Run Linux from USB Flash Drive

Do you want to run Linux anytime, anywhere? In this article, we will show you what you can do with popular Linux distributions like Puppy Linux, Ubuntu, and Fedora to be able to launch directly from a USB drive.

Serdar Yegulalp

Do you want to run Linux anytime, anywhere? In this article, we will show you how to run popular Linux distributions such as Puppy Linux, Ubuntu, and Fedora directly from a USB drive.

Linux is almost always run by installing it on a hard drive or on a CD/DVD. The first way is fast but not portable, the second way can run anywhere you have a computer and a CD drive, but it is not fast. However, in recent years, a way has begun to appear that combines the speed of a hard drive with the convenience of a CD: running Linux from a USB flash drive.

While flash memory is still quite expensive, a 100GB flash hard drive may be out of reach for many people, but lower capacity drives, such as 2GB or 4GB, are not out of reach or even affordable for almost anyone. Let's see what you can do with these USB drives in the context of the problem presented in this article. It may not be

enough to carry your entire MP3 collection and large photo albums, but you will be able to run Linux with most of the necessary applications, as well as carry some other data. The other thing to note here is that there are ways to run Linux from a USB flash drive without rebooting the operating system, especially if you are running Windows. What you

need to do

To install Linux on a USB drive and run it, you'll need the following:

1. A Linux distribution that supports installing and running from a USB drive. You can rest assured that most recent distributions will allow this, although the exact steps to get it running may vary slightly. Some distributions will have the ability to do this by default.

2. A USB flash drive, not all USB drives are created equal, but we'll discuss this below.

3. A computer that can boot from a USB port. This is a tricky one for some of you because it depends on the computer – very few recent computers support booting from a USB drive.

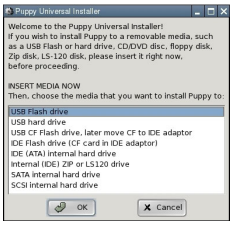
If you set up a Linux installation to run from within Windows itself, you don't need USB boot support, you just need access to the hard drive plugged in via USB. Note that some computers in many environments – for example, in corporate environments – may be managed to prevent users from using USB storage devices.

Which Linux distribution to use?

The distribution you choose depends on a few things: how big the flash drive is, what Linux distribution you are familiar with, and what features you want to support. Drive size

is a factor, but it's not really something you should worry about because it can be based on your budget. Flash drives are really cheap these days, a 1GB USB stick will do just fine for your Linux distribution.

What you do care about is how small and lightweight a Linux distribution is designed to be. Two good Linux distributions in this regard are **DamnSmallLinux** and **Puppy Linux** .



Puppy Linux's installer supports everything from USB drives to flash card interfacing.

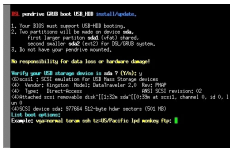
In general, you will need a USB with a capacity of about 2-4GB to perform an installation with all the functions from the installer. In case there is not enough space for the installation, the system will warn you about it. If you have an installation with a large capacity, it means that it will have more applications and more features supported.

Choose a Flash drive?

Flash drives may seem similar on the surface, but you need to be aware that there are many differences in speed between drives. A drive rated "100x" or higher will work quite well, and most drives that support Vista's ReadyBoost feature will do as well.

The one thing that most manufacturers don't make clear is whether the drive's performance is sequential or random. Does each byte on the drive have the same access time? Sometimes manufacturers will mix memory speeds as a cost-saving measure, using a small block of fast flash alongside a large block of slow flash. This means big changes in performance depending on what's being done in the USB. To determine that, you can see big differences in performance between two drives that have the same or very similar manufacturer ratings.

Another interesting thing to note here is that flash drives don't have to be removable devices, but flash cards, digital cameras, and music players can work just as well. Also, whether Linux or your computer supports booting and running from such a device will depend on how the device presents itself to the computer and whether your Linux distribution has drivers set up to handle it.



DamnSmallLinux "pen drive" installer

Enabling USB Boot Support

Most computers manufactured in recent years support booting from USB devices. That said, it's not always enabled by default, and it's not always enabled for all USB ports on the system.

One of the first places to look for information about USB boot support is the system BIOS. On Dell computers (a specific example), the default USB management option is to enable booting from attached USB devices – but you have to press F12 every time you boot to choose it over the system's default boot setting, which is usually from the hard drive.

Another important thing to note here is that some USB ports in the XPS can be disabled. This is done so that you can add external USB drives (either flash or physical).

Folks who frequent the **PenDriveLinux** Web site —a great resource for all things Linux on USB—have a quick way to test USB boot support for your USB and your computer. They use the SysLinux utility to make the drive bootable and add a copy of Memtest86+, a way to see if boot support is working. Note that if you're using them on Vista, you'll need to open an administrative command prompt to run the makeboot.bat file or it won't work.

SysLinux is widely used to make flash drives bootable for Linux distributions, so it's a tool we'll cover here. Note that if you encounter a "boot error" or other error message when trying to boot a drive made with SysLinux, you'll need to run the **DISKPART CLEAN** command on the disk (again from the administrative command line) to clear the master boot record information, reformat the drive, and try again.

From installation CD to bootable flash drive

Most Linux distributions, like Ubuntu, are capable of installing directly from a USB drive as if it were a hard drive, but some Linux distributions work with USB drives as a special case. DSL, for example, has a special installer for creating USB-drive installations of the operating system. It can also install itself directly to a USB drive using the included Universal Installer utility. The default options didn't work in this case, though; we had to use the "SysLinux" option in its settings menu to make the USB bootable.

In most cases, the USB drive will be mounted and recognized as a virtual SCSI device, such as */dev/sda or /dev/sdb* . If you are installing on a system that already has a hard drive, you need to pay close attention to the devices listed in the partition manager and make sure you are installing to the correct one. Otherwise, you might accidentally erase the hard drive. You also need to make sure that the installer makes the target device bootable and writes the appropriate master boot record to the disk, although most of the time this is done automatically.

From Live .ISO to Bootable Flash Drive

If the Linux distribution you're considering has a live CD (most do), it's also possible to mount the .ISO for that live CD on a flash drive and run the program from there. The advantage of this is that it's quick, and most machines that can boot from a CD can also boot from a flash drive. The advantage

of this is that a live CD is not persistent by default. Any changes you make to the system will be lost the next time you reboot, unless you make a backup of your user data. Some live distributions allow this, but it's not always guaranteed. If you want a live installation that leaves no trace—for example, for safe browsing or system restore—this is the way to go.

Some PenDriveLinux members have created scripts that can automate the process for popular distributions like **Ubuntu 8.04** and **PCLinuxOS** . Their trick (which is quite clever) is to use the open source 7-Zip compression application to browse the .ISO and extract the main files from it to allow SysLinux to create a bootable drive.

The basic techniques can be applied to any Linux distribution running from the ISOLINUX live file system. If you want to create a bootable flash drive from the .ISO of a live CD distribution, follow these steps:

1. Format the flash drive. You may need to use **DISKPART CLEAN** as described in the section above. For compatibility, use FAT or FAT32 for the file system. (FAT seems to be the best choice here).

2. Use 7-Zip or another .ISO reader to unzip the entire contents of the .ISO onto the flash drive. Make sure the root directory of the .ISO corresponds to the root directory of the flash drive; do not extract the contents into subdirectories on the flash drive.

3. Use **SYSLINUX -ma** : (in the command line if you are using Vista) to write the system files, with is the current drive letter for the flash drive. If you are running SysLinux from something other than Windows, the *-ma* switches here are not necessary.

4. For many live distros, there is a file in the root directory of the drive labeled *ISOLINUX.CFG* . This label may need to be renamed to *SYSLINUX.CFG* to make sure everything works.

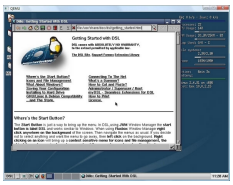
5. Remove the flash drive and try booting it on your target system .

Some live distros of Linux will have a root *boot directory in the .ISO and an isolinux* directory inside it. For example, you will need to move the contents of *bootisolinux*to the root directory of the flash drive before renaming the files and then applying SysLinux. The order of the above events is important.

QEMU Linux

Some readers may be friendly with using a virtual machine to run Linux on another operating system – often a way to test an existing Linux build inside Windows without creating a dual-boot scenario.

The same can be done with a flash-drive version of Linux, courtesy of the open source computer emulation package **QEMU** .



The open source QEMU lets you run Linux in a virtual machine on another operating system

This package is small enough to fit on most flash drives and whatever distribution you're using and can be started by executing a simple batch file. You even get support for connecting to whatever network is running on the host. Here's how to do it.

1. Format the flash drive with FAT or FAT32 (for compatibility purposes).

2. Copy the .ISO file you want to use to the root of the drive

. 3. Extract the QEMU executable to a folder called *QEMU* .

4. Use the following command to run QEMU from the root of the drive (you can edit it into a batch file):

.qemuqemu.exe -L .qemu -cdrom

There are a few other options that can be skipped here. If you run QEMU without the command line switches, you will be presented with a set of available options, from which you can choose whichever suits your needs.

5. Run QEMU inside Windows. The emulated machine will appear in Windows and you can completely release the cursor using the Ctrl+Alt key.

Note that you will not be able to see performance if you boot the system natively. QEMU has a Windows driver to allow some performance effects.

If you want to create persistent storage for a flash drive, there are a few ways to do this. QEMU allows you to place any file on the flash drive as you would a hard drive, so you can just create a large file on the flash drive that will hold your user data, use the *-hd* QEMU option to place it, format the drive in the live Linux session and store your user data there between sessions.

The second option is a little different than the first, except that instead of saving the data on that drive, you install Linux on it and then set up QEMU to boot the hard drive image instead of the .OSI. Note that the best way to do this is to prepare all the files on your computer first, then copy the image and the QEMU executables to the flash drive.

To that end, the PenDriveLinux folks have created a package for you to use that includes QEMU and a batch file with some general options. This batch file can be customized with the command line options you want to use, and can also be changed to match the ones described above.

MokaFive

Another way to run a Linux installation from a flash drive is through **MokaFive** . This is a commercial application that allows you to package virtual machines onto a flash drive for distribution over the network. MokaFive uses VMware to do the emulation, which is generally better than QEMU and also allows for some more sophisticated locking techniques. The basic version of MokaFive is completely free (although not open source) but works very well. It allows you to access devices on the host such as scanners and printers.

MokaFive allows you to create bootable flash drives that can run on emulated operating systems.

The MokaFive toolkit includes the BareMetal player, which is a tool that allows you to install the MokaFive VM onto a flash drive, boot from it, and run it by default. The device must have at least 2GB of free space for this to work. Note that since the BareMetal player is protected against tampering, the only way to get the VM inside is to load it on the BareMetal player interface itself.