Building a Slackware Offline Installation Kit

Patrick G Horneker

July 5, 2014

Introduction

Since the days of MS-DOS and CP/M[1], the operating system has been an important component of every personal computer, mainframe, and (these days) mobile device. As personal computers came with operating systems already installed, we tended to not worry about that detail, that is, until something happens such that our computer no longer works as it is supposed to.

We worry about backing up the data on our computers in case we have to restore accidentally erased or corrupted files. But how many of us think to worry about reinstalling the software, or worse, the entire operating system in case of accidental erasure or corruption (or perhaps something much worse such as a hard disk failure), or if we want to upgrade the hard disk in our computer?

Before the advent of cloud services, smartphones and tablets, we typically created restore disks, usually with a utility and a pack of floppy disks and/or blank CDs or DVDs.

*For those of you who are old enough to remember, **floppy disks** are called that as the original medium for data storage was a disc made of thin magnetic material and encased in a jacket so thin, the feel of the product was floppy, hence the term floppy disc. When it came to the design of the 3.5 inch floppy disc, the enclosure was made of a hard plastic to ensure durability, especially when ejected from a Macintosh whose 3.5 inch drive automatically ejected the disc when you tell it to from the desktop.*

With the introduction of Linux to the mainstream market back in 1998, creating installation discs was a common practice as it was needed to replace the current operating system with Linux (or install Linux on the same machine as Windows (or OS/2, which was a viable alternative for personal computers at that time).

The introduction of Linux to the mainstream market changed the way software was distributed. Floppy disks were used for distribution with the early distributions. This was particularly true of Slackware and Red Hat (pre-Fedora and RHEL). When Linux was available as a retail product, a single floppy was used to boot Linux for machines that did not boot from a CD/DVD drive, with the distribution available on CD sets or a single DVD. Today, you can purchase magazines such as Linux Journal (http://www.linuxjournal.com), LinuxVoice (http://www.linuxvoice.com) or Linux Format (http://www.linuxformat.com) and get a distribution on DVD. But what you get on distributions today is only a small subset of what is available in Linux distributions. The rest have to be downloaded from repositories located on the Internet.

Here is where some issues come to light. If you need to reinstall Linux, not only do you need the installation CD/DVD, you also need to reinstall **all** software you added to the distribution from repositories. This, of course, takes significant time and network bandwidth as you need to access the Internet to restore

[1] http://www.digitalresearch.biz/CPM.HTM

your system. At that point, you would wish you had all the software available locally so you would not have to go on the Internet every time you need to reinstall software.

Thankfully, there are ways to get around this problem. One way is to order a copy of your distribution's repository from OSDisc.com, (for a fee of course), or configure your distribution's package manager to retain packages after downloading, so you can copy the downloaded packages to a empty directory.

Fortunately, Slackware allows you to store software packages you download using the latter method. This is because *you create the packages locally*, rather than having someone else do the job and uploading packages to the repositories. The Slackbuilds website[2] makes this task easier. For each package you intend to install, you will need to download two files, the source package, and the build package. Move both the source and build package into an empty directory, then unpack the build package first. This creates a new directory containing a script to build the package and supporting files. Move the source package into the new directory, then change into the directory and run the build script (the file named with the package name appended with `.Slackbuild`) [3].

When finished, the resulting package is available in `/tmp` on your system. You can then install the package using `installpkg`. Here is where the idea for creating an offline installation kit comes in. Simply create a empty directory and copy the resulting packages into this directory. You just created a local repository of software installed on your machine. From here, you can backup the repository to DVD media or to a USB flash drive.

It is here where we get the idea for a offline installation kit you can use not only to restore Slackware, but also to replicate Slackware on all machines on your network efficiently, and *without having to go on the Internet* to restore (or install Slackware on) your system.

What We Need

Obviously, to create the offline installation kit, we need a functional Slackware installation, on a machine with a CD/DVD/Bluray Writer, and at least one available USB port (preferably one with USB 2 or USB 3 specifications). You may use a external DVD writer *if you can also boot your system from that device.* In addition to the Slackware machine, we will need the following:

- One blank DVD, any DVD specification (i.e. DVD-R, DVD+R, DVD-RW, DVD+RW, etc.), or a pack (of at least 10) CD-R or CD-RW discs for creation of the Slackware installation disc.
- A USB flash drive of at least 8GB capacity. *You will need at least this much storage* to backup your source and binary packages you download and/or create. The larger the capacity of the flash drive, the better.
- A reliable (and fast) internet connection. You should have at least a DSL connection, and a cable connection is preferred for best results. Mobile broadband connections are not recommended *due to the limitations of your mobile broadband plan*[4]. Dialup connections are strongly not recommended due to the (painfully) slow speed of 56KB modems. (Who uses this type of connection these days?)

Why not use blank DVDs to backup the source and binary software packages? If we need to update the installation kit, we will need to use additional discs if we use DVD-R and DVD+R discs. DVD-RW discs

[2] http://www.slackbuilds.org

[3] You must be logged in as *root* to perform this operation as packages are built in the `/tmp` directory

[4] Mobile broadband plans were never intended to be used for software updates, even though IMO they should. Even iPhones and Android devices need to be updated every so often.

require reformatting. However, DVD+R discs can be rewritten to simply by writing over the old disc. Even then, the capacity of a DVD+RW is only 4.7GB per disc. If you have many packages that you have installed, 4.7GB may not be enough to backup both the binary and source packages. Also, it is much easier to backup files to a flash drive. You simply copy the files to the flash drive, rather than going through the process of creating a ISO image, then burning the image to a DVD. To update a file on a flash drive, remove the old file from the flash drive and then copy the new file.

Download a Slackware Installation Disc

Now that we have the materials ready, the first thing we need to do is to download the latest Slackware ISO image. There are five ways of downloading the ISO.

- Download direct from the Slackware website (or one if its mirrors). Be sure to download the `install` ISO images, and not the `source` ISO images.
- Download using an FTP client connected to **ftp://ftp.slackware.com/pub/slackware-iso/**
  Be sure to download the `install` ISO images, and not the `source` ISO images.
- Download using Bittorrent or another torrent client
- Purchase a book about Slackware, either from Slackware's online shop, or order it from Amazon or Barnes and Noble. These books contain a release of Slackware. *Note: Not all books about Slackware contain the latest release, so be aware of what you are purchasing.*
- Purchase a DVD disc online from **OSDisc.com** [5]or another online retailer. These discs are inexpensive to obtain, and if you access your Internet through a dialup connection, or a mobile broadband connection[6], this is your best (and maybe your only) option.

As of this writing, **version 14.1** is the current release. I prefer the single DVD image as opposed to six CD images. This means you only have to burn and test one DVD rather than burning and testing six CDs. Also, the DVD image contains the entire installation installable at one time, rather than having to change the CD six times.

Do You Need 32-bit or 64-bit For Your Machine?

Slackware comes in 32-bit or 64-bit variants. If you are not sure whether your machine supports a 64-bit operating system[7], then go with the 32-bit variant as the 32-bit variant works well with 32-bit and 64-bit processors. However, if you are sure your machine can accomodate a 64-bit operating system, by all means, download the 64-bit version.

Special Consideration for 64-bit Slackware: Many 64-bit Linux distributions come with software packages that provide compatability with 32-bit applications. Slackware 64, however, *does not come with that compatibility on the installation disc.* You can, however, add additional packages[8]

[5] I recommend this retailer for purchasing installation discs as they have very good service, and the turnaround time from ordering to receiving of the disc is what you would expect from mail order. Also, the discs received are very reliable.

[6] Refer to the data plan associated with the wireless service you are using for details. Wireless providers typically charge for **how much data you download/upload** rather than how long you are on the Internet. Also, many plans I have come across do not provide enough available data to update your Slackware system (if and when you need to).

[7] Since 2007, all new desktops and laptops on the market support 64-bit operating systems. You can use either a 32-bit or a 64-bit Slackware disc on these machines.

[8] http://alien.slackbook.org/dokuwiki/doku.php?id=slackware:multilib

Torrent vs. Direct Download

It has been argued that the use of torrents (peer to peer networking) is a efficient way of downloading DVD images. While this may be true, what is also true is the perception of torrents in the mainstream media is that *torrents are used only for pirating of copyright material.* There are legitimate uses of torrent, and downloading of Slackware (and other Linux distribution) DVD images is one such legitimate use.

Direct downloading of ISO images from FTP or Web servers is preferred **if you care about your privacy**. In recent days providers of torrent services have been targeted for surveillance, primarily for detection of piracy of copyright material. Depending on the speed of your Internet connection, direct download of Slackware ISO images can take as little as one-half an hour (on a cable connection) to seven hours (on a busy DSL connection). The time is well worth it, not only will privacy be assured, but so will the quality of the resulting ISO image.

Create a DVD From the Downloaded ISO Image

Slackware comes with K3B for burning of disk images. IMO, this is the best and most reliable way to create a DVD for installation of Slackware. Since this is a offline installation kit we are creating, we are not really going to use it, unless we are also planning to install Slackware on another machine. When you downloaded the disk image, the ISO image should be in the *Downloads* directory in your user account. Type `cd ~/Downloads` on a command line to get to that directory. If you used Bittorrent, your disk image is in `~/Slackware-14.1-iso`.

The `cdrecord` command also performs the same task. K3B uses this command and takes care of all the details about writing to blank DVD medium. The current version of `cdrecord` now supports Bluray writers, in addition to most CD and DVD writers. We only need to concern ourselves with the DVD image as that is what we will be writing to a blank DVD.

If you are running the XFCE, LXDE, KDE, Mate, or GNOME (Dropline), you should use **K3B** or **Brasero** (if installed) to burn the disc image. You can also run K3B from WindowMaker, Fluxbox, Blackbox, FVWM (also available on Slackware), or any other desktop you have installed.

If you choose to use the command line, unless you have an CD burner with a proprietary interface (or on a parallel port), `cdrecord` should automatically detect your CD/DVD/Bluray burner, and you should not need to supply some command line options to `cdrecord`. If that is the case, insert a **blank DVD** into your burner and *then* change to the directory where you downloaded your ISO image, usually `~/Downloads`. Then type **cdrecord slackware-14.1-install-dvd.iso** (or **cdrecord slackware64-14.1-install-dvd.iso** if you are using the 64-bit variant).

Once you have the burned the ISO image to DVD, you have the installation media part of the offline installation kit completed. Test the DVD by rebooting your system with the DVD still in the drive (make sure you can boot from the CD/DVD first). If everything is successful, you should be in the first installtion screen for Slackware.

Alternately, if you have `unetbootin` installed, you can use that to create installation media on a 4GB or larger USB flash drive or memory card. If you choose this option, make sure your system can boot from USB media.

Organize Your Software

For this process, let us assume that we are using a Slackware installation, and that we have a terminal window open on any desktop.

Now that we have a Slackware installation disc (or a USB flash drive or memory card if you chose to use that instead of a DVD), we need to create a local repository of software you have installed. For this, we will need to create an empty directory in which to build this repository. I recommend `Packages` for the name as that is what we will be working with throughout this process. On a command line, type `mkdir Packages`. Then type `cd Packages` to get into the new directory.

Software for Slackware comes from many sources. The first source we shall work with is the Slackware updates. *I recommend separate directories for each of the different sources we will be using to help keep things organized, and to help make backing up of the software easier.* For this example, we shall type `mkdir updates`.

Midnight Commander Can Be Your Friend

Slackware contains an easy to use file manager, named Midnight Commander. Type `mc` to launch this program. Now you have a convenient setup to manage the files on your system. There are two panels available for you to use when managing files. We can use this application to download Slackware updates. Type `cd ~/Packages/updates`. The left panel now shows an empty directory where we are going to download and store Slackware updates. Use the `Tab` key to switch between the two panels. Now type in
cd ftp://ftp.slackware.com/pub/slackware/slackware-14.1/patches/packages.
(If you are using the 64-bit version of Slackware, type
cd ftp://ftp.slackware.com/pub/slackware/slackware64-14.1/patches/packages.)
Now the right panel should show the available updates to the version of Slackware we are using. The pointer should be highlighted at the top of the right panel. Press the down arrow key, then press *Insert* key to select files in the Slackware FTP server. Keep pressing *Insert* until you get to the bottom of the listing. The listing should be in alphabetical order so you will know where the bottom of the list is, and the entire list is highlighted. When finished, press `F5` to copy the updates to your machine in `~/Packages/updates`. This will take some time depending on your Internet connection speed. When finished, press `F10` and confirm to exit Midnight Commander.

The Slackbuild Repository

The next source of software we shall work with is Slackbuilds. This is an unofficial repository for most software you can install on Slackware. Each package comes in two parts, a Slackbuild package, and a source package. **You will need to download one source package and one Slackbuild package for every software package you intend to install**. I recommend making two directories for this source, one named `slackpkg` and one named `build`. `slackpkg` is used to store completed packages that have been installed. `build` is used to store the source packages and Slackbuild packages you download. You will use this directory everytime you update a package from this software source.

To keep track of what has been built, and what has not already been built, I recommend creating two subdirectories within `build` to keep the Slackbuild packages organized, namely `0pending` and `0done`. The prepended zero is necessary so you can easily spot the pending and done packages and distinguish them from subdirectories created when decompressing Slackbuild packages inside `build`.

About Slackbuild Packages

Slackbuild packages, when decompressed, create a subdirectory where you decompress that contains a script that builds the software package as well as any supporting files and patches to get that software working on Slackware. Because Slackbuild packages do not contain the source code for the software, you need to download the source package for each Slackbuild package you download, then insert the package into the subdirectory created when you decompressed the Slackbuild package.

You can easily tell which packages are the Slackbuild packages and which are the source packages by the size of the files that have been downloaded. Slackbuild packages are very small in size (usually less than 10KB).

Here is where applications like Midnight Commander are very useful (type `mc` to access Midnight Commander). Under Midnight Commander, type cd `~/Downloads` to set the left pane to the directory where Slackbuild packages are stored. Then press `Tab` to switch to the right pane. type cd `~/Packages/build`. This is where you want to decompress Slackbuild packages. Press `Tab` again to switch to the Downloads directory.

Suppose we want to install Digikam. We go to Slackbuilds.org and find Digikam. There are two files to download. One file is called **digikam.tar.gz** and the other is called **digikam-4.1.0.tar.bz2**. (Let us assume we already have the dependencies for this package installed for the sake of this example.) You need to download *both* files. These files will appear in the Downloads directory on your user account, i.e. `~/Downloads`. Next, we use the arrow keys to move the pointer to the file called **digikam.tar.gz**. This is the Slackbuild package. Press `Return` to decompress the file. Notice that Midnight Commander decompresses **digikam.tar.gz** without any problems. You get a directory named `digikam`. Now highlight the entry with the down arrow key, and press `F5` and confirm. You have just copied the decompressed contents of the Slackbuild package to the directory you want to use to build Digikam. Now press the up arrow key and return to go back to `Downloads`. Next, move the pointer so **digikam-4.1.0.tar.bz2** is highlighted. Press `Tab` to switch to the right pane. Highlight `digikam`, and press `Return`. Now press `Tab` again to go back to the `Downloads` directory. Press `F6` and confirm to move the source package to the Slackbuild created directory named `digikam`. As we no longer need the downloaded Slackbuild package, we can now delete **digikam.tar.gz**. Highlight that entry and press `F8`. Now press `F10` and confirm to exit.

The next thing we need to do is to build Digikam. Type `cd ~/Packages/build/digikam`. Next, type `su` and press Return. Enter the administrator password to login as the administrator. This step is necessary as packages are built in the `/tmp` (for temporary) directory. Then type `./digikam.Slackbuild` to build the package.

If everything succeeds, a package called **/tmp/digikam-4.1.0-1.tgz** will be created.

Here is where the `slackpkg` directory inside `build` comes into play. First, we need to install Digikam. If Digikam is already installed, we type `upgradepkg /tmp/digikam-4.1.0-1.tgz`. Otherwise, we type `installpkg /tmp/digikam-4.1.0-1.tgz`. Once this is done, then we type `mv /tmp/digikam-4.1.0-1.tgz /home/`*`username`*`/Packages/slackpkg` to move the package to the directory containing finished packages. *username* represents your username you used to login to Slackware. This procedure should be followed for *every* software package downloaded from Slackbuilds. The result will be a `slackpkg` directory full of binary packages that can be reinstalled if necessary, rather than rebuilding and then installing every package.

Other Sources of Software

Some packages such as LibreOffice, Eclipse and NetBeans come in archives that do not depend on any particular distribution to install. For these, I suggest creating a directory named `tarball`. The term `tarball` refers to a compressed archive. `tar`, stands for **t**ape **ar**chive, and if such packages were distributed on magnetic tape, the tape could be physically rolled up into a ball, or so it looks like a ball. Of course, magnetic tape would come in a cartridge or on a reel.

LibreOffice

In the case of LibreOffice, this is a huge (as of this writing a whopping 175MB for the distribution, and that is without the help files) file to download. What you get is a package of RPM packages. RPM stands for Red Hat Package Manager, and was intended to be used with distributions that use RPM for package management (e.g. PCLinuxOS, OpenSuSE, Fedora, and CentOS/RHEL/Oracle). Slackware can use these files.

After uncompressing the files (to `~/Packages` directory), there is a long named subdirectory called **LibreOffice_*version*_Linux_*platform*_rpm** where *version* is the version release number, and *platform* is either `x86` for 32-bit or `x86-64` for 64-bit. This directory contains a subdirectory called `RPMS`. This is where the actual software is located.

To get to the software, type **cd ~/Packages/LibreOffice_***version*`_Linux_`*platform*`_rpm/RPMS`.

Login as administrator with the **su** command, and then type `rpm2tgz *.rpm ; installpkg *.tgz` to install LibreOffice.

Summary

Once you have populated the directories with all software you wish to have installed, you will have a directory structure that you can then copy to a USB flash drive or memory card with at least 8GB of capacity.

If you are running KDE, Mate, XFCE, LXDE, or Dropline GNOME, simply plug in a empty USB flash drive. The icon representing the flash drive will appear on your desktop. Open the `Home` directory (by double clicking or single clicking depending on your desktop), then open the icon representing the flash drive. Two windows will appear. One represents your home directory. The other represents the flash drive. From the window representing your home directory, hold down the `Control` key, then with your mouse, click and drag the folder labelled `Packages` to the window representing your flash drive. The contents of Packages should copy to your flash drive.

Watch the progress bar on the screen as the files get copied.

When copying is finished, **you will have completed the second half of your offline installation kit**. Close the window representing the flash drive, then with the other window, click on the down arrow next to the name assigned to the flash drive to unmount the flash drive. Once the icon disappears, you can then remove the flash drive. For KDE, select the device manager applet and click on the down arrow to the right of the device name to unmount the flash drive. KDE will let you know the device can be removed.

If you are using a command line, you will need to do the following to backup the contents of the `Packages` directory to a USB flash drive.

First, plugin the USB flash drive. Then type **su** to login as administrator.

Next, type `dmesg` and read the last lines of the output. You should see something like this:

```
[17127.770150] usb 2-1: new high-speed USB device number 7 using ehci-pci
[17127.886832] usb 2-1: New USB device found, idVendor=0c76, idProduct=0005
[17127.886842] usb 2-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
```

```
[17127.886850] usb 2-1: Product: USB Flash Disk
[17127.886857] usb 2-1: Manufacturer: 3SYSTEM
[17127.886863] usb 2-1: SerialNumber: 00EA0109207BF9B1A4282BE4
[17127.887432] usb-storage 2-1:1.0: USB Mass Storage device detected
[17127.887606] scsi11 : usb-storage 2-1:1.0
[17128.889890] scsi 11:0:0:0: Direct-Access     3SYSTEM  USB Flash Disk   1.00 PQ: 0 ANSI: 2
[17128.891981] sd 11:0:0:0: [sdc] 8002864 512-byte logical blocks: (4.09 GB/3.81 GiB)
[17128.892825] sd 11:0:0:0: [sdc] Write Protect is off
[17128.892835] sd 11:0:0:0: [sdc] Mode Sense: 03 00 00 00
[17128.893455] sd 11:0:0:0: [sdc] No Caching mode page found
[17128.893464] sd 11:0:0:0: [sdc] Assuming drive cache: write through
[17128.896572] sd 11:0:0:0: [sdc] No Caching mode page found
[17128.896582] sd 11:0:0:0: [sdc] Assuming drive cache: write through
[17128.897493]  sdc: sdc1
[17128.900227] sd 11:0:0:0: [sdc] No Caching mode page found
[17128.900231] sd 11:0:0:0: [sdc] Assuming drive cache: write through
[17128.900235] sd 11:0:0:0: [sdc] Attached SCSI removable disk
```

In this example, the USB flash drive has been assigned the device name `/dev/sdc` with a single partition, namely `/dev/sdc1`. Your results will vary.

To mount this USB flash drive, you will need to assign a mount point, preferably within either `/mnt` or `/media` to the device `/dev/sdc1` as we will be using the first (and only) partition of the USB flash drive. Slackware thankfully provides some usable mount points in the `/media` directory, namely `/media/memory`, `/media/zip`, and `/media/disk`. Since this is a USB flash drive, it only makes sense to mount the USB flash drive at `/media/memory`.

While still logged in as administrator, type the following:

```
mount -t vfat /dev/sdc1 /media/memory
```

You are then ready to start the backup. Type

```
cp -Rv /home/username/Packages/ /media/memory
```

to copy the entire contents of `Packages` to your USB flash drive. When copying is finished, you will need to unmount the USB flash drive. You accomplish this by typing:

```
umount /dev/sdc1
```

or

```
umount /media/memory
```

At that point, you have backed up the entire contents of your `Packages` directory. Your basic offline installation kit is now complete, with an installation disc and a flash drive with supplementary software.

Web Resources Used

- Slackware (http://www.slackware.com)
- Slackbuilds (http://www.slackbuilds.org)
- 32-Bit Compatibility for Slackware 64 (http://alien.slackbook.org/dokuwiki/doku.php?id=slackware:multilib)

Legal Information

- Slackware is a registered trademark of Patrick Volkerding.
- MS-DOS and Windows are registered trademarks of Microsoft Corporation
- OS/2 is a registered trademark of International Business Machines Corporation and Serenity Systems (the current owner)
- CP/M is a registered trademark of Caldera Systems
- iPhone is a registered trademark of Apple, Inc
- Android is a registered trademark of Google