Portable Linux v2

Prologue :

This is a fairly quick yet comprehensive guide to setting up a linux distribution install to be portable. We will get GRUB to boot linux on UEFI and BIOS to achieve this.

The guide is step-by-step and does not presume much prior knowledge of Linux. It has been tested on Ubuntu Desktop `20.04` and Manjaro, but ought to work on any distro. If you find any bugs or have issues following the guide or wish to critique this material, post it as an issue in the GitHub repo.

Requirements :

1. The install media (USB stick, DVD, ...) with the desired ISO flashed to it.
2. Computer(s) with BIOS and UEFI boot options.
3. Internet connection, ideally a wired ethernet connection for minimal hassle, but most WiFi shoud just work.

Install Guide

1. Boot from the install media in **BIOS/Legacy** mode.

   How exactly you do this varies by manufacturers & models. Try to search the web for `<your_computer>` or `<your_motherboard_model> + boot options menu` . After boot-up, connect/insert the target device to your computer.

2. We will use GParted, a standard disk partitioning utility that may come prebuilt in your system.

   Open a terminal window, type the `gparted` command at the prompt and hit the ENTER key to execute it.

   If you get an error message like `command not found` or similar, then you may need to install the tool.

   Install instructions vary by distribution :

   - For Debain & Debian-based distros, like Ubuntu/Mint/PopOS/Zorin , execute `sudo apt install gparted` .
   - For an Arch based distro, like Manjaro/Endevour/Garuda/Arco, execute `sudo pacman -S gparted` .
   - For others, consult distro wikis or documentation on how to install packages or just search the web.

Otherwise, you can also use various alternatives to do the same job if you know how to use them.

3. Launch Gparted.

   Change the dropdown menu in the top right-hand corner to select your target device. Right click any existing mounted partitions on said target USB and unmount/swapoff them.

4. Click `Device` on the top bar, and click `Create partition table`. Change the dropdown to `GPT` (**note:** this will erase *all* data on the disk, ensure data is backed up to your satisfaction *first!*).

5. Right click the disk representation to create a new partition. The size should be **1MB** and the file system should be **unformatted**. Similarly create another partition of size **100MB** and file system **FAT32**. Create one last partition, which will be used as the root filesystem of your installation. The size can be all the space left on your disk, but ideally not less than 10 GB. File system should be **ext4**.

   If you leave some space remaining, you can use it for other things or extend the root partition later in GParted if needed. However, once allocated to root, shrinking the partition is not so simple.

   Apply the changes by clicking the tick button.

6. Right click the 1MB partition, select `Manage flags`, and select `bios_grub`. Similarly set `boot` and `esp` flags for the 100MB partition.

   Then, close GParted.

7. Open your distro's installation program. Proceed normally until asked which disk to install to. In the disk selection section, select the `Manual Partitioning` or `Advanced` or a similar option. Set the ext4 partition as the **root partition** (mount point = `/`).

   If the option is availible, don't forget to change the bootloader dropdown to your target drive !

   Confirm/Continue and finish the rest of the install. Then, reboot.

8. Boot from your installation media again, this time in UEFI mode, and connect target device again. Open a terminal and execute `su` to swtich to root user.

9. Run `fdisk -l`. Take note of the `/dev/sdX` label of your target drive, it will probably be something like `/dev/sdb` or `/dev/sdc`. In these next commands, `/dev/sdX` will refer to your target. Make sure you substitute the `X` for the actual letter shown for your one!

10. Execute the below commands :

Note : The `#` denotes a comment. These need not be copied & pasted to the terminal. They are simply for your refrence.

```
umount /dev/sdX3 # Make sure root is unmounted
umount /dev/sdX2 # Make sure 100MB partition is unmounted.
mount /dev/sdX3 /mnt
mkdir -p /mnt/boot/efi
mount /dev/sdX2 /mnt/boot/efi
mount --bind /dev /mnt/dev
mount --bind /dev/pts /mnt/dev/pts
mount -t proc proc /mnt/proc
mount -t sysfs sysfs /mnt/sys
mount -t tmpfs tmpfs /mnt/run
chroot /mnt
grub-install --efi-directory=/boot/efi --target=x86_64-efi --removable # Replace x86_6
update-grub
blkid | grep /dev/sdX2 # Note UUID, which is in format XXXX-XXXX.
echo "UUID=XXXX-XXXX /boot/efi vfat umask=0077 0 1" >> /etc/fstab
sync
exit
sync
poweroff
```

`poweroff` will cause an immediate shutdown. You can now boot from the target device on both UEFI and BIOS systems.

Post Install

- You may wish to install proprietary drivers to allow important functionality like graphics or networking. For example, you may want to install the Nvidia drivers or wireless drivers for macs, etc. depending on what you need.
- If your target device was a USB stick, SD card, or SSD, then it is advisable to set the system to not make unnecessary writes.

    i. Systemd Journald

    Open `/etc/systemd/journald.conf` in a text editor as root, for example `sudo nano /etc/systemd/journald.conf`. Uncomment/add `Storage=volatile`. Uncomment/add `SystemMaxUse=16M`.

    ii. Fstab mount options

    Open `/etc/fstab` in a text editor as root. Find the line for the `/` ext4 partition. Find the mount options, they may look like `defaults`. If not already present, append `,noatime` to the options for `/`.

- Make sure to do a system upgrade ! On a debian-based system, execute `sudo apt update && sudo apt upgrade`

    On an Arch-based system, execute `sudo pacman -Syu`

Epilogue :

- Thanks to Daniel Masey for answering my original AskUbuntu question. Installation steps 2-10 are abridged versions of his answer.
- See also :
    i. Archwiki page
    ii. c-magyar's guide
    iii. Mac wifi hw docs for Debian-based, Ubuntu and Arch-based
    iv. Nvidia graphics docs for Debian-based and Arch-based