Slackware 100% Libre

Nunca estuve en el equipo de Software 100% Libre. Utilizo algunos programas propietarios, pero todos los años trato de deshacerme del software propietario, y los chicos que tienen una licencia mala, eso no es gratis. Slackware es una poderosa distribución que viene con un conjunto de software que sin internet está bien abastecido. Y en este conjunto, está presente un software patentado y con licencia que a Stallman no le gusta.

Después de buscar y deambular por un tiempo, encontré una alternativa para poner Slackware en estado Debian. Calma sin apto! Estoy hablando del estado libre en el que Debian viene por defecto, sus repositorios por defecto son MAIN, que a su vez solo tiene software libre.

Entonces, ¿cómo haces esta magia? Simple, usemos Freenix, anteriormente Freeslack, que está basado en Slackware pero es 100% gratuito. Freenix tiene una lista de paquetes no libres tomados de la base de Slackware. Usemos esta base para hacer que nuestro Slackware sea 100% Libre.

Para comenzar, necesitaremos movernos a través de las listas de repositorios de Slackware y agregar el repositorio de Freenix en la última línea. Como root corremos

```
echo '
# 100% Libre
https://freenix.net/fxp/freeslack64-14.2/
' >> /etc/slackpkg/mirrors
```

Si ya tiene una instalación anterior de Slackware en su disco duro, comente el antiguo repositorio y agregue # delante de él.

Después de eso necesitaremos cambiar / agregar la variable **SLACKKEY** presente en */etc/slackpkg/slackpkg.conf*, encontraremos la variable SLACKKEY en la línea 65. Si la antigua variable SLACKKEY no está comentada, comente. 2 mismas variables en conflicto.

```
SLACKKEY="Connie Dobbs (Free eXpansion Pack for Slackware)"
```

Después de eso, necesitamos actualizar la clave GPG del proyecto y luego actualizar la lista de repositorios.

```
slackpkg update gpg && slackpkg update
```

El repositorio está activo y ahora es el momento más divertido para desinstalar software con licencia no libre que no es del agrado de la FSF. Como diría Julio Neves, si está en una línea, es aún mejor. Y hagamos eso. ¡Ejecutemos el comando removepkg con algunos paquetes!

```
**removepkg**               \
  getty-ps             \
  lha                  \
  unarj                \
  amp                  \
  seamonkey-solibs     \
  bluez-firmware       \
```

```
  ipw2100               \
  ipw2200               \
  trn                   \
  zd1211-firmware       \
  font-bh-ttf           \
  font-bh-type1         \
  mozilla-thunderbird   \
  mozilla-firefox       \
  seamonkey             \
  xfractint             \
  xgames                \
  xv                    \
  kernel-firmware       \
  kernel-generic        \
  kernel-huge           \
  kernel-modules        \
  kernel-headers        \
  kernel-source
```

Sí, mozilla es gratuito pero permite complementos no gratuitos. Por lo tanto, estará sin navegador. Puede compilar e instalar Icecat, presente en Slackbuilds;) O incluso seguir con Firefox. La decisión es tuya ... Solo quítala de la lista. Pero no será 100% gratuito. ¡Una nota! Icecat necesita un procesador potente para construir, su doble núcleo no funcionará. Lamento informarle, si usted es del equipo "Pc Débil", use un paquete precompilado, busque este paquete en el sitio web de Slackonly.

El siguiente paso es instalar Linux Libre, el núcleo de Linux que se ha eliminado de todos los blobs. Hagamos esto con slackpkg, ahora nos estamos retirando del repositorio de Freenix :)

```
slackpkg install        \
   linux-libre-headers  \
   linux-libre-image    \
   linux-libre-source
```

Paso hecho? Ok, estamos en el tramo final, ahora necesitamos apuntar nuestro nuevo kernel a lilo.conf. Pero Linux Libre Generic no funcionará, todavía no ... Tendremos que apuntar primero a Huge, reiniciar la máquina y luego generar los módulos y finalmente hacer el punto en lilo.conf.

Abra *lilo.conf* y agregue al final de la línea:

```
image = /boot/vmlinuz-huge-4.4.172-gnu
  root = /dev/sda1
  label = Huge-Libre
  read-only
```

¡Tenga en cuenta que image = call cambiará este artículo en el futuro! Entonces haga la llamada de acuerdo con el núcleo enorme del sistema. Liste en / boot / y vea a qué archivo apuntar.

Ejecute lilo y reinicie la máquina.

```
lilo
```

Ok, después de reiniciar la máquina y si NO le ha dado un kernel de pánico, lo cual es un poco difícil si siguió este CÓMO, es hora de cargar Kernel Generic. Usemos el script en */usr/share/mkinitrd*.

```
**/usr/share/mkinitrd/mkinitrd_command_generator.sh**
mkinitrd -c -k 4.4.172 -f ext4 -r /dev/sda1 -m
hid-logitech-hidpp:hid-lenovo:hid-microsoft:hid_multitouch:jbd2:mbcache:ext4
-u -o /boot/initrd.gz
```

La salida será para módulos que se pueden cargar en mi máquina. El tuyo será variable. Ejecute el mkinitrd que se generó.

```
mkinitrd -c -k 4.4.172 -f ext4 -r /dev/sda1 -m
hid-logitech-hidpp:hid-lenovo:hid-microsoft:hid_multitouch:jbd2:mbcache:ext4
-u -o /boot/initrd.gz
```

Después de eso, haga que el script vuelva a llamar, pero esta vez señalando el núcleo genérico en /boot/. Usa el parametro *-l* .

```
/usr/share/mkinitrd/mkinitrd_command_generator.sh \
-l /boot/vmlinuz-generic-4.4.172-gnu
```

La salida será con una configuración para lilo.conf, agregue en lilo.conf la salida generada, luego ejecute lilo para obtener las nuevas llamadas del menú.

```
image = /boot/vmlinuz-generic-4.4.172-gnu
  initrd = /boot/initrd.gz
  root = /dev/sda1
  label = Generic-Libre
  read-only
```

```
lilo
```

Sources

- Original source: http://slackjeff.com.br/artigos/slackware_free.html
- Originally written by Slackjeff
- — *M3rsy 2020/01/13 16:54 (UTC)*

howtos, slackware, libre, 100% libre, author slackjeff

From:
https://docs.slackware.com/ - **SlackDocs**

Permanent link:
https://docs.slackware.com/es:howtos:slackware:slackware_libre

Last update: **2020/01/13 17:05 (UTC)**