linux** y se refugien en **Manjaro**.



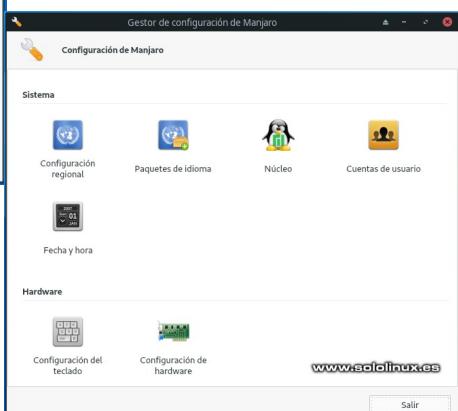
Manjaro Linux tiene un instalador gráfico muy intuitivo, además de muchos entornos gráficos de escritorio.

Otro motivo importante es el tiempo, **Manjaro** no te hará buscar e instalar drivers para tu sistema, él lo hace por ti, ademas siempre son los últimos (estables). Lo mismo sucede con las aplicaciones y herramientas; A través de su **gestor de software** podrás instalar y desinstalar todo tipo de aplicaciones que no encontraras en otra distribución linux.

Destaca porque siempre ofrece las ultimas versiones para producción. Recuerda que **Manjaro se basa en Arch**, por tanto sobran las palabras en este aspecto, como decía un amigo... si no está en Arch, es que no existe.



Manjaro tiene su propia herramienta de «configuración de hardware», que detecta automáticamente todos los componentes de tu maquina, y descarga los controladores necesarios si la intervención del usuario. Como no podía ser menos, también tiene una herramienta para el manejo y prueba de kernels muy similar a **Ukku**, de la cual hablamos en un artículo anterior.



Toneladas de software, excelentes personalizaciones y una estabilidad a prueba de bombas ya son motivos suficientes para probar Manjaro. Si aun no estas convencido, tranquilo, que hay más.

Por ejemplo, la versión GNOME viene precargada con muchos temas para que tu elijas el que más se adapta a tus necesidades. Más de 20 extensiones de GNOME Shell, ajustes de GNOME, Caffeine, Steam, SUSE Studio Writer, Lollypop, Timeshift, uGet y muchas más que cubren nuestras necesidades.

Manjaro es la opción perfecta para usuarios noveles y avanzados.

Esta distro también ofrece una versión un tanto especial, hablamos de Manjaro Architect. Con Architect puedes **instalar Manjaro** desde la CLI, tu eliges los paquetes y las opciones que quieras usar durante la instalación. Manjaro descargara e instalara tu nuevo sistema operativo personalizado de forma rápida y segura (recomendado para evitar cosas innecesarias). .



Resumiendo, esta distribución contiene todo lo necesario para empezar a trabajar de inmediato, no le podemos pedir más. **Manjaro** es una excelente manera de introducirte en el mundo de Arch para aquellos que no quieren tener problemas con sus tecnicismos.

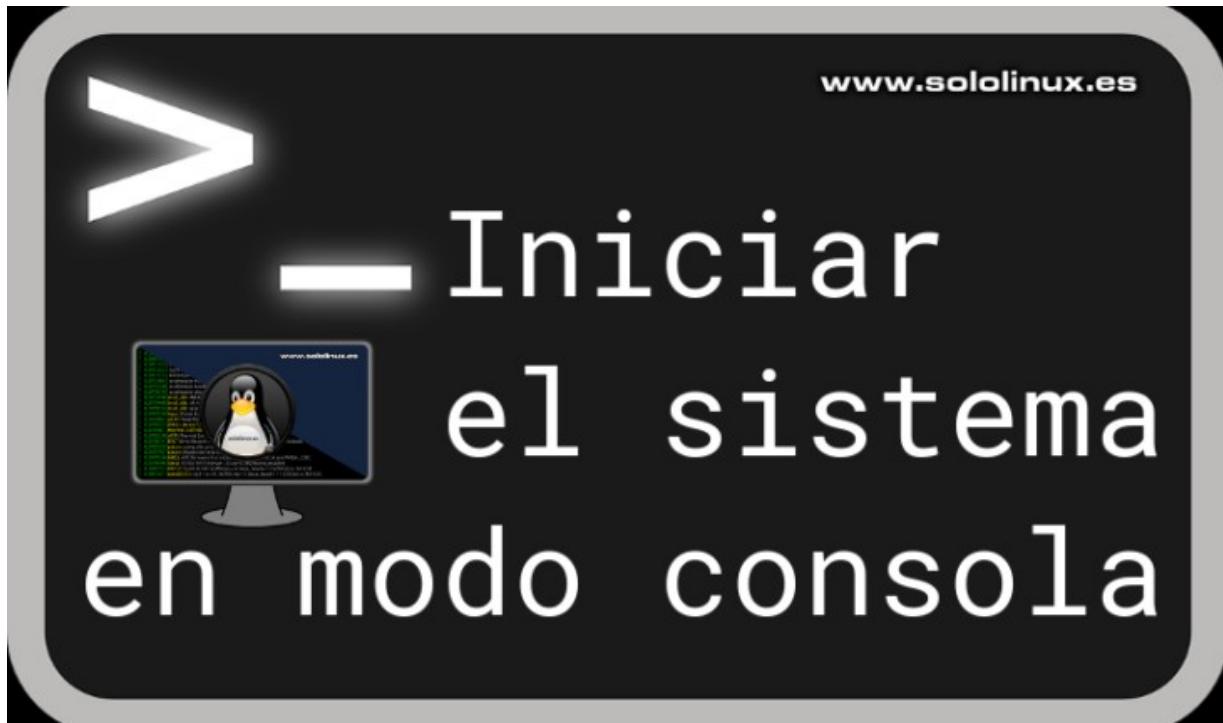
Como punto final al artículo de hoy, insertamos los enlaces oficiales para que puedas descargar Manjaro con tu entorno de escritorio preferido.

- Manjaro XFCE – Oficial
- Manjaro Gnome – Oficial
- Manjaro KDE – Oficial
- Manjaro Awesome – Comunidad
- Manjaro Bspwm – Comunidad
- Manjaro Budgie – Comunidad
- Manjaro Cinnamon – Comunidad
- Manjaro Get I3 – Comunidad
- Manjaro LXDE – Comunidad
- Manjaro LXQt – Comunidad
- Manjaro Mate – Comunidad
- Manjaro OpenBox – Comunidad
- Manjaro Architect – Oficial

Síguenos en las Redes:



Como iniciar el sistema en modo consola



No es extraño que tengamos la necesidad de trabajar exclusivamente en la **consola / terminal**. En esta situación, no necesitamos para nada el consumo de recursos que genera un entorno de escritorio gráfico.

Lo mejor y más cómodo es, que nuestro sistema inicie directamente en la consola de nuestra distribución linux. En este pequeño tip, vemos cómo hacer que nuestro linux siempre arranque en la consola.

El método que propongo funciona en la mayoría de sistemas linux modernos, siendo más específico... en las distros **linux que usan systemd**. Debian, Ubuntu, Linux Mint, Arch, Manjaro, Fedora, CentOS, RHEL, etc, etc, la lista es interminable.

Como iniciar el sistema en modo consola

Con un simple comando, vamos a establecer el modo texto como el inicio de sistema predeterminado de nuestra maquina. Mira que fácil:

```
sudo systemctl set-default multi-user.target
```

Verificamos que el cambio a sido efectivo.
`systemctl get-default`

Si todo está bien, recibirás la salida...
`multi-user.target`

Solo falta reiniciar el sistema.
`reboot`

Como iniciar el sistema en modo gráfico

Puedes revertir los cambios en cualquier momento. Para que linux inicie otra vez en modo gráfico (con tu escritorio favorito), tan solo debes ejecutar lo siguiente...

```
sudo systemctl set-default graphical.target
```

Verificamos que el cambio es correcto.
`systemctl get-default`

La respuesta valida, es
`graphical.target`

Ya puedes reiniciar el sistema y volver a tu escritorio.

```
reboot
```

SolWordpress

SolLinux

Síguenos en las Redes:



Canales de Telegram: [Canal SoloLinux](#) – [Canal Solowordpress](#)

Esta revista es de **distribución gratuita**, si lo consideras oportuno puedes ponerle precio.
Tu también puedes ayudar, contamos con la posibilidad de hacer donaciones para la REVISTA, de manera muy simple a través de **PAYPAL**

AYUDANOS A SEGUIR CRECIENDO

PayPal™
Apoyo con una donación