How can I mount my iPhone 6s on Ubuntu 16.04?

I'm new to Ubuntu, and I don't know how to mount my iPhone.

I would like to download iTunes, but I saw that USB doesn't work, but I just need to access to all my files and APP because would like to transfer APP from pc to iPhone.

16.04 usb mount iphone

edited Aug 27 at 18:26
Zanna
39.9k 12 90 190

asked Aug 14 '16 at 1:14
Rizio A.
96 1 2 5

Is the phone jailbroken? – fakedad Aug 14 '16 at 1:19

No it is not, fresh as new – Rizio A. Aug 14 '16 at 15:24

As far as I know, there is no way to mount the iPhone's filesystem normally via USB on any OS. Moreover, I don't know of any applications for Ubuntu (or any Linux distro for that matter) that allow the syncing of apps to an iOS device. – fakedad Aug 14 '16 at 20:09

You can try to run Windows in a virtual machine using VirtualBox, and manage apps via iTunes in the VM. – fakedad Aug 14 '16 at 20:11

I read somewhere that it could work installing GKSU NAUTILUS...(honestly, dunno wtf it is), but it gave some error...and cannot download VirtualBox...do you have any link? – Rizio A. Aug 15 '16 at 1:10

3 Answers

Evidently I was wrong about being able to mount an iPhone on Ubuntu. You can perform this using the following steps on yakkety. Note, you *would* need the device to be jailbroken in order to load apps onto the device this way, but this method will suffice for getting media from the device.

Option 1: Using a script

If you want to save yourself some time, you can download a script here to do most of the work of the process for you.

Once downloaded, you will need to change the permisions so you can execute the script. Assuming you downloaded it with the default name, `iphone_setup.sh` , `cd` to the directory in which you downloaded the file and do

```
chmod u+x iphone_setup.sh
```

Convert the Windows line endings by doing

```
ex -bsc '%!awk "{sub(/\r/,\"\")}1"' -cx iphone_setup.sh
```

Then run the script with root privileges using

```
sudo ./iphone_setup.sh
```

This will complete all of **Step 1** of the manual setup for you, as well as **Step 3** and **Step 4**. You will then need to do **Step 2** and **Step 5** of the manual setup after the script finishes running.

Option 2: Doing it manually

Step 1: Installing the tools

Before plugging in the iPhone, you will need to install the several programs to make it possible to mount the iPhone.

Step 1.1: Installing several important tools with apt-get

Do the following in the terminal to install a few packages that will be needed for any version of iOS.

```
sudo apt-get install ideviceinstaller python-imobiledevice libimobiledevice-utils
python-plist usbmuxd
```

If you are connecting an iPhone with an iOS version before iOS 9, you can skip the remaining substeps of step 1 and instead just do the following:

```
sudo apt-get install libimobiledevice6 libplist3 ifuse
```

Step 1.2: Installing tools for building

Use `apt-get` to install a few programs needed to build the programs in the following steps

```
sudo apt-get install libtool autoconf automake
```

Step 1.3: Installing libplist

First, install the required dependencies for building `libplist`. In order to do this, do the following:

```
sudo apt-get install libxml2-dev python-dev
```

Then download the latest version of `libplist` from GitHub, and extract the contents of the zip file to some directory. For instance, if you are in the directory where you downloaded the `libplist` zip file, do `unzip libplist-master.zip`.

You should now have a directory called "libplist-master" in the directory to which you extracted the `libplist` zip file. `cd` into this directory from the terminal, and the run

```
./autogen.sh
```

When the `./autogen.sh` script is done running, run

```
make
```

And, finally, run

```
sudo make install
```

Step 1.4: Installing libusbmuxd

This step is similar to the previous step, except we are installing `libusbmuxd` instead of `libplist`.

First, download the latest version of `libusbmuxd` from GitHub. Again, extract the contents to a directory, and `cd` to the directory `libusbmuxd-master`. Then run the following:

```
./autogen.sh
```

When this is finished, run

```
make
```

followed by

```
sudo make install
```

Step 1.5: Installing libimobiledevice

First, install the build dependencies by doing the following:

```
sudo apt-get install libssl-dev
```

Then download the latest version of `libimobiledevice` from GitHub. Extract as in the previous two steps; you should get a directory inside the directory to which you extracted called `libimobiledevice-master`. `cd` into this directory, and, again, run

```
./autogen.sh
```

When this is finished, run

```
make
```

followed by

```
sudo make install
```

Step 1.6: Installing a better version of usbmuxd

First, uninstall the old version of `usbmuxd` by doing

```
sudo apt-get remove usbmuxd
```

Then, install the build dependencies by doing

```
sudo apt-get install libimobiledevice-dev libplist-dev libusb-dev libusb-1.0.0-dev
libtool-bin libtool
```

Then, download the latest version of `usbmuxd` from GitHub. Extract and `cd` to the `usbmuxd-master` directory. Again, run

```
./autogen.sh
```

When this is finished, run

```
make
```

followed by

```
sudo make install
```

Step 1.7: Installing ifuse

This is the last thing you will need to install!

First install, the build dependencies by doing

```
sudo apt-get install libfuse-dev
```

Download the latest version of `ifuse` from GitHub. Extract it to some directory, and `cd` into the directory `ifuse-master` , and `cd` into that directory.

This time there is an extra step in building the program. Do

```
./autogen.sh
```

as usual, but then do

```
./configure
```

as well. Then, continue on to the normal

```
make
```

and

```
sudo make install
```

Step 2: Running usbmuxd and attaching iPhone

This step is simple. Run `usbmuxd` in the terminal, and then plug in the iPhone.

Now check to see if the device was recognized correctly by doing

```
dmesg | grep ipheth
```

If nothing shows up, try disconnecting the iPhone, running `usbmuxd` again, and then plugging back in. Then check again.

Step 3: Creating a mount point for the iPhone

You can manually create a mount point for the iPhone by doing

```
sudo mkdir /media/iPhone
```

You will then likely want to change the permissions for the mount point. Do

```
sudo chmod 777 /media/iPhone
```

Step 4: Editing the ifuse configuration file

The ifuse configuration file `/etc/fuse.conf` requires editing if you want to access the iPhone without being root.

Edit the configuration file using your favorite editor, for example gedit

```
sudo gedit /etc/fuse.conf
```

In the file ensure that the following two lines are under the line that says `# Allow non-root users to specify the allow_other or allow_root mount options` :

```
op$
user_allow_other
```

Save the file and quit the editor.

Step 5: Pairing the iPhone

Run the following line in order to pair your iPhone using `idevicepair` :

```
idevicepair pair
```

Step 6: Mounting with ifuse

Run the following line to mount the device at the mount point specified earlier:

```
ifuse /media/iPhone
```

NOTE: At this point you may mount the root filesystem if you have your phone jailbroken by doing the following line instead

```
ifuse /media/iPhone/ --root
```

The iPhone should now be accessible at `/media/iPhone` through your file browser.

When you want to unmount, do the following two lines

```
fusermount -u /media/iPhone/
idevicepair unpair
```

These steps were adapted for xenial from this tutorial at dedoimedo, then further modified to suit devices with iOS 9+.

edited Nov 8 '16 at 2:22

answered Aug 15 '16 at 3:24
fakedad
631 4 16

[1461.760447] ipheth 1-2:4.2 enp0s20u2c4i2: renamed from eth0 stream13@stream13-HP-Stream-Notebook-PC-13:~$ mkdir /media/iPhone mkdir: cannot create directory '/media/iPhone': Permission denied I don't know what to do mate... – Rizio A. Aug 15 '16 at 3:55

@RizioA.Sorry, that should have been `sudo mkdir /media/iPhone` – fakedad Aug 15 '16 at 3:57

@RizioA.Same with `sudo chmod 777 /media/iPhone` – fakedad Aug 15 '16 at 3:58

4 This does not work for iOS 10: "GnuTLS error: Error in the pull function. Failed to connect to lockdownd service on the device. Try again. If it still fails try rebooting your device." – zgoda Oct 12 '16 at 9:08

1 Step#6 is not working for me, but file explorer worked, thanks a lot – Anand Rockzz Nov 25 '16 at 1:56

After having built new versions of libplist, libusbmuxd, libimobiledevice, ifuse and usbmuxd under /usr/local/lib, those versions have to be taken into use when calling the commands to connect the iPhone (instead of using the old library versions, which is done by default). So I did the following in a bash-shell (make sure to unlock (enter your code or use your fingerprint) your iPhone before each action):

```
sudo LD_LIBRARY_PATH=/usr/local/lib usbmuxd
export LD_LIBRARY_PATH=/usr/local/lib
idevicepair pair
ifuse /media/iPhone
```

(comment originated from user639768 and was posted as a question, but has since been deleted.)

edited Apr 13 at 12:23

community wiki
2 revs
Thomas Ward

It seems that with iOS 10.2 Apple has broken it again. To fix:

1. Download from GitHub latest versions of: `libplist` , `libusbmuxd` , `libimobiledevice` , `ifuse` and `usbmuxd`
2. Extract ZIP files in a folder that you like, enter in each directory and launch compilation of the corresponding library (if you don't know how to do, just follow instructions inside the readme file inside each lib and remember that the development version of `openssl`

is called `libssl-dev` when required). Please note that last command 'sudo make install' should put compiled libraries in `/usr/local/lib` .

3. Make sure that environment vars point to this new versions rather than the old ones included in official packages:

```
sudo LD_LIBRARY_PATH=/usr/local/lib usbmuxd
export LD_LIBRARY_PATH=/usr/local/lib
```

4. Connected your iOS 10.2 device to your computer:

```
idevicepair pair
```

5. Select "Trust" to the warning "Trust This Computer?" on your device:

```
idevicepair pair
```

6. Mount the iOS filesystem

```
ifuse Mountpoint_Directory/
```

Tested on Ubuntu 16.04 and iPhone 7 with iOS 10.2

edited Nov 19 at 20:08 answered Feb 24 at 22:44

bytepan
51 1 3

On my stock Ubuntu 16.04 system, the phone doesn't even show up in lsusb. Would this library set fix that so that I can get iTunes working within VirtualBox? – Greg Bell Jul 25 at 0:06

Several months have passed and I honestly don't remember perfectly, but I think that my iPhone was detected in lsusb even before applying the new libraries, so I don't know if they can fix the issue you're talking about. – bytepan Jul 26 at 8:17

protected by Community ♦ Dec 12 '16 at 4:08

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 reputation on this site (the association bonus does not count).

Would you like to answer one of these unanswered questions instead?