

APPENDIX I. How to install a Linux distribution on a USB hard drive.

1. Prepare a Linux Live Bootable USB hard drive

The first step is to download a Linux distribution. We will use **LUbuntu¹ 20.04² LTS Desktop 64-bit**. After downloading the corresponding ISO image from <http://lubuntu.me/downloads/>, we will connect the USB hard drive to the laptop, prepare it to boot the downloaded ISO, and finish the installation **on the same drive**³. Modern live Linux ISO images support booting both in **[UEFI]** and legacy **[BIOS]** modes. Whichever mode we boot the ISO installer in, that is the mode Linux will be installed in. Typically, one would choose the same mode as the native operating system installed on the laptop⁴, as this would allow a dual-boot system. In our case, this is not needed, hence if your boot device menu (accessible by pressing ESC, F2, F9, or F12 key as corresponding) allows booting from USB drives in both modes⁵, we recommend following the legacy **[BIOS]** approach, which is simpler and less prone to fatal errors. Note also that a **[UEFI]** boot installation requires that you **disable SecureBoot** at the BIOS setup. Be aware also that a new entry for the newly installed system may appear in the UEFI boot device menu even when the USB hard drive is not present. In a nutshell, we must create a new partition (2 GiB) with the live Linux image to boot and make it bootable. Depending on the native operating system installed on your laptop, take the following steps:

a. **Windows**

- i. Open a command prompt and run **DiskPart** utility (as administrator) to partition the external USB hard drive:
 1. Display the list of available disks:

```
list disk
```
 2. Select the USB hard drive and shift the focus to it:

```
select disk X #ENSURE THAT THIS DISK IS THE EXTERNAL USB HARD DRIVE!!!!
```
 3. Remove all the partition formatting from the disk with focus.

```
clean
```
 4. Create a new partition table:
 - a. **[UEFI]** Create a new empty GPT partition table:

```
convert GPT
```
 - b. **[BIOS]** Create a new empty MBR partition table:

```
convert MBR
```
 5. Create a 2 GiB **FAT32** partition in the USB hard drive. Leave the rest of the drive unpartitioned:
 - a. create partition primary size=2048
 - b. Check that the partition has been created:

```
list partition
```
 - c. Format the new partition to FAT32:

```
format quick fs=fat32
```
 - d. assign letter="X"
 - e. exit
- ii. Download the **Universal USB Installer** binary from <https://www.pendrivelinux.com/universal-usb-installer-easy-as-1-2-3/>⁶
- iii. Choose a Linux distribution (i.e. Lubuntu), browse for the ISO image of LUbuntu downloaded previously, select the letter assigned to the partition created before in step ii, and click 'Create' to create a live bootable LUbuntu distribution in the USB hard drive (enabling persistence is not necessary).
- iv. Now, the external USB drive is ready and it should boot a live LUbuntu OS.

¹ This distribution has lower resource requirements than other Linux distributions, which makes it a good option for virtualized environments.

² We have tested this version, but you might use a more recent one, especially if your laptop is new and you want to ensure that the corresponding drivers are supported. All the steps described in this document should work, but double check everything, just in case.

³ We will use the same disk as source and destination media and therefore we need to make some additional steps. Alternatively, you could save some time (and avoid some potential errors) if you use a USB pen drive to create the live bootable LUbuntu.

⁴ In Windows, you can check the mode by opening a command line prompt, executing **MSINFO32** and looking at *System summary* → *BIOS mode* (https://blogs.technet.microsoft.com/home_is_where_i_lay_my_head/2012/10/02/how-to-check-in-windows-if-you-are-using-uefi/). In Linux, you can check if the directory `/sys/firmware/efi` exists. (<https://askubuntu.com/questions/162564/how-can-i-tell-if-my-system-was-booted-as-efi-uefi-or-bios#162896>)

⁵ You might need to set "Enable Legacy Option ROMs" in the BIOS setup.

⁶ Other tools such as 'unetbootin' (<https://unetbootin.github.io/>) or 'Linux Live USB creator' (<http://www.linuxliveusb.com/en/download>) might also work depending on your system.

b. **Linux**

- i. Run `sudo fdisk -l` or `lsblk` to identify the USB hard drive. Let's use `/dev/sdX`, from now on, to refer to the USB drive (**THIS IS A VERY IMPORTANT STEP: A WRONG IDENTIFICATION MAY DESTROY YOUR DATA LATER**).
- ii. Open a command prompt and run **fdisk** utility to partition the USB hard drive:
 1. `sudo fdisk /dev/sdX #CHECK NUMBERING`
 - a. **[UEFI]** Create a new empty GPT partition table: `'g'`
 - b. **[BIOS]** Create a new empty MBR partition table: `'o'`
 2. **[UEFI]** For compatibility reasons, we reserve the first 16 MiB by creating an initial empty partition:
 - a. Print the current partitions of the drive: `'p'`
 - b. Create a new partition: `'n'`
 - i. Partition number: The first available number [default]
 - ii. First sector: The first available sector [default]
 - iii. Last sector: The size of the partition `'+16M'`
 - c. Check that partition has been created: `'p'`
 - d. Change the partition type: `'t'`
 - i. `'10'` ... to Microsoft Reserved
 3. Create a 2 GiB **FAT32** partition in the USB hard drive. Leave the rest of the drive unpartitioned:
 - a. Print the current partitions of the drive: `'p'`
 - b. Create a new partition: `'n'`
 - [BIOS]** Partition type: Select primary partition [default]
 - i. Partition number: The first available number [default]
 - ii. First sector: The first available sector [default]
 - iii. Last sector: The size of the partition `'+2G'`
 - c. Check that partition has been created: `'p'`
 - d. Change the partition type: `'t'`
 - i. Partition number: X **#CHECK NUMBER**
 - ii. **[BIOS]** `'b'` ... to a FAT32
 - iii. **[UEFI]** `'11'` ... to Microsoft Basic Data
 - e. **[BIOS]** Make the partition bootable: `'a'`
 - f. Write the changes and exit: `'w'`
 4. Format the new 2 GiB partition, so-called `/dev/sdXX`, to FAT32:
`sudo mkfs.vfat -F 32 /dev/sdXX`
- iii. Mount the 2 GiB partition somewhere, for example `/media/usb` (this is required by the program installed in the following steps):
`sudo mount -t vfat /dev/sdXX /media/usb`
- iv. Download the **unetbootin** binary from <https://unetbootin.github.io>
- v. Give execution permissions to the binary:
`chmod u+x unetbootin`
- vi. Run **unetbootin** from the command line passing as a parameter the FAT32 partition created before:
`sudo unetbootin installtype=USB targetdrive=/dev/sdXX`
- vii. Create a live bootable LUbuntu distribution in the USB hard drive:
 1. Select the 'Diskimage' option.
 2. Browse for the LUbuntu ISO image downloaded previously.
 3. Ensure that the drive is the right one.
 4. Install it by pressing the 'OK' button.
- viii. Unmount the FAT32 partition:
`sudo umount /media/usb`
- ix. Now, the external USB drive is ready and it should boot a live LUbuntu OS.

2. Install Linux on the USB hard drive

- i. Restart your system, enter the boot device menu (pressing ESC, F2, F9, or F12 key as corresponding), and boot from the USB hard drive.
- ii. Boot the live LUbuntu **without persistence**, which will simplify the partitioning of the USB drive. Disabling persistence is not necessary if you have created the live bootable LUbuntu in an independent USB pen drive.
 - a. **[BIOS]** Press <TAB>⁷ to edit boot options and add the *'nopersistent'* option.
 - b. **[UEFI]** On the GRUB menu, edit the boot options of the 'Start LUbuntu' entry by typing 'e'. Add the *'nopersistent'* option (before '---') and press Ctrl+x or F10.
- iii. Open a command prompt and run **fdisk** utility to partition the USB hard drive (`sudo fdisk /dev/sdX`):
 - a. Create a 15 GiB primary partition to store the LUbuntu system (`/dev/sdXROOT`)
 1. Print the current partitions of the drive: 'p'
 2. Create a new partition: 'n'
 - i. **[BIOS]** Partition type: Select primary partition [default]
 - ii. Partition number: the first available number [default]
 - iii. First sector: The first available sector [default]
 - iv. Last sector: The size of the partition '+15G'
 - b. **[UEFI]** Create a new 200 MiB EFI partition (`/dev/sdXEFI`)
 1. Create a new partition: 'n'
 - i. Partition number: the first available number [default]
 - ii. First sector: The first available sector [default]
 - iii. Last sector: The size of the partition '+200M'
 2. Change partition type: 't'
 - i. Partition number: X (`/dev/sdXEFI`) **#CHECK NUMBER**
 - ii. Hex Code: '1' ... to an EFI System
 - c. Create another primary partition with the remaining space to store data (`/dev/sdXDATA`)
 1. Create a new partition: 'n'
 - i. **[BIOS]** Partition type: Select primary partition [default]
 - ii. Partition number: the first available number [default]
 - iii. First sector: The first available sector [default]
 - iv. Last sector: The size of the partition [default]
 - d. Write the changes and exit: 'w'
- iv. Format the new partitions:
 - a. `sudo mkfs.ext4 /dev/sdXROOT`
 - b. **[UEFI]** `sudo mkfs.vfat -F 32 /dev/sdXEFI`
 - c. `sudo mkfs.ext4 /dev/sdXDATA`
- v. Run the installer by clicking the icon on the desktop and follow the installation wizard steps.
- vi. On the 'Partitions' step:
 - a. **Select the USB hard drive `/dev/sdX` as storage device** and 'Manual partitioning' and move to the next screen.
 - b. Check that any partition on your current disk is not used. It usually is `/dev/sda`, but it may be `/dev/nvme0n1` if you are using a NVMe port.
 - c. **[BIOS]** Install boot loader on: 'USB hard drive `/dev/sdX`'
 - d. Set the mount points of the partitions created before in the USB hard drive `/dev/sdX`. Set "Content: Keep" as the partitions are already formatted.
 1. 15 GiB partition to store the LUbuntu system (`/dev/sdXROOT`)
 - mount point: /
 - **[BIOS]** flags: boot
 2. **[UEFI]** 200 MiB partition to store the EFI partition (`/dev/sdXEFI`)
 - mount point: `/boot/efi`
 - flags: boot
 3. Partition with the remaining space to store data (`/dev/sdXDATA`)
 - mount point: `/opt`
- vii. Follow the remaining installation wizard steps.

⁷ This assumes that the live bootable LUbuntu has been created with the Universal USB Installer.

- a. Each teammate should assign a distinct hostname for its system.
 - b. We recommend creating a common user that all the team members have to use for the software installation and experiments.
- viii. **[UEFI]** To ensure that only the new EFI partition is detected at boot time in the USB drive, delete the live 2 GiB partition⁸.
 - a. `sudo fdisk /dev/sdX #CHECK NUMBERING`
 - b. Delete partition: `'d'`
 - c. Partition number: `x` (your 2 GiB partition number) `#CHECK NUMBER`
 - d. Write the changes and exit: `'w'`
- ix. Restart your laptop, boot the newly installed LUbuntu system.
- x. Change the settings of the 'Power Management' at the *Preferences*→*LXQt settings* menu so that the system is not suspended when inactive for some time.

⁸ You can ignore the warning about the partition being in use. Changes will be visible after a reboot.