Booting opencore without the USB on an AsRock motherboard

I just spent *many* hours over 2 days trying to get High Sierra to boot without the installer USB using opencore on my AsRock mobo with the H97 chipset. The problem was that the UEFI just didn't want to acknowledge the existence of the EFI partition on my Hackintosh drive. I could select the correct SSD from the boot menu, but no matter what I did, the UEFI kept telling me to "insert boot media" and try again.

I finally figured it out, but the solution was a bit difficult to find. There's no single guide or forum post on the internet that led me there, I had to piece the solution together by reading dozens of Hackintosh and Arch Linux forum posts. I'm making this post here so that future people who may find themselves in this situation don't have to look as hard.

Anyway, apparently the UEFI on some AsRock boards is weird and just copying the contents of the EFI folder from the USB to the internal EFI partition may not be enough to boot your Hackintosh without the USB installer plugged in. Especially when you have multiple disks in the computer where another disk already has a working EFI partiton (like a Windows installation or whatever). In that case you have to manually create a bootloader entry that points at the right partition on the right disk. I did it by booting into Linux Mint and creating the entry via the terminal, but it can probably be done on Windows with something like EasyUEFI or whatever IF YOU ALREADY HAVE WINDOWS INSTALLED ON THE SAME MACHINE. On Linux it works like this:

1. First you'll need to have a working Hackintosh install that can boot fine WITH the USB. This is a post-install operation. You'll also need to copy the contents of the EFI partition from your USB installer to the internal EFI partition on the drive you installed MacOS on, as described in the opencore guide. The following steps just enable that to work, they're not an alternative or a replacement.

2. You'll need a live Linux distro that has the `efibootmgr`package installed. Linux Mint's live environment has it, probably others have it too. Mint is a good choice because it's very easy to create a live USB that just works. Plenty of guides out there to find out how to create a live USB and boot into it. If the only USB stick you currently have is your Hackintosh installer, make a backup of your working EFI folder before making it a live linux USB, because you may need that if this guide doesn't fix things for you.

3. Boot into the live environment.

4. Launch a terminal using `ctrl+alt+t` or by clicking on the icon at the bottom of the screen.

5. Type `sudo -s` and hit enter. You don't have to do this as it's just a little quality of life improvement. Every command you're going to run requires root privileges and this step is gonna save you from typing "sudo" every time.

6. Now you need to identify where your MacOS EFI partition resides.

- Type `lsblk` into the terminal and hit enter. It will output a neatly organized list of all every partition in your PC. Find the EFI partition that sits on the same disk as your MacOS installation. That's the one the bootloader entry needs to point at. The specific pieces of information you need from the list is the NAME of the partition, which has the format "sdXY" where X is a letter and Y is a number. The sdX part is the disk name, and the Y is the partition number. For example on my machine the EFI partition is partition 1 on disk sdc, which makes it sdc1.

- Next, type `blkid` into the terminal and hit enter. That will output more information. Find the row that begins with /dev/sdXY where X and Y are your disk name and partition number. It hopefully has a string that's something like PARTUUID=<xxxxxxxx>. You'll need the PARTUUID.

1. Now open a second terminal and repeat step 5. You don't have to do this step but I just think it's nice to have the necessary information visible on a second window.

2. Type `efibootmgr` into the new and clean terminal, and hit enter. That's gonna list all the bootloader entries in your UEFI. Have a good look, see how it looks right now so you can compare it with the result later to make sure the new entry is added.

3. Now to create the bootloader entry. Type `efibootmgr -c -L "<label>" -l "\EFI\BOOT\BOOTx64.efi" -d /dev/<sdX> -p <Y> -u root=PARTUUID=<partuuid>` and hit enter. The output should just be the previous list + your new entry.

- The argument `-c` tells the command we're creating a new entry

- `-L`means the following string `<label>` is whatever you want to call the new bootloader entry. To name it "High Sierre" for example, type in `-L "High Sierra"`

- `-l`means you're specifying the loader file. With Clover you point it at `\EFI\CLOVER\CLOVER.efi`, but applying the same logic to Opencore and typing in `\EFI\OC\OpenCore.efi` WILL NOT work. It needs to be pointed at `\EFI\BOOT\BOOTx64.efi`.

- `-d`means you're specifying which disk you want the entry to point at. Replace `<sdX>`with the **disk name** you got from the `lsblk` command. For me it was `-d /dev/sdc`DO NOT TYPE IN THE PARTITION NUMBER HERE

- `-p`means you're specifying which partition on the specified disk you want the entry to point at. Replace `<Y>` with the partition number you got from the `lsblk` command. For me it was just `1`. That's the number one.

- The final argument `-u root=PARTUUID=<partuuid>` specifies the partition's PARTUUID. Replace `<partuuid>` with the `<xxxxxxxx>` from the output of the `blkid` command. On Linux terminals, the keyboard shortcut to copy the highlighted text is `ctrl+SHIFT+c`, not just ctrl+c. The latter is of course something else entirely. Likewise, you paste into the terminal with `ctrl+SHIFT+v`.

1. Type `efibootmgr -v` and hit enter. That's gonna output details about your bootloader entries. It won't be particularly easy to read but it shouldn't be too bad. Find the entry you just created. It should look something like `Boot000x <your label> HD(something, something, <your partuuid>, something, something)/FILE(\EFI\BOOT\BOOTx64.efi)`. Note the

000x part of the `Boot000x`, and make sure all the stuff inside the <>s are correct. Don't worry about the "something" fields. If everything looks good, proceed. If you think they're incorrect, delete the entry with the command `efibootmgr -Bb 000x` and go back to #9. MAKE TRIPLE SURE YOU'RE REMOVING THE CORRECT ENTRY BEFORE HITTING ENTER. You don't want to break an existing and working bootloader if there are any.

2. Type `efibootmgr -o 000x` where the 000x is from the previous step. That's gonna move your new entry up the list. Not a 100% necessary step, you can always move the entries around from your UEFI's boot menu.

3. Reboot by typing `reboot now`and hitting enter. Or click the relevant buttons, who cares.

4. If you skipped step 11, or if that command just didn't move the entry all the way to the top, go into your UEFI's boot menu and do it manually. Reboot again when the new entry is at the very top of the boot order.

Hopefully that should do it. I hope I used all the relevant keywords so it's easier to find.