

Slackware 100% Libre

Eu nunca fui do time 100% Software livre. Uso alguns programas que são proprietários mas a cada ano que passa eu tento me desfazer ao máximo de softwares proprietários, e de caras que tem uma licença RUIM, que não seja livre. O Slackware é uma distribuição potente e que vem com um conjunto de softwares que sem internet você está bem abastecido. E nesse conjunto alguns softwares proprietários e com licenças que não são do agrado do Stallman estão presentes.

Depois de procurar e vagar por um tempo eu achei uma alternativa para por o Slackware em estado Debian. Calma, sem apt! Estou falando do estado livre que o Debian vem por padrão, os repositórios do mesmo é padronizado com o MAIN, que por sua vez só possui software livre.

Então como fazer essa mágica? Simples, vamos usar como base o Freenix, antigo Freeslack, que é baseado em Slackware mas 100% Libre. O Freenix possui uma lista de pacotes não livres retirados da base do Slackware. Vamos usar esta base para tornar nosso Slackware 100% Libre.

Para começar vamos precisar mexer na listas de repositório do Slackware e adicionar na última linha o repositório do Freenix. Como root executamos:

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

Se caso você já tem uma instalação mais antiga do Slackware me seu HD, comente o antigo repositório e adicione # na frente do mesmo.

Após isso vamos precisar alterar/adicionar a variável SLACKKEY presente em /etc/slackpkg/slackpkg.conf, vamos achar a variável SLACKKEY na linha 65. Se caso a antiga variável SLACKKEY não está comentada, comente. 2 mesmas variáveis entram em conflito.

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

Após feito isso, precisamos atualizar a chave GPG do projeto em seguida atualizar a lista de repositório

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

Repositório está em pé e agora chegou a hora mais divertida, desinstalar softwares não livres e com licenças que não são bem vistas pela FSF. Como diria Júlio Neves, se for em uma linha é melhor ainda. E vamos fazer exatamente isso. Vamos rodar o comando removepkg com alguns pacotes!

```
**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    \  
xfreetype    \  
xgames       \  
xv           \  
kernel-firmware \  
kernel-generic \  
kernel-huge   \  
kernel-modules \  
kernel-headers \  
kernel-source
```

Sim, o mozilla é livre mas permite plugins não livres. Por tanto você vai ficar sem browser. Você pode compilar e instalar o Icecat, presente no Slackbuilds ;) Ou até mesmo ficar com o Firefox. A decisão é sua... Só remova da lista. Porém não ficará 100% Livre. Uma nota! O Icecat precisa de um processador potente para compilar, seu dual-core não vai dar conta do recado. Sinto-lhe informar, se você é do time "Pc Fraco" use um pacote pré compilado, busque no site do Slackonly esse pacote.

Próxima etapa é instalar o Linux Libre, o kernel Linux que foi removido todos os Blobs. Vamos fazer isso com o slackpkg, agora estamos puxando do repositório do Freenix :)

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

Passo feito? Ok, estamos na reta final, precisamos agora apontar o nosso novo kernel no lilo.conf. Porém o Linux Libre Generic não vai rolar, não por enquanto... Vamos precisar apontar o Huge primeiramente, reiniciar a máquina para depois gerar os módulos e finalmente fazer o apontamento no lilo.conf.

Abra o *lilo.conf* e adicione no final da linha:

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

Perceba que na chamada da `image =` mudará futuramente a este artigo! Por isso faça a chamada de acordo com o kernel huge do sistema. Faça a listagem em `/boot/` e veja qual arquivo apontar.

Execute o lilo e reinicie a máquina.

lilo

Ok, depois que a máquina foi reiniciada e se NÃO deu um kernel panic, o que é meio difícil se você seguiu este HOW-TO, chegou a hora de fazer o carregamento do Kernel Generic. Vamos usar o script presente em `/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
```

A saída será para os módulos que podem ser carregados na minha máquina. A sua vai ser variável. Execute o `mkinitrd` que foi gerado.

```
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
```

Após isso faça a chamada novamente do script, mas dessa vez apontando o kernel generic em `/boot/`. Utilize o parâmetro `-l`

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

A saída será com uma configuração para o `lilo.conf`, adicione em `lilo.conf` a saída gerada, em seguida execute o `lilo` para pegar as novas chamadas no menu.

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

lilo

Fontes

- Original source: http://slackjeff.com.br/artigos/slackware_free.html
- Originally written by [Slackjeff](#)

[howtos](#), [slackware](#), [libre](#), [100% libre](#), [author slackjeff](#)

From:

<https://docs.slackware.com/> - **SlackDocs**

Permanent link:

https://docs.slackware.com/pt-br:howtos:slackware:slackware_libre

Last update: **2020/01/02 18:33 (UTC)**

