

---

Subject: Slackware-Libre BASH script suite  
Posted by [KRT1](#) on Thu, 15 Jun 2023 01:18:31 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

DISCLAIMER: This is mostly the result of my wanting to learn more BASH scripting. It is not intended to be official or supported or endorsed by anyone at all. Use at your own risk. If you like it, feel free to use it. If you want to change it, feel free to fork it. If you think it's wrong, or bad, or stupid, then feel free to ignore it. I just wanted to put this out there so that I can officially move onto my next learning project.

Slackware-Libre is a set of scripts designed to liberate a stock Slackware-15.0 system. Works on both 32- and 64-bit systems.

<https://git.disroot.org/shelldweller/Slackware-Libre>

I've tested this in 32- and 64-bit Slackware 15.0 virtual machines. From a stock installation, just execute, as root:

```
# git clone https://git.disroot.org/shelldweller/Slackware-Libre.git
# cd Slackware-Libre
# chmod a+x *.sh
# ./slackware-libre.sh
```

In a few hours (2 to 5 in my test virtual machines, depending on architecture, tested on 4th gen Intel i5) you will have a completely liberated system, ready to reboot. You'll also have the packages and SlackBuilds that result from the process left over in /tmp, if you want to use those for anything else.

This requires around 4GB of free space to complete. More on 32-bit because of the smp and non-smp kernels.

The main script slackware-libre.sh runs the other three with the necessary arguments to do the full conversion.

The script remove-blacklist-nonfree-packages.sh removes the list of nonfree packages as determined in this thread. It also adds them to the blacklist. This can be run on non-X86 systems (ARM, RISCV, etc). The script can be edited to exclude specific packages from this process.

The script linux-libre-kernel-builder-installer.sh builds (and optionally installs) the latest Linux-libre kernel packages based on the most recent kernel-config from the stock Slackware system. It modifies Pat's stock SlackBuild scripts to use the Linux-Libre source code. Can be run independently to build packages only without installing them.

The script linux-libre-firmware-builder-installer.sh builds (and optionally installs) the latest Linux-libre firmware package. It modifies Pat's stock SlackBuild script to use the Linux-Libre firmware code source. Can be run independently to build package only without installing it.

It's a work in progress, in the sense that I'm sure it can be optimized in many ways. It probably would not work with -current very well, for example. That might come later. But it does the job

well enough to share with the world.

It does NOT rebuild any stock packages or address any of the branding issues (both of which I think are important and worthy topics to address), so it does NOT help with the FSF-endorsement issue by any means. This is just something to play around with until the full Freenix rebuild and rebranding project occurs.

Oh, and of course, there are always people who don't have the time or the patience to build from source, or else their machine is underpowered and/or overheats. In that case, I've put some example packages from the process here:

<https://krt.beauxbead.com/projects/slackware-libre/>

I threw packages for arj in there too, just because unarj gets removed and blacklisted. Why not? Those may or may not be kept updated, depending. My main Libre machine has an Atom processor, so I would build specifically for that instruction set, and not the generic instruction sets that are used in the stock kernel configs. In the case that I do update those files (no promises), there's an atom.php file in the respective directories that will alert a feed reader when new files have been uploaded.

That's all. Have fun. Thanks for listening. Keep the faith.

---