Linux Installation:

(Boot process, partitioning, and dual boot)

hitbullseye

Table of Contents

Boot Process

- When we push the system's power button, several processes are active in the background. To comprehend how any operating system functions, it is crucial to grasp the booting process. Understanding how the kernel boots are essential for fixing the booting fault. Let's start from the beginning as it is a really intriguing subject to study.

Boot Process

Below is a brief explanation of the boot method:

The Linux Booting Process - 6 Steps Described in Detail

Press the power button on your system, and after a few moments you see the Linux login prompt. Have you ever wondered what happens behind the scenes from the time you press the power button until the Linux login prompt appears?

Boot Process

The following are the 6 high-level stages of a typical Linux boot process.

1. BIOS

- BIOS stands for Basic Input/Output System
- Performs some system integrity checks
- Searches, loads, and executes the boot loader program.
- It looks for boot loader in floppy, cd-rom, or hard drive. You can press a key (typically F12 of F2, but it depends on your system) during the BIOS startup to change the boot sequence.
- Once the boot loader program is detected and loaded into the memory, BIOS gives the control to it.
- So, in simple terms BIOS loads and executes the MBR boot loader.

2. MBR

- MBR stands for Master Boot Record.
- It is located in the 1st sector of the bootable disk. Typically /dev/hda, or /dev/sda
- MBR is less than 512 bytes in size. This has three components 1) primary boot loader info in 1st 446 bytes 2) partition table info in next 64 bytes 3) mbr validation check in last 2 bytes.
- It contains information about GRUB (or LILO in old systems).
- So, in simple terms MBR loads and executes the GRUB boot loader.

3. GRUB

- GRUB stands for Grand Unified Bootloader.
- If you have multiple kernel images installed on your system, you can choose which one to be executed.
- GRUB displays a splash screen, waits for few seconds, if you don’t enter anything, it loads the default kernel image as specified in the grub configuration file.
- GRUB has knowledge of the filesystem (the older Linux loader LILO didn’t understand filesystem).
- Grub configuration file is /boot/grub/grub.conf (/etc/grub.conf is a link to this). The following is sample grub.conf of CentOS

3. GRUB

```
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.18-194.el5PAE)
        root (hd0,0)
        kernel /boot/vmlinuz-2.6.18-194.el5PAE ro
root=LABEL=/
        initrd /boot/initrd-2.6.18-194.el5PAE.img
```

3. GRUB

- As you notice from the above info, it contains kernel and initrd image.
- So, in simple terms GRUB just loads and executes Kernel and initrd images.

4. Kernel

- Mounts the root file system as specified in the “root=” in grub.conf
- Kernel executes the /sbin/init program
- Since init was the 1st program to be executed by Linux Kernel, it has the process id (PID) of 1. Do a ‘ps -ef | grep init’ and check the pid.
- initrd stands for Initial RAM Disk.
- initrd is used by kernel as a temporary root file system until kernel is booted and the real root file system is mounted. It also contains necessary drivers compiled inside, which helps it to access the hard drive partitions, and other hardware.

5. Init

- Looks at the /etc/inittab file to decide the Linux run level.
- Following are the available run levels

  0 – halt

  1 – Single user mode

  2 – Multiuser, without NFS

  3 – Full multiuser mode

  4 – unused

  5 – X11

  6 – reboot

5. Init

- Init identifies the default initlevel from /etc/inittab and uses that to load all appropriate program.
- Execute ‘grep initdefault /etc/inittab’ on your system to identify the default run level
- If you want to get into trouble, you can set the default run level to 0 or 6. Since you know what 0 and 6 means, probably you might not do that.
- Typically you would set the default run level to either 3 or 5.

6. Run level programs

- When the Linux system is booting up, you might see various services getting started. For example, it might say “starting send mail …. OK”. Those are the run level programs, executed from the run level directory as defined by your run level.
- Depending on your default init level setting, the system will execute the programs from one of the following directories.

  Run level 0 – /etc/rc.d/rc0.d/

  Run level 1 – /etc/rc.d/rc1.d/

  Run level 2 – /etc/rc.d/rc2.d/

  Run level 3 – /etc/rc.d/rc3.d/

  Run level 4 – /etc/rc.d/rc4.d/

  Run level 5 – /etc/rc.d/rc5.d/

  Run level 6 – /etc/rc.d/rc6.d/

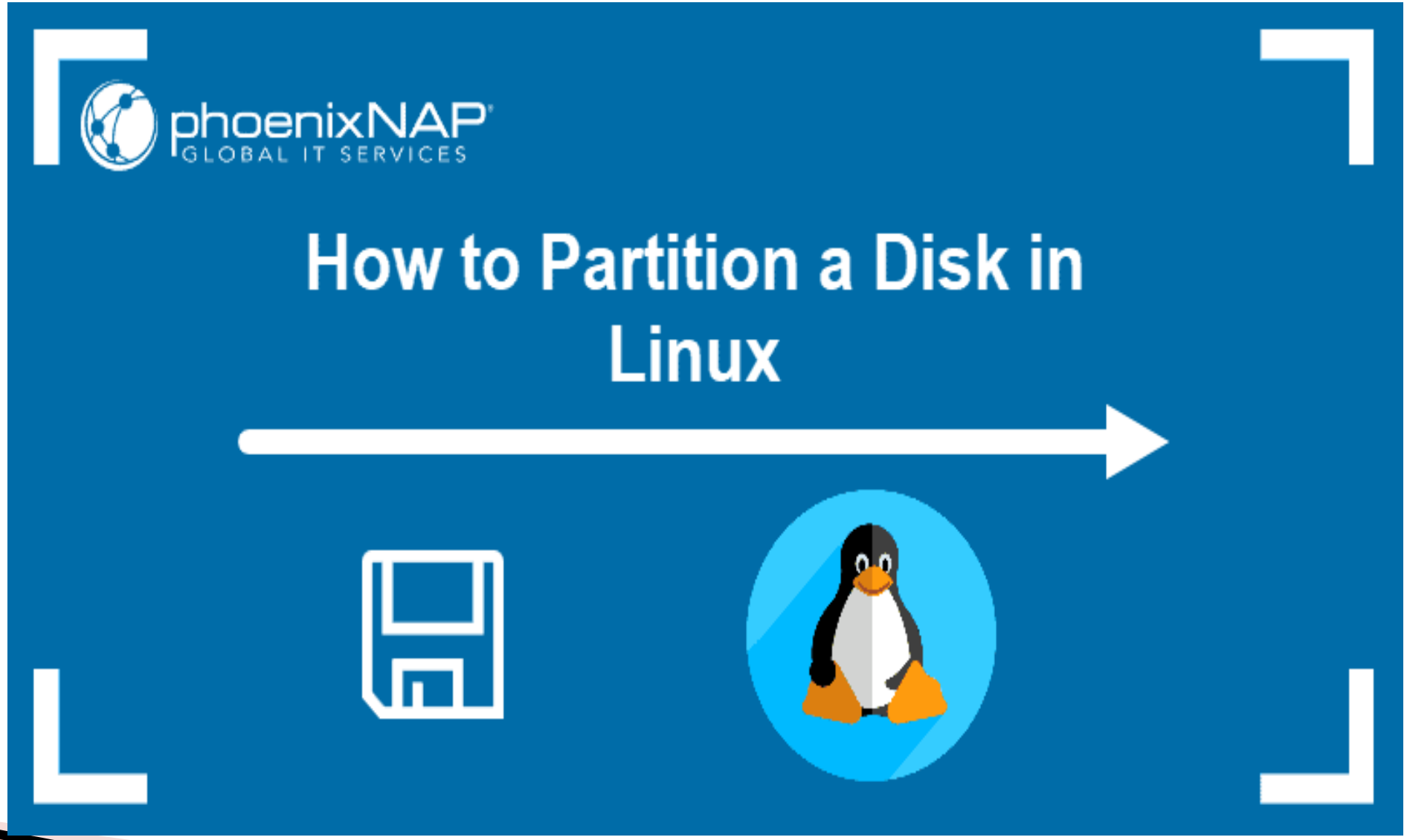
6. Run level programs

- Please note that there are also symbolic links available for this directory under /etc directly. So, /etc/rc0.d is linked to /etc/rc.d/rc0.d.
- Under the /etc/rc.d/rc*.d/ directories, you would see programs that start with S and K.
- Programs starting with S are used during startup. S for a startup.
- Programs that start with K are used during shutdown. K for the kill.
- There are numbers right next to S and K in the program names. Those are the sequence number in which the programs should be started or killed.
- For example, S12syslog is to start the Syslog daemon, which has the sequence number 12. S80sendmail is to start the Send mail daemon, which has the sequence number of 80. So, syslog program will be started before send mail.

Partitioning

- By setting up disk partitions, you may divide your hard drive into a number of independent parts. Users must setup storage devices (USB and hard drives) before utilizing them with Linux. Partitioning is also useful when installing multiple operating systems on a single workstation. In this step-by-step tutorial, you will learn how to create a partition using the Linux parted or fdisk command.

Partitioning



Prerequisites

- A system running Linux
- A user account with sudo or root privileges
- Access to a terminal window/command line (Activities > Search > Terminal)

Option 1: Partition a Disk Using parted Command

Prerequisites

- Follow the steps below to partition a disk in Linux by using the parted command.

Step 1: List Partitions

- Before making a partition, list available storage devices and partitions. This action helps identify the storage device you want to partition.

Run the following command with sudo to list storage devices and partitions:

- **Sudo parted -l**

The terminal prints out available storage devices with information about:

- **Model – Model of the storage device.**
- **Disk – Name and size of the disk.**
- **Sector size – Logical and physical size of the memory. Not to be confused with available disk space.**
- **Partition Table – Partition table type (msdos, gpt, aix, amiga, bsd, dvh, mac, pc98, sun, and loop).**
- **Disk Flags – Partitions with information on size, type, file system, and flags**

Partitions types can be

- Primary – Holds the operating system files. Only four primary partitions can be created.
- Extended – Special type of partition in which more than the four primary partitions can be created.
- Logical – Partition that has been created inside of an extended partition.

In our example, there are two storage devices (/dev/sda and /dev/sdb):

```
nevena@nevena-VirtualBox:~$ sudo parted -l
[sudo] password for nevena:
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sda: 33,3GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type      File system  Flags
 1      1049kB  538MB   537MB   primary   fat32        boot
 2      539MB   33,3GB  32,8GB  extended
 5      539MB   33,3GB  32,8GB  logical   ext4

Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdb: 10,6GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name     Flags
 1      17,4kB  1396MB  1396MB               primary
```

- **The operating system is located on the first storage drive (dev/sda or dev/vda), as appropriate. A partition on this drive can prevent your system from booting. Only secondary drives (dev/sdb, dev/sdc, dev/vdb, or dev/vdc) can be partitioned.**

Step 2: Activate the storage disk

- **Run the following command to access the storage device that you want to partition:**
- **sudo parted /dev/sdb**

```
nevena@nevena-VirtualBox:~$ sudo parted /dev/sdb
GNU Parted 3.3
Using /dev/sdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

- **Always specify the storage device. If you don’t specify a disk name, the disk is randomly selected. To change the disk to dev/sdb run:**
- **select /dev/sdb**
- **The dev/sdb disk is open:**

```
nevena@nevena-VirtualBox:~$ sudo parted
GNU Parted 3.3
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) select /dev/sdb
Using /dev/sdb
(parted)
```

Step 3: Make a Partition Table

- Create a partition table before partitioning the disk. A partition table is located at the start of a hard drive and it stores data about the size and location of each partition.
- Partition table types are: aix, amiga, bsd, dvh, gpt, mac, ms-dos, pc98, sun, and loop.
- The create a partition table, enter the following:
- mklabel [partition_table_type]
- For example, to create a gpt partition table, run the following command:
  - mklabel gpt

- **Type Yes to execute:**

```
(parted) mklabel gpt
Warning: The existing disk label on /dev/sdb will be destroyed and all data on this
disk will be lost. Do you want to continue?
Yes/No?
```

- **Note: The two most commonly used partition table types are gpt and msdos. The latter supports up to sixteen partitions and formats up to 16TB of space while gpt formats up to 9.4ZB and supports up to 128 partitions**

- **Type Yes to execute:**

```
(parted) mklabel gpt
Warning: The existing disk label on /dev/sdb will be destroyed and all data on this
disk will be lost. Do you want to continue?
Yes/No?
```

- **Note: The two most commonly used partition table types are gpt and msdos. The latter supports up to sixteen partitions and formats up to 16TB of space while gpt formats up to 9.4ZB and supports up to 128 partitions**

Step 4: Check Table

- **Run the print command to review the partition table. The output displays information about the storage device:**

```
(parted) print
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdb: 10,6GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start  End  Size  File system  Name  Flags

(parted)
```

- **Note: Run help mkpart command to get additional help on how to create a new partition.**

Step 5: Create Partition

- **Let’s make a new 1854MB-partition using the ext4 file system. The assigned disk start shall be 1MB and the disk end is at 1855MB.**
- **To create a new partition, enter the following:**
- **mkpart primary ext4 1MB 1855MB**

- **After that, run the print command to review information on the newly created partition. The information is displayed under the *Disk Flags* section:**

```
(parted) mkpart primary ext4 1MB 1855MB
(parted) PRINT
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdb: 10,6GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name     Flags
 1      1049kB  1855MB  1854MB  ext4         primary
```

- **In a gpt partition table, the partition type is the mandatory partition name. In our example, the primary is the name of the partition, not the partition type.**

- **To save your actions and quit, enter the quit command. Changes are saved automatically with this command.**

```
(parted) quit
Information: You may need to update /etc/fstab.

nevena@nevena-VirtualBox:~$
```

- **Note: The *"You may need to update /etc/fstab file"* message signals that the partition can be mounted automatically at boot time.**

Option 2: Partition a Disk Using fdisk Command

- Follow the steps below to partition a disk in Linux by using the fdisk command.
- Step 1: List Existing Partitions
- Run the following command to list all existing partitions:
  - sudo fdisk -l

- **The output contains information about storage disks and partitions:**

```
Disk /dev/sda: 31,3 GiB, 33312931840 bytes, 65064320 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x558d572e

Device     Boot   Start      End  Sectors  Size Id Type
/dev/sda1  *       2048  1050623  1048576  512M  b W95 FAT32
/dev/sda2       1052670 65062911 64010242 30,5G  5 Extended
/dev/sda5       1052672 65062911 64010240 30,5G 83 Linux


Disk /dev/sdb: 9,91 GiB, 10621960192 bytes, 20746016 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 96200583-C460-44DD-A69A-7B376C533B5D

Device     Start     End Sectors  Size Type
/dev/sdb1   2048 3622911 3620864  1,7G Linux filesystem
```

Step 2: Select Storage Disk

- **Select the storage disk you want to create partitions on by running the following command:**
- **sudo fdisk /dev/sdb**
- **The /dev/sdbstorage disk is open:**

```
nevena@nevena-VirtualBox:~$ sudo fdisk /dev/sdb

Welcome to fdisk (util-linux 2.34).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.
```

Step 3: Create a New Partition

1. Run the n command to create a new partition.
2. Select the partition number by typing the default number (2).
3. After that, you are asked for the starting and ending sector of your hard drive. It is best to type the default number in this section (3622912).
4. The last prompt is related to the size of the partition. You can choose to have several sectors or to set the size in megabytes or gigabytes. Type +2GB to set the size of the partition to 2GB.

A message appears confirming that the partition is created

```
Disk /dev/sda: 31,3 GiB, 33312931840 bytes, 65064320 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x558d572e

Device     Boot    Start       End   Sectors  Size Id Type
/dev/sda1  *        2048   1050623   1048576  512M  b W95 FAT32
/dev/sda2        1052670 65062911 64010242 30,5G  5 Extended
/dev/sda5        1052672 65062911 64010240 30,5G 83 Linux


Disk /dev/sdb: 9,91 GiB, 10621960192 bytes, 20746016 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 96200583-C460-44DD-A69A-7B376C533B5D

Device        Start      End Sectors  Size Type
/dev/sdb1      2048  3622911 3620864  1,7G Linux filesystem
/dev/sdb2   3622912  7528447 3905536  1,9G Linux filesystem
```

Format the Partition

- Once a partition has been created with the parted of fdisk command, format it before using it.
- Format the partition by running the following command:
- sudo mkfs -t ext4 /dev/sdb1

```
nevena@nevena-VirtualBox:~$ sudo mkfs -t ext4 /dev/sdb1
mke2fs 1.45.5 (07-Jan-2020)
Creating filesystem with 452608 4k blocks and 113344 inodes
Filesystem UUID: d2d0d85b-aac8-4c67-a92a-901779c9ce2d
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

Note: Check out our guide and learn how to format and mount disk partitions in Linux using the ext4, FAT32, or NTFS file system!

Mount the Partition

To begin interacting with the disk, create a mount point and mount the partition to it.

1. Create a mount point by running the following command:

```
sudo mkdir -p /mt/sdb1
```

2. After that, mount the partition by entering:

```
sudo mount -t auto /dev/sbd1 /mt/sdb1
```

The terminal does not print out an output if the commands are executed successfully.

Mount the Partition

3. Verify if partition is mounted by using the df hT command:

```
nevena@nevena-VirtualBox:~$ sudo mkdir -p /mt/sdb1
nevena@nevena-VirtualBox:~$ sudo mount -t auto /dev/sdb1 /mt/sdb1
nevena@nevena-VirtualBox:~$ df -hT
Filesystem     Type      Size  Used Avail Use% Mounted on
udev           devtmpfs  1,2G     0  1,2G   0% /dev
tmpfs          tmpfs     249M  1,4M  248M   1% /run
/dev/sda5      ext4       30G  7,1G   22G  25% /
tmpfs          tmpfs     1,3G     0  1,3G   0% /dev/shm
tmpfs          tmpfs     5,0M  4,0K  5,0M   1% /run/lock
tmpfs          tmpfs     1,3G     0  1,3G   0% /sys/fs/cgroup
/dev/loop0     squashfs   55M   55M     0 100% /snap/core18/1880
/dev/loop1     squashfs   56M   56M     0 100% /snap/core18/1885
/dev/loop2     squashfs   63M   63M     0 100% /snap/gtk-common-themes/1506
/dev/loop3     squashfs  256M  256M     0 100% /snap/gnome-3-34-1804/36
/dev/loop4     squashfs   30M   30M     0 100% /snap/snapd/8790
/dev/loop5     squashfs   50M   50M     0 100% /snap/snap-store/467
/dev/loop6     squashfs   31M   31M     0 100% /snap/snapd/9279
/dev/sda1      vfat      511M  4,0K  511M   1% /boot/efi
tmpfs          tmpfs     249M   20K  249M   1% /run/user/1000
/dev/sdb1      ext4      1,7G  5,2M  1,6G   1% /mt/sdb1
nevena@nevena-VirtualBox:~$
```

Dual boot:

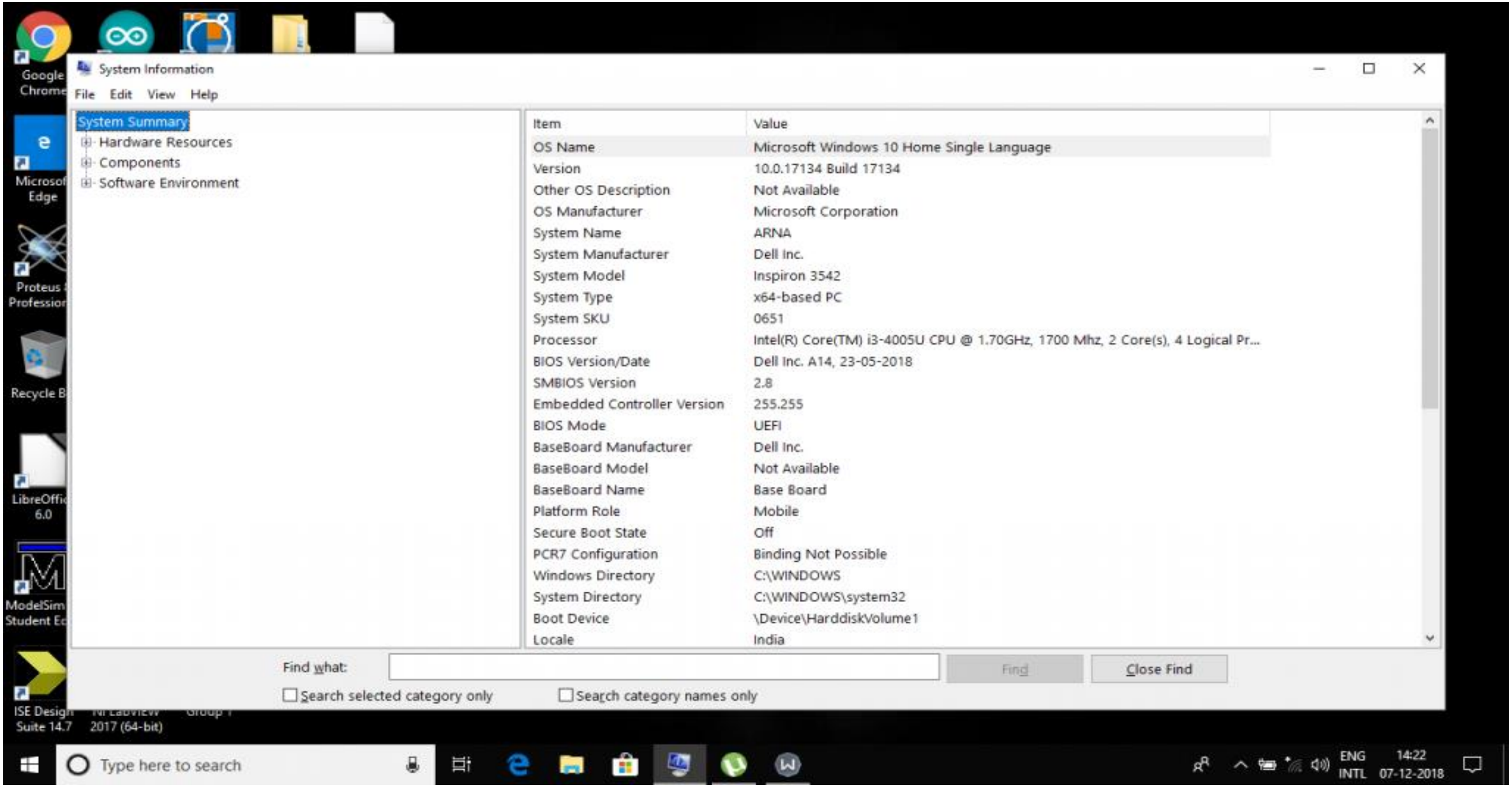
Here’s the basic process you’ll need to follow:

In this lecture, we will see how to install a Linux-based OS alongside a Windows OS already installed on a System. Before we begin here are some of the Advantages of a Linux OS over a Windows OS and why you should consider using a Linux System. (Purely based on my experience)

- Linux Systems are generally light on system resources and hence run faster even on old hardware setups.
- Linux has native support for the gcc compiler and python interpreter.
- Linux also has native support for Git which makes it easier for people to synchronize their work with remote repositories on GitHub.
- Linux is Open Source and hence tinkering around with its source code can give us a better understanding of the Structure of an OS.

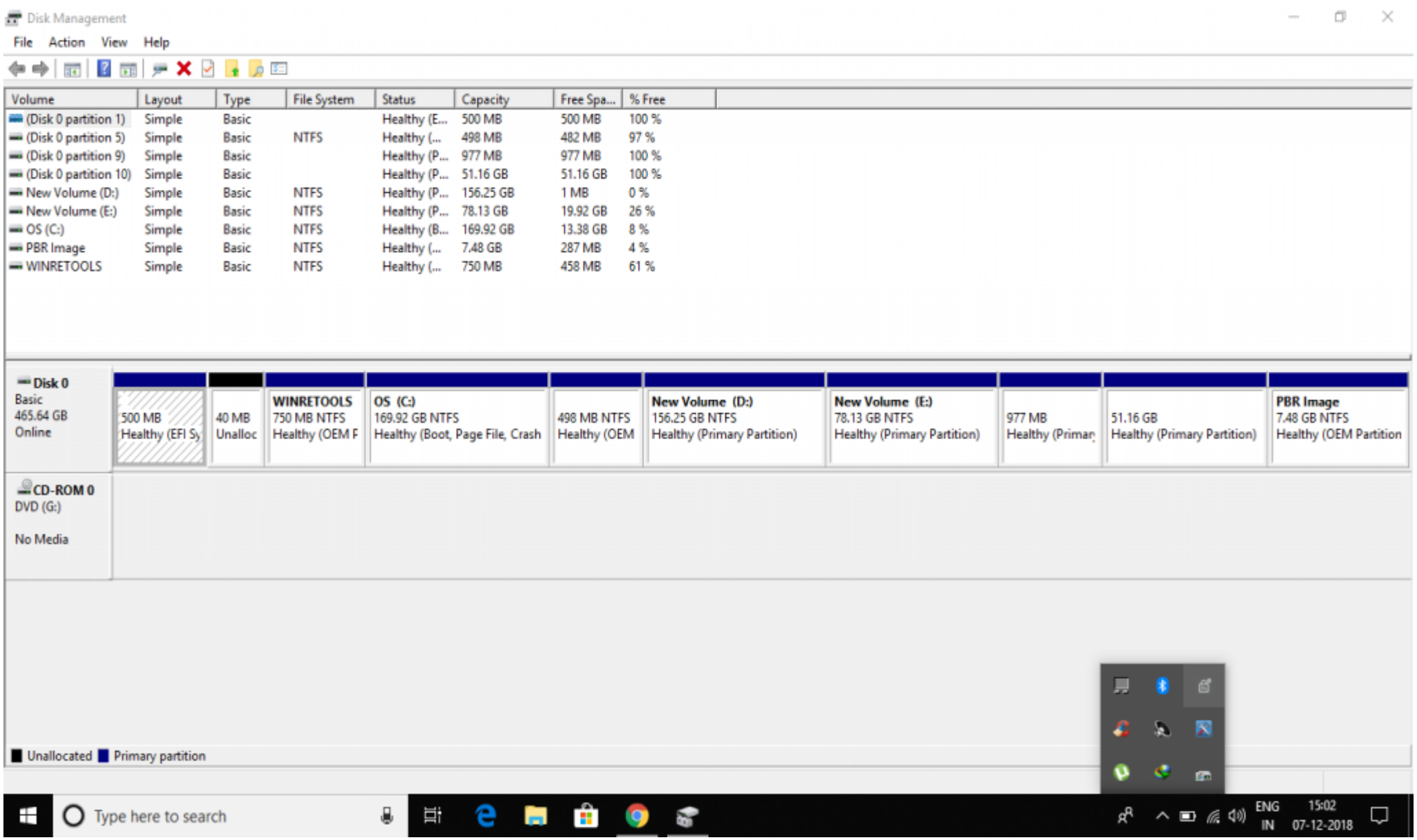
Prerequisites, before we begin the process:

- It is recommended that you have 100 – 150 Gb of free Hard Disk space, although anything above 40 Gb would do the job. (This step is important as it is the amount of space you will be dedicated to your Linux OS, and once Linux is installed you will not be able to access this space through Windows)
- A USB flash drive has a minimum space of 8Gb.
- (Optional, Only for newer machines) Make sure whether Windows is installed in UEFI mode or Legacy BIOS Mode.

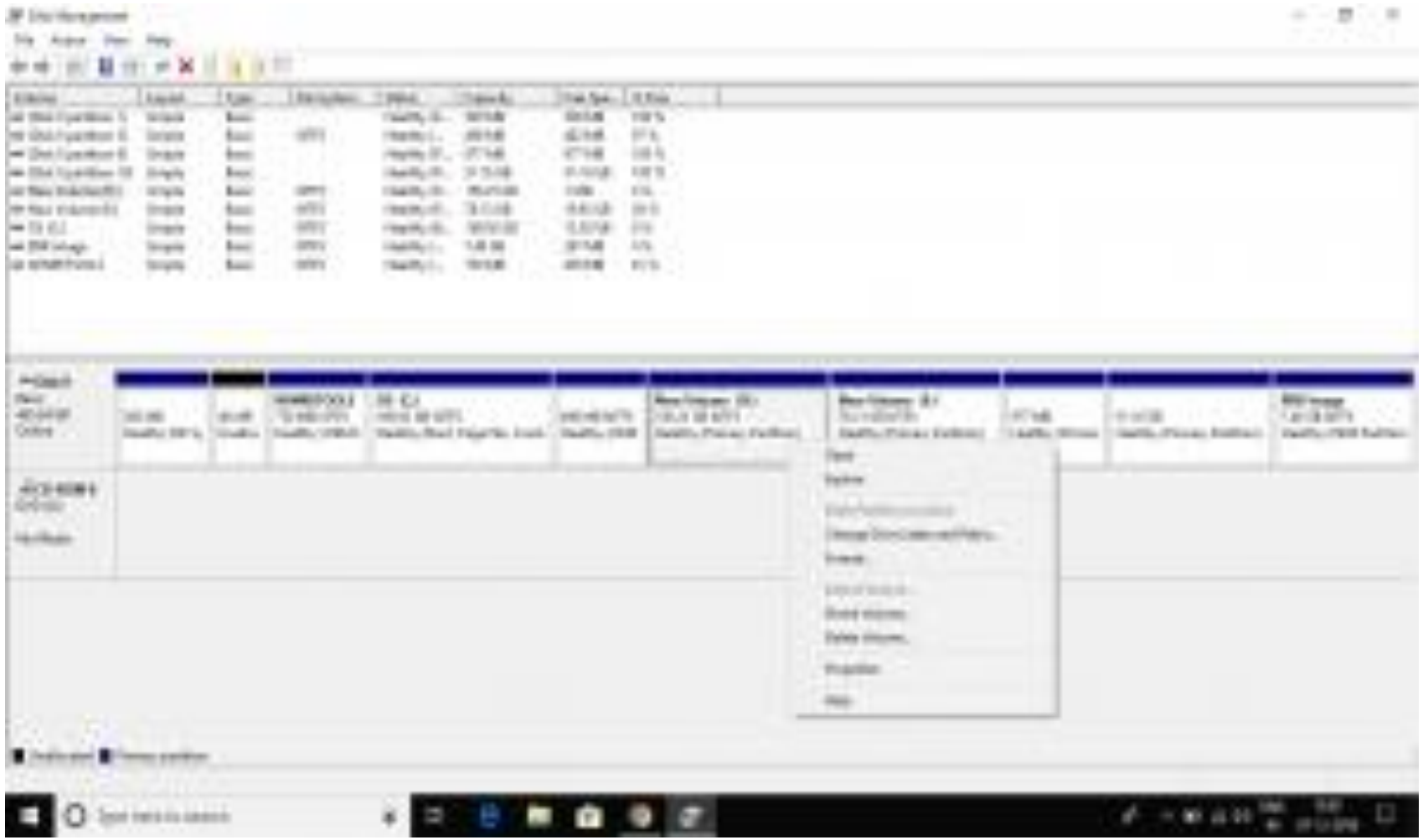


Making Sure whether Windows is installed in legacy BIOS mode or UEFI mode

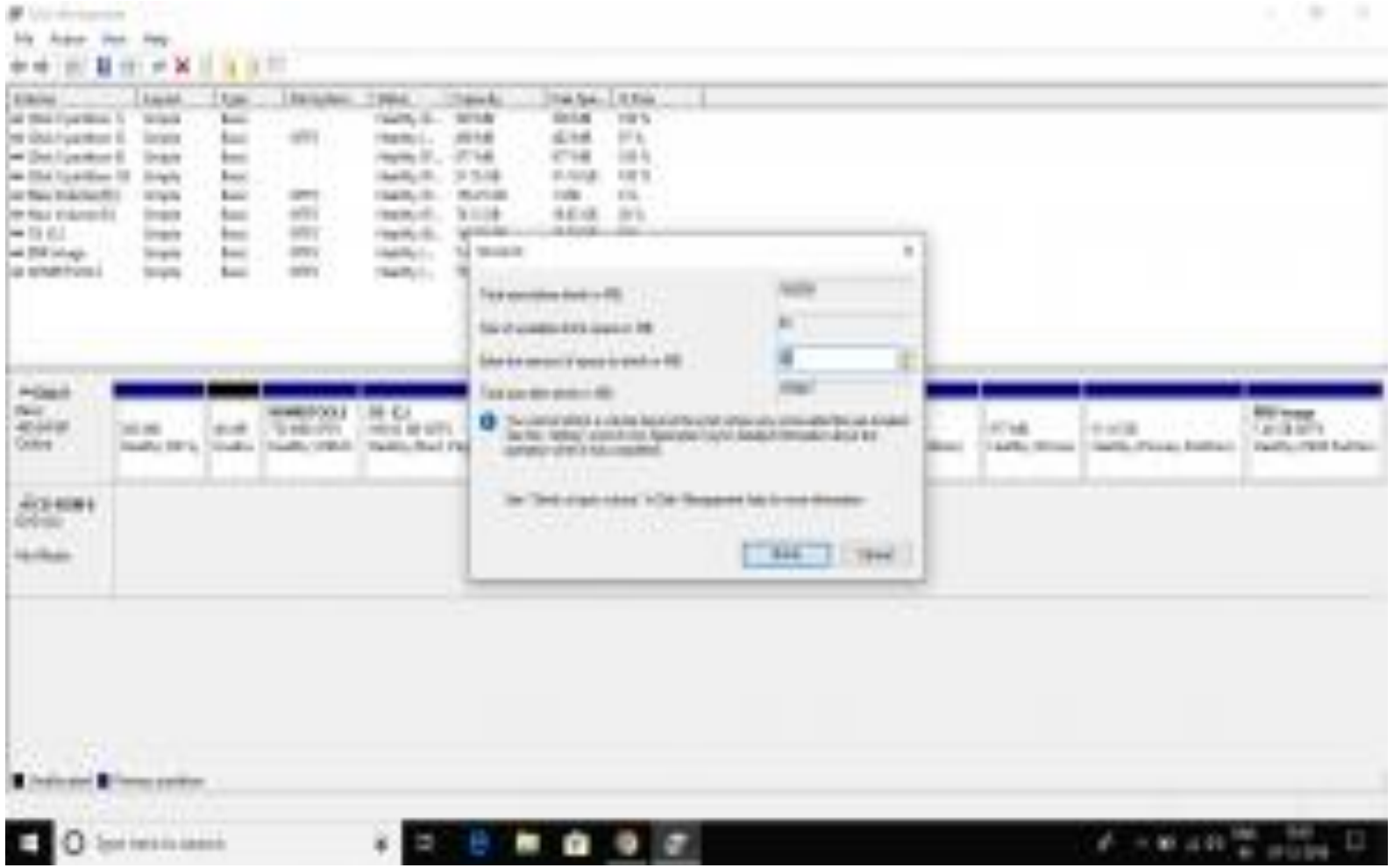
STEP 1: Unallocating the Required amount of Hard Disk Space. This step is required to free up some space from your NTFS formatted Hard Disk. This unallocated space will contain the Linux System Partition and hence it is necessary for you to decide carefully how much space you will be devoting to linux. To do this go to the Windows Disk Management Service. You can go to ‘Run’ enter ‘diskmgmt.msc’ and and run.



Windows Disk Management Service



Shrinking an existing Volume to create unallocated space

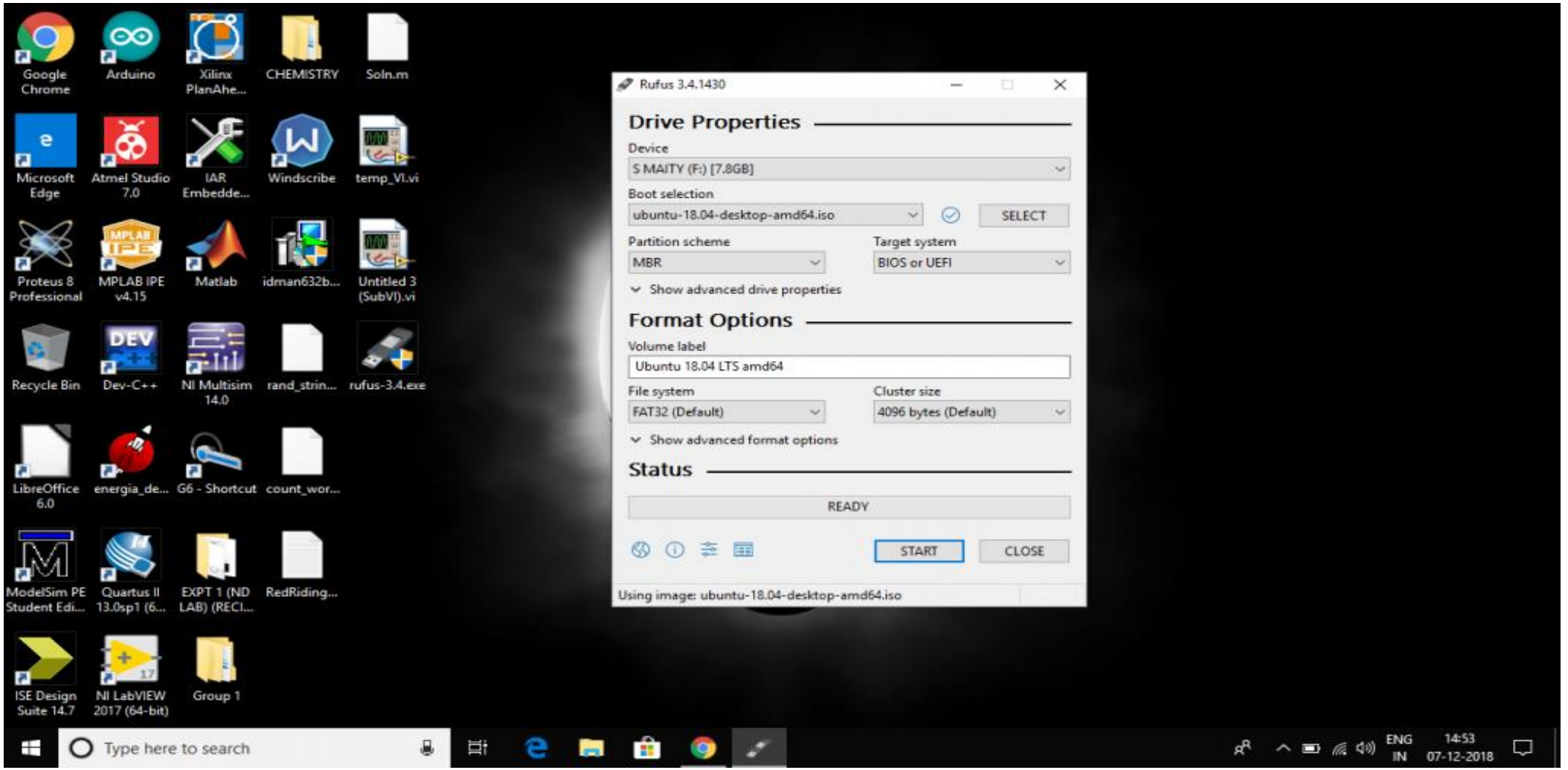


Specifying Amount to be Shrinked

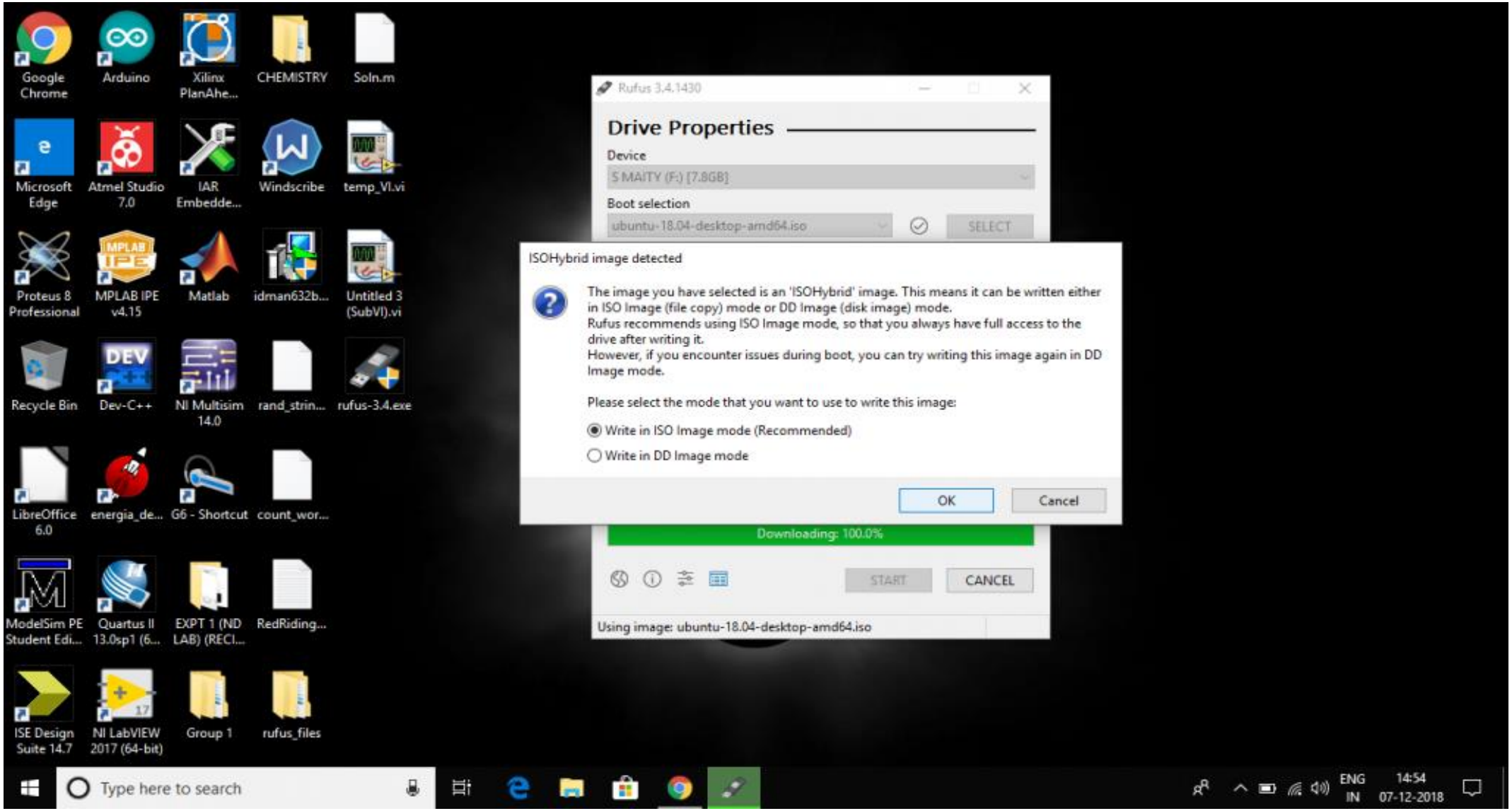
1. On the Disk Management Window right click on the Partition from which you want to extract the required (recommended 100 – 150 Gb) amount of free space, and click of ‘Shrink Volume’. In the Dialog Box that pops up enter the ‘amount of space to shrink’ as 102400 Mb (100 Gb) and continue. This would reduce the space of the present volume by 100 Gb and the remaining 100 Gb would be shown as Unallocated Space. This Completes the 1st Step.

STEP 2: Creating a Bootable Linux USB Drive. For the purpose of this article, i will be installing 'Ubuntu' as the Linux OS of choice as it is the most popular Linux OS and has a large support base. We first need to Download the Disc Image File (.iso file) of the Ubuntu OS from the Official Ubuntu website (www.ubuntu.com). The Latest LTS (Long Term Support) Version is preferred over the non – LTS version, as they provide update support for 5 year rather than 9 months for the non – LTS versions. We also need another software called Rufus for the creation of a Bootable USB Drive. A Bootable Drive is one from which the machine can load the OS without installing it. Then we need to backup all the Contents of the USB flash Drive and Start Rufus.

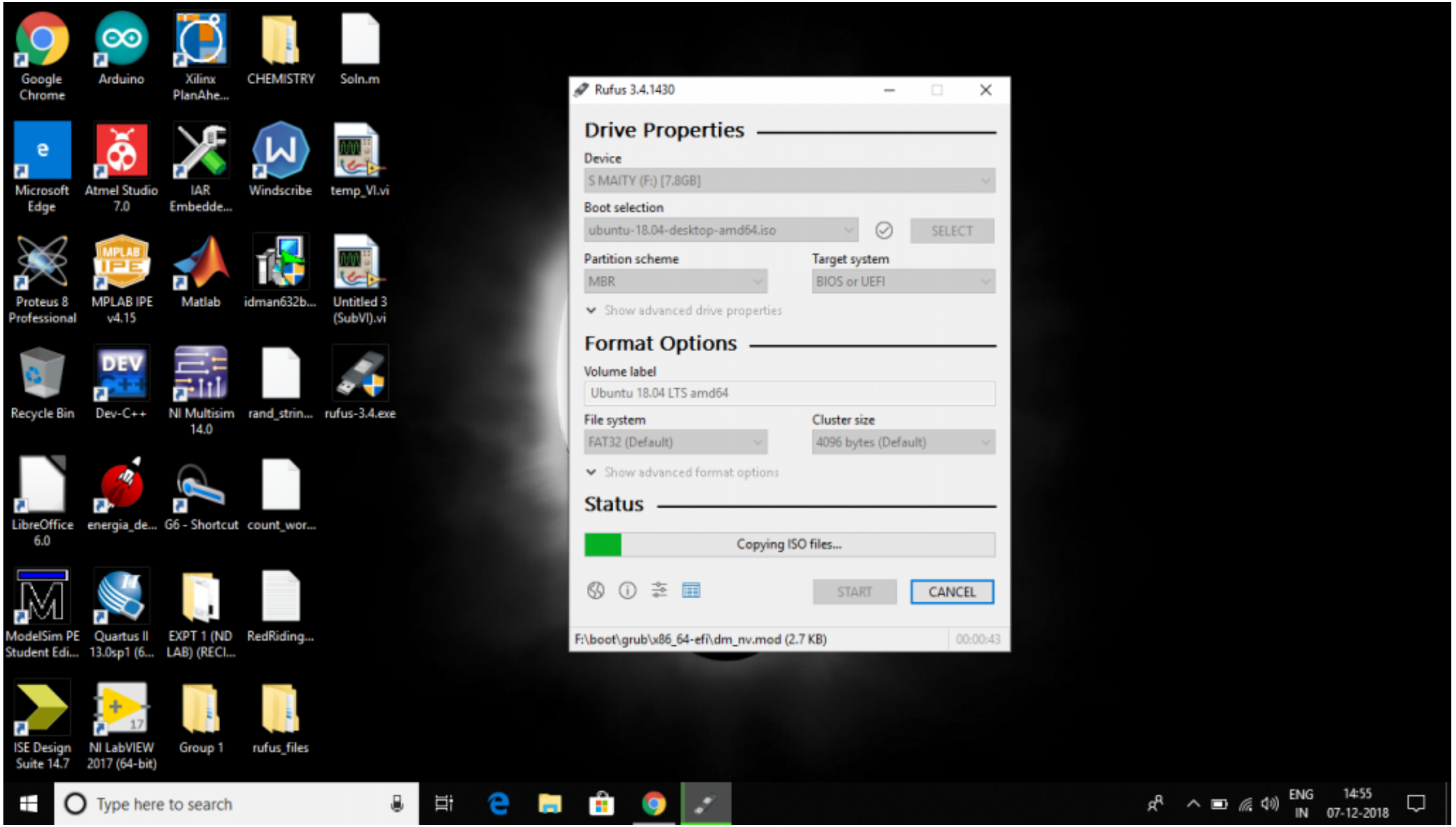
In Rufus we make the following settings a shown below.



Setting Up Rufus



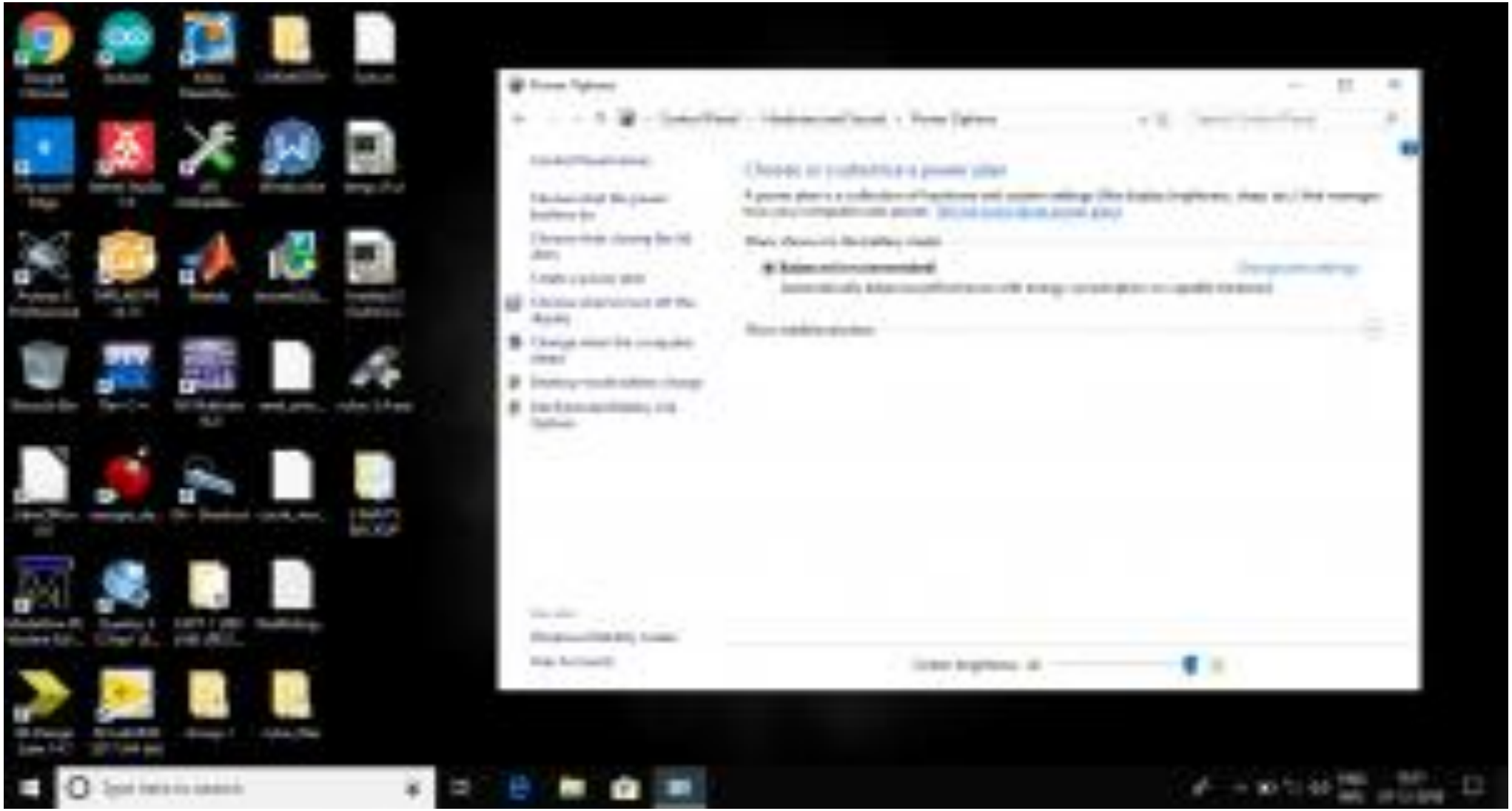
Writing Image in ISO mode



Creation of Bootable USB

After the completion of the above Process we are ready with our Bootable USB Drive.

STEP 3: Disabling Secureboot and Fastboot (Only for Newer Machines) This step is not required for older machines say, older than 2012. In this step we need to enter the BIOS/UEFI menu of our Computer. In case you are running Windows 8.1/10, before entering the BIOS/UEFI Menu you need to disable ‘Fast Startup’. For this go to the ‘Power Options’. On the left side options go to ‘Choose what the power buttons do’ then click of ‘Change Settings that are currently unavailable’ and Untick ‘Fast Startup’ Below and save these changes.



Power Options

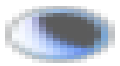
Control Panel Home

Choose what the power buttons do

Choose what closing the lid does

Create a power plan

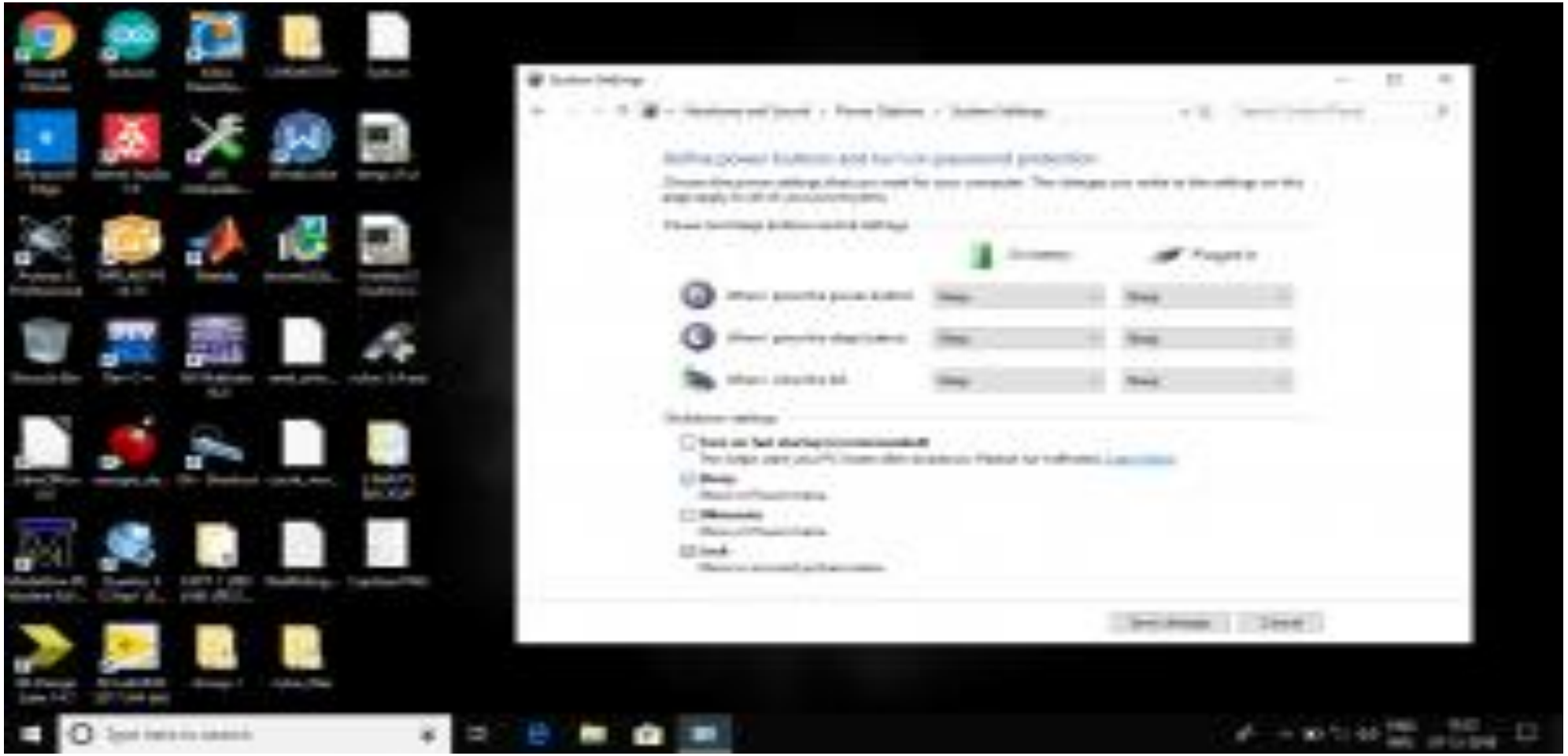

Choose when to turn off the display


Change when the computer sleeps


Desktop mode battery charge


Dell Extended Battery Life Options

Choosing what the power buttons do



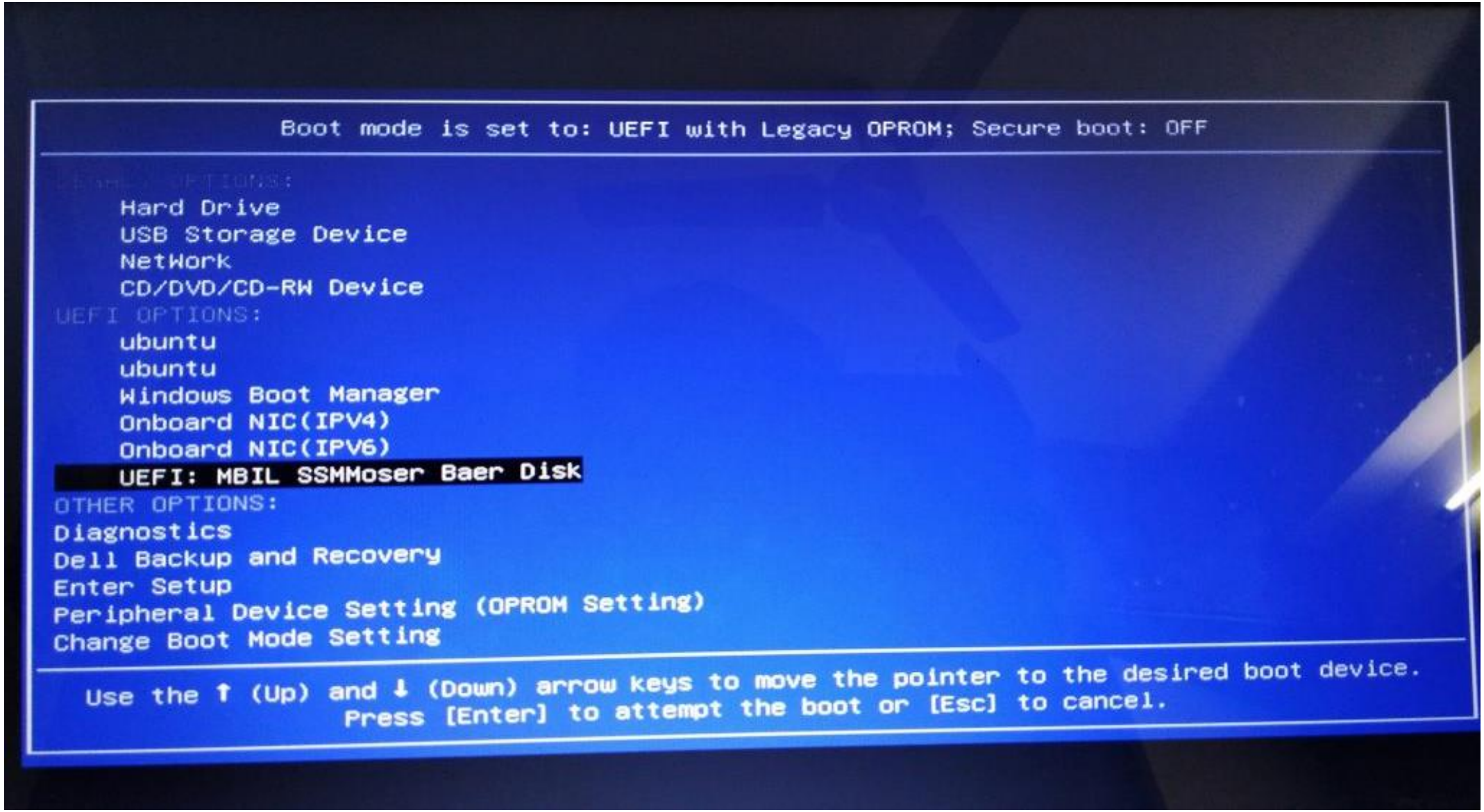
Turning Off Fast Startup

Now this step of the Process is manufacturer specific. I suggest you to Google a bit to get the exact key combination you need to press during the machine startup that will get you into the BIOS/UEFI Menu. In most Cases you need to press either one of the f1, f2 or f12 keys after turning on the machine but before windows start booting up.

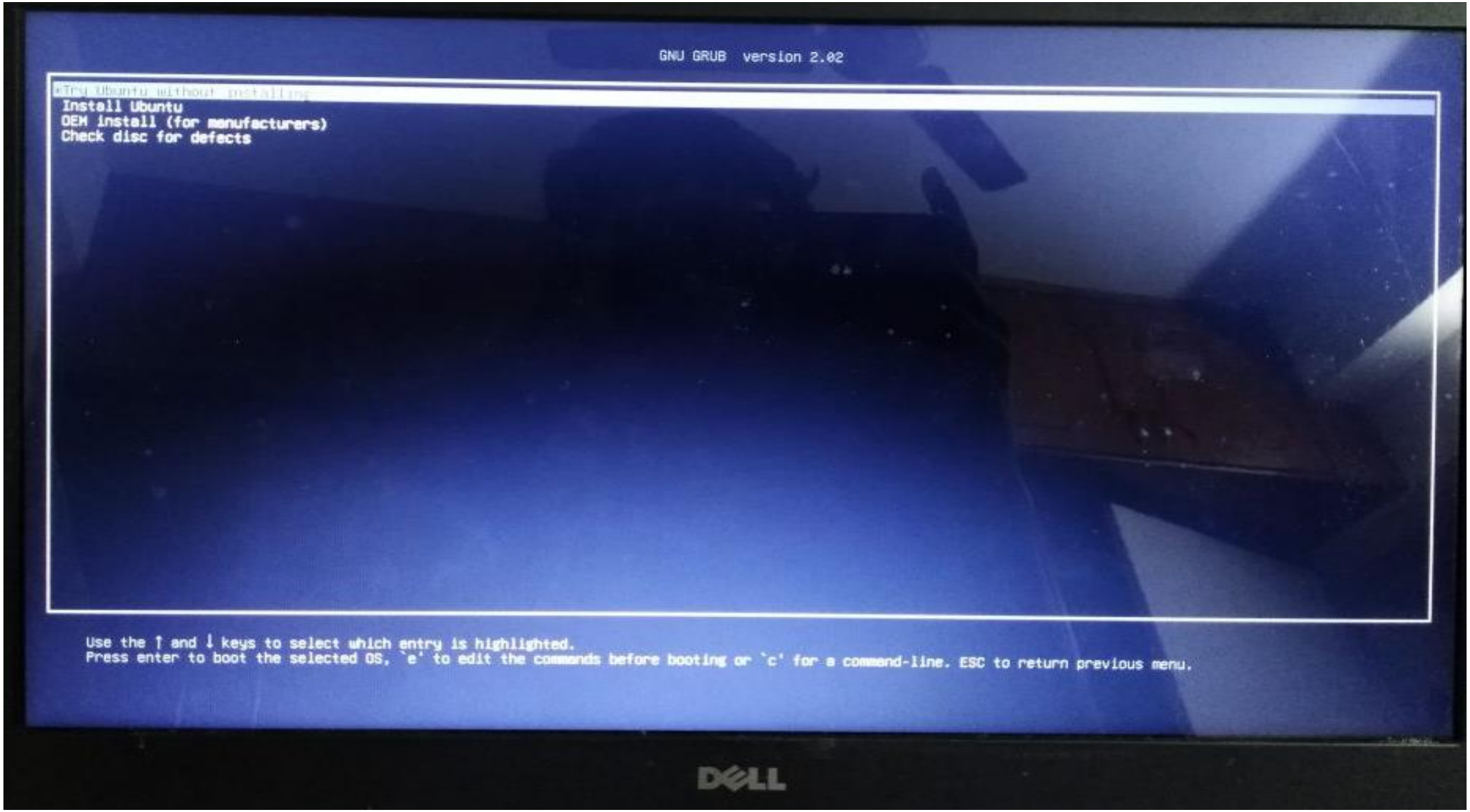
Disabling Secure boot

Disabling Fast boot

STEP 4: Booting Ubuntu Linux From USB Drive. In the next step we boot Linux from the USB Flash Drive we just created. To do this we enter into the 'Boot Options' during the machine starts up but before windows starts loading. The 'Boot Options Menu' is shown below:

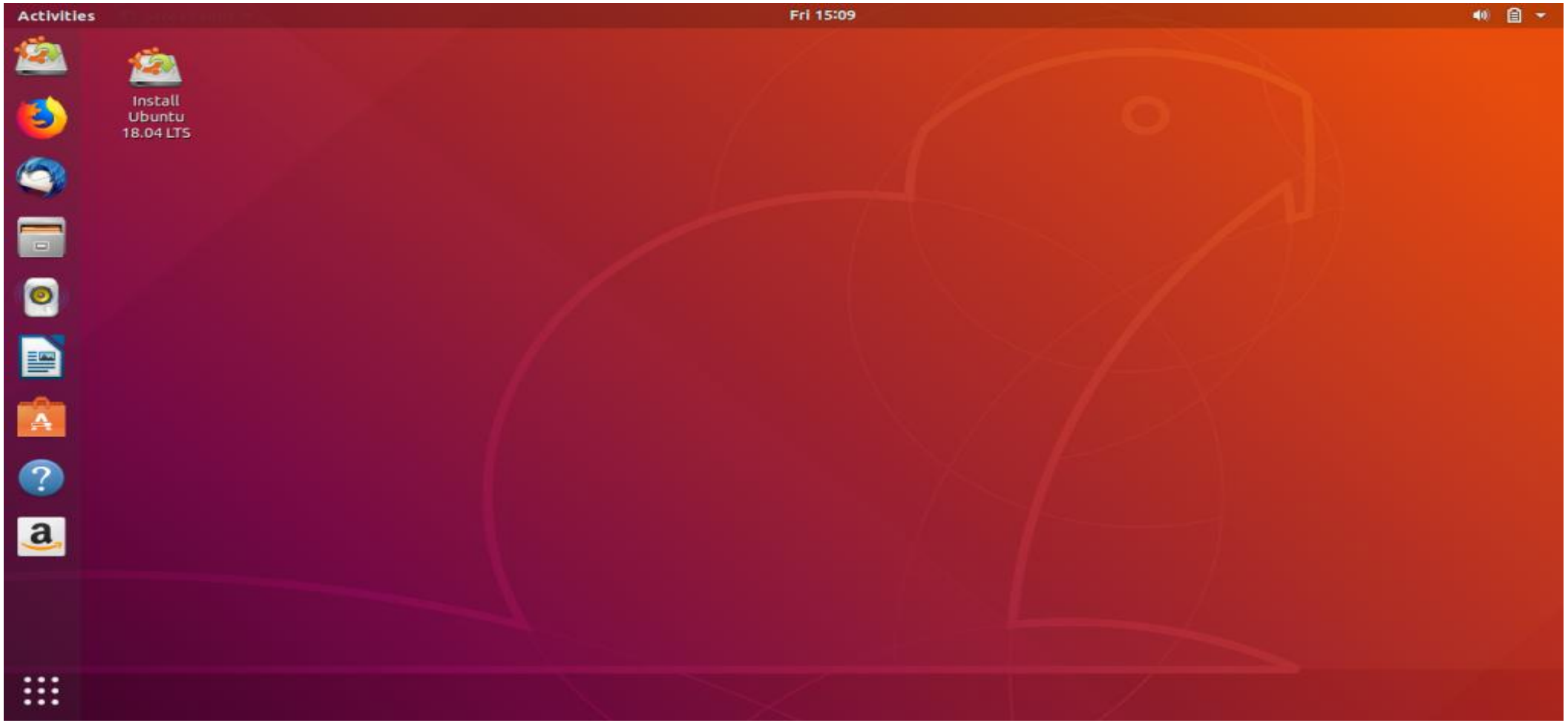


Boot Options Menu



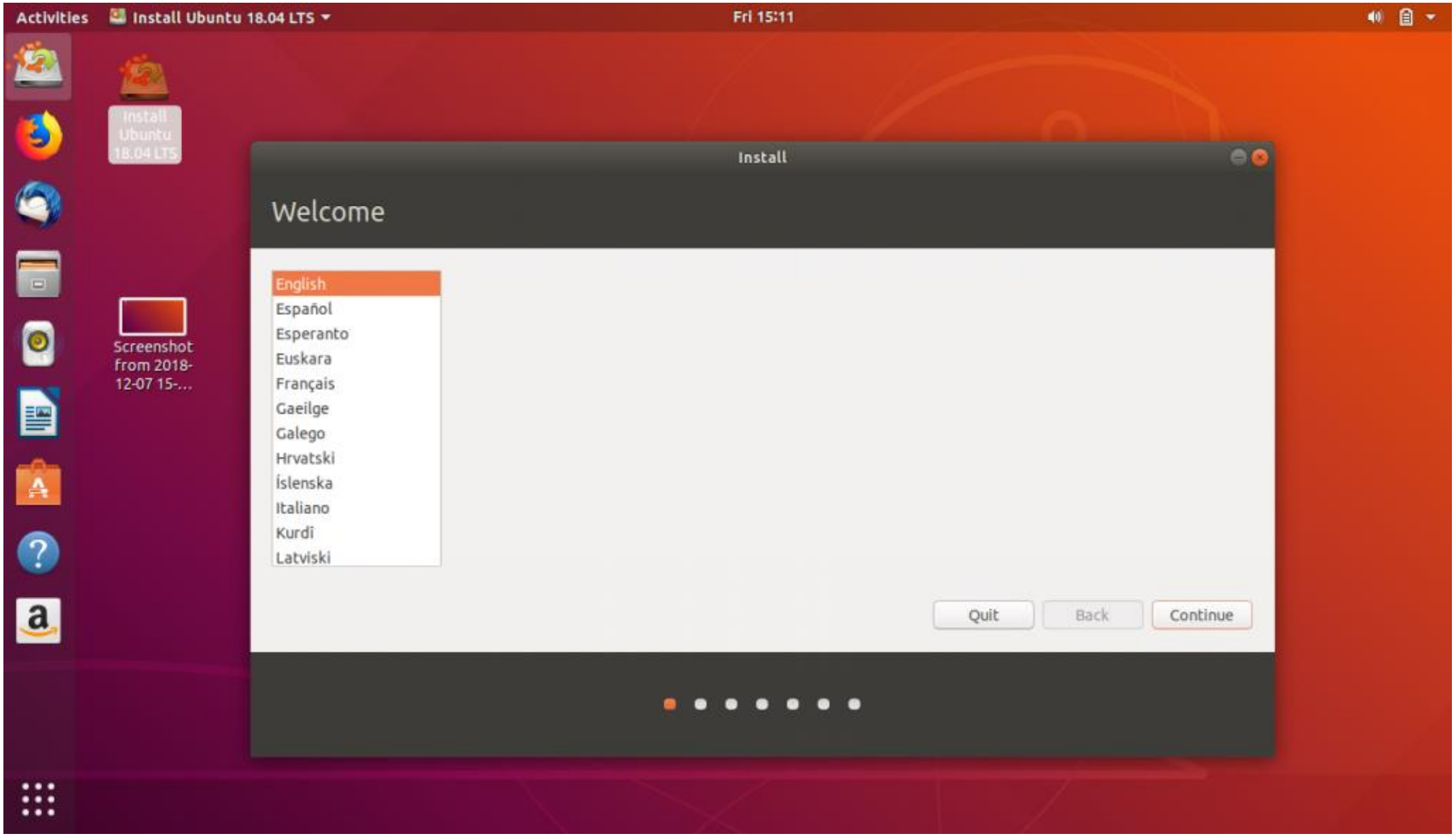
First Screen after USB Boot

First Bootup of Ubuntu Linux:

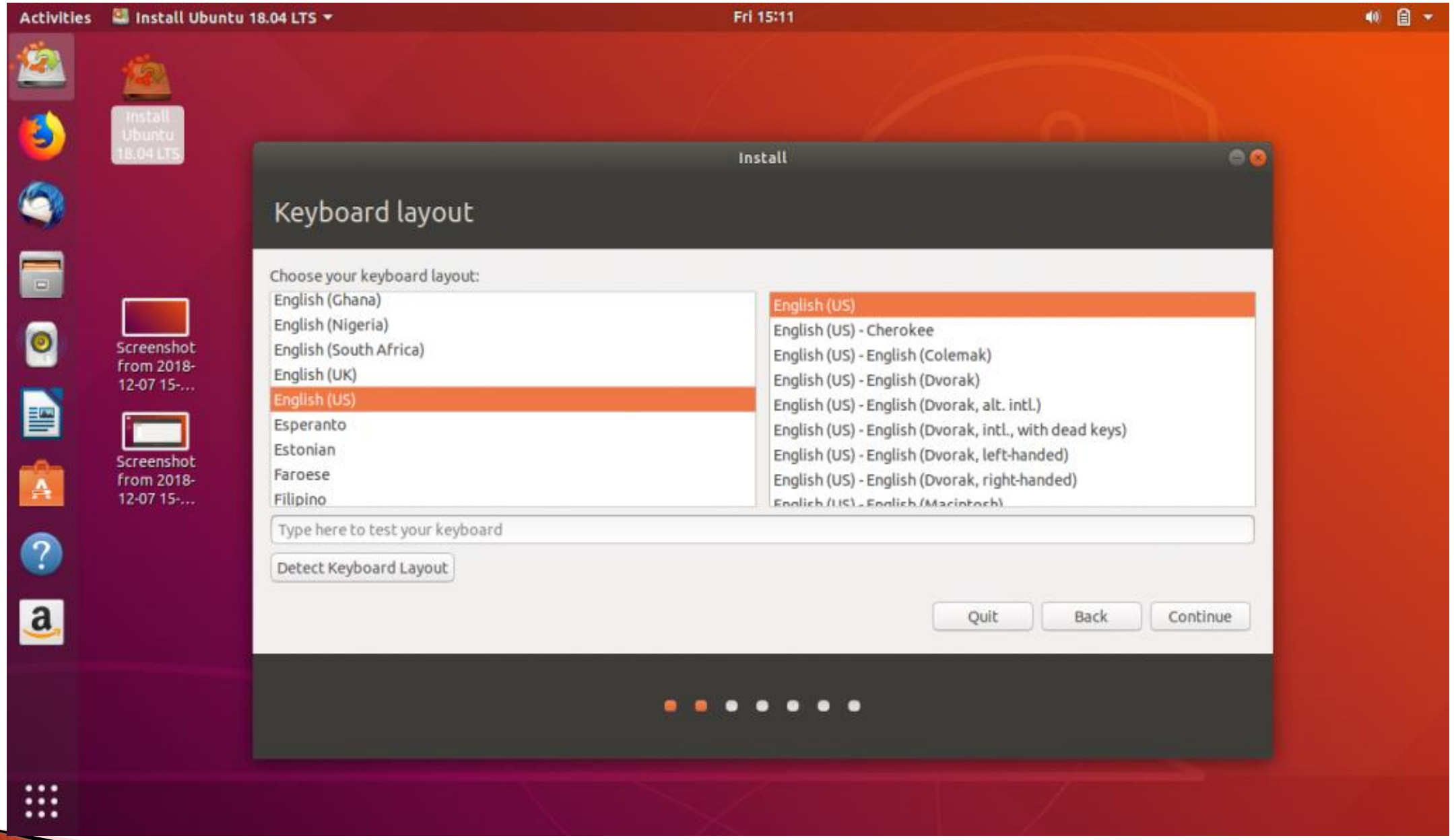


Ubuntu Desktop

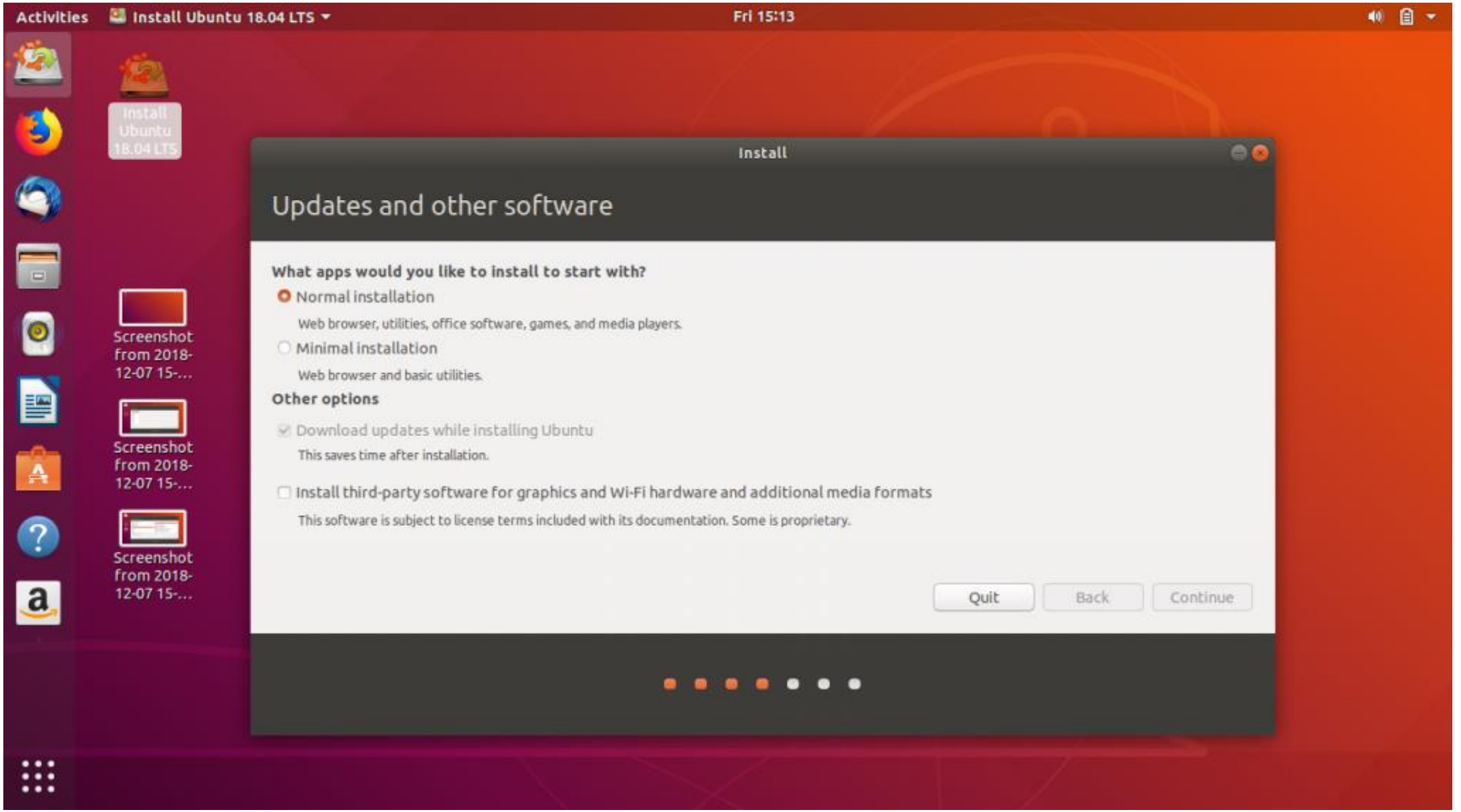
STEP 5: Installing Ubuntu Linux Once Ubuntu has booted off the USB Flash Drive you can try out the different applications that the Linux distribution has to offer and once you verify all the basic functionality of the OS that you will be using work on your machine, we can proceed with the installation. Run the 'Install Ubuntu' icon shown on the Ubuntu Desktop by Double Clicking it and Follow the steps as shown below.



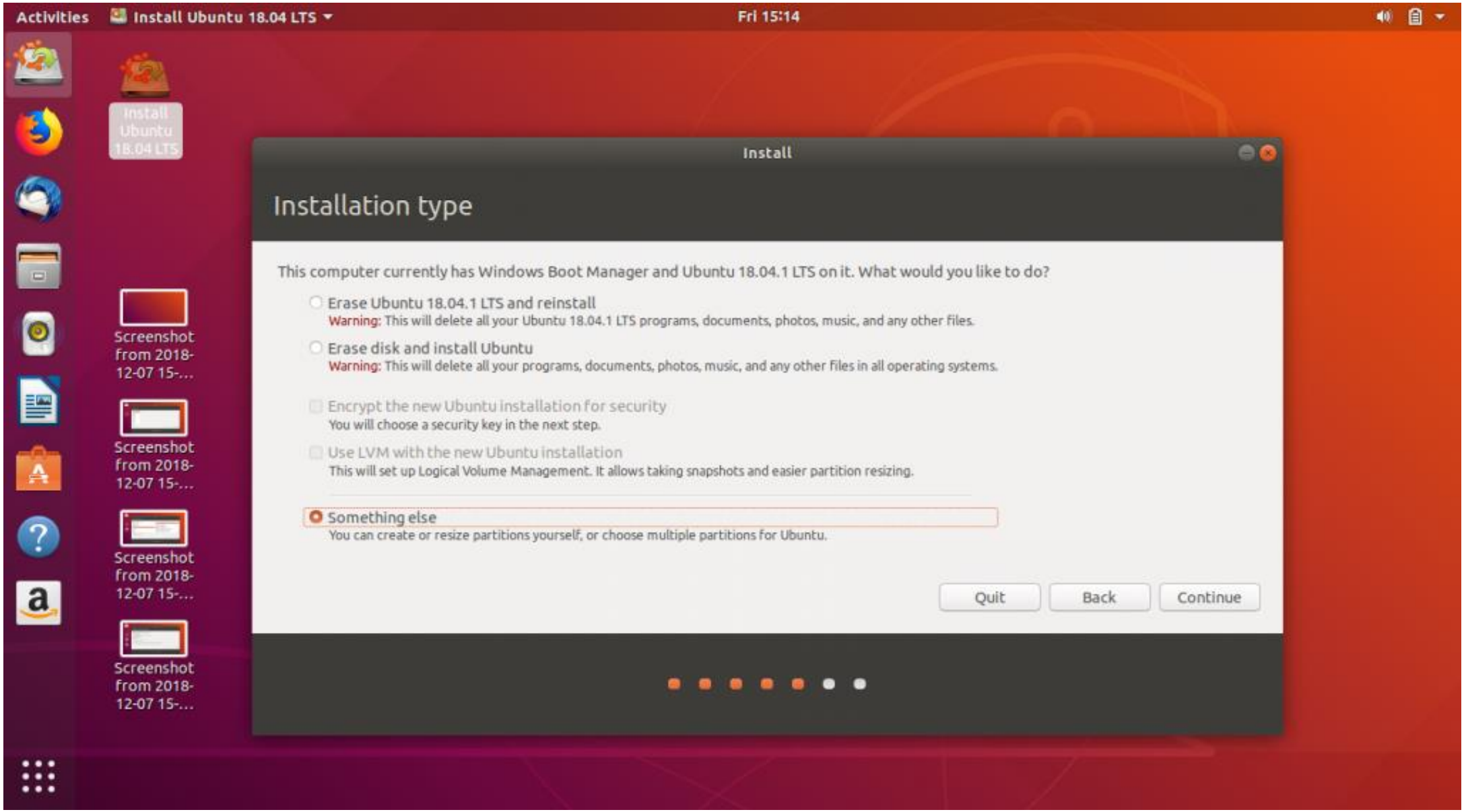
Selecting Language



Choosing Keyboard Layout

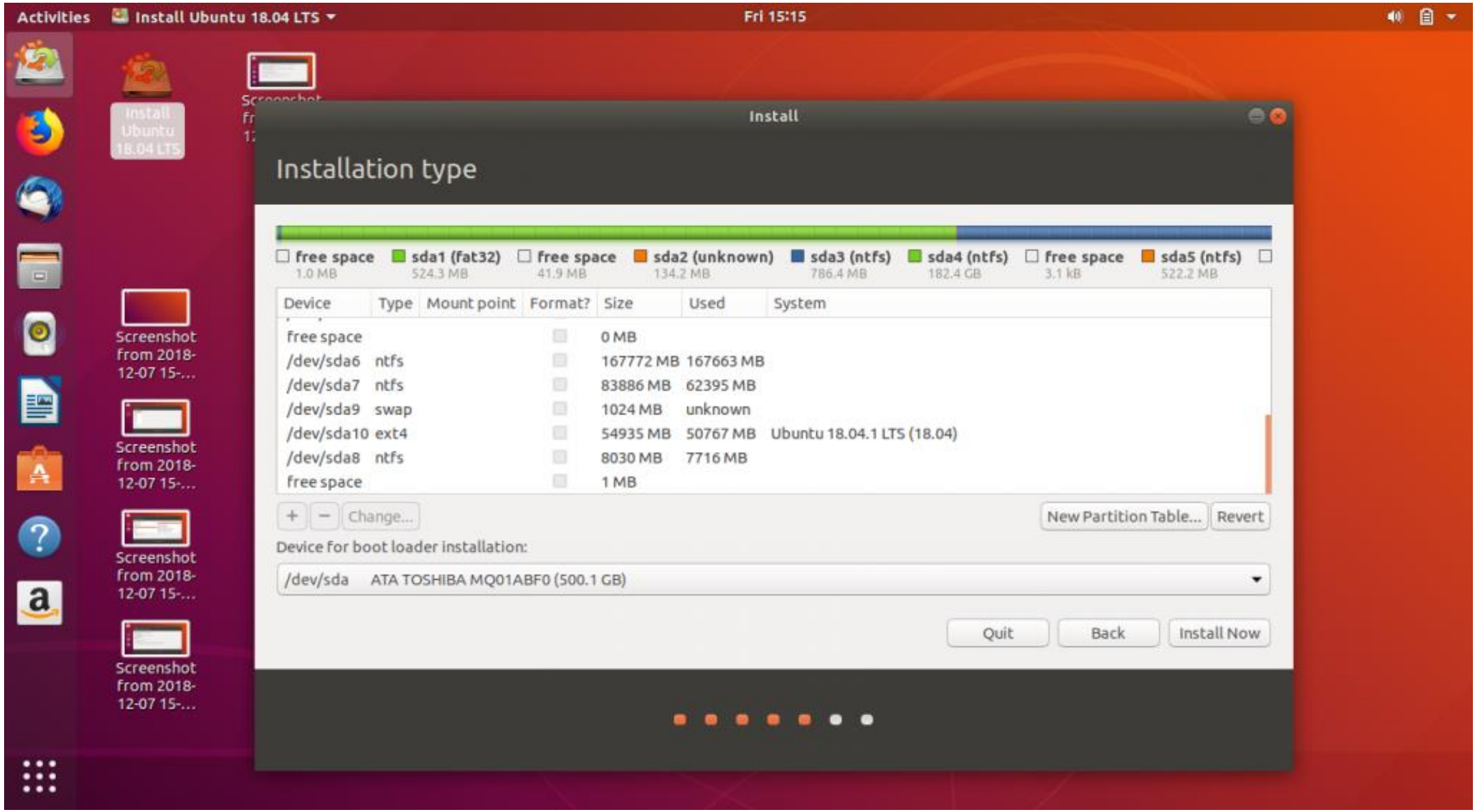


Choosing Type of Installation: Normal or Minimal



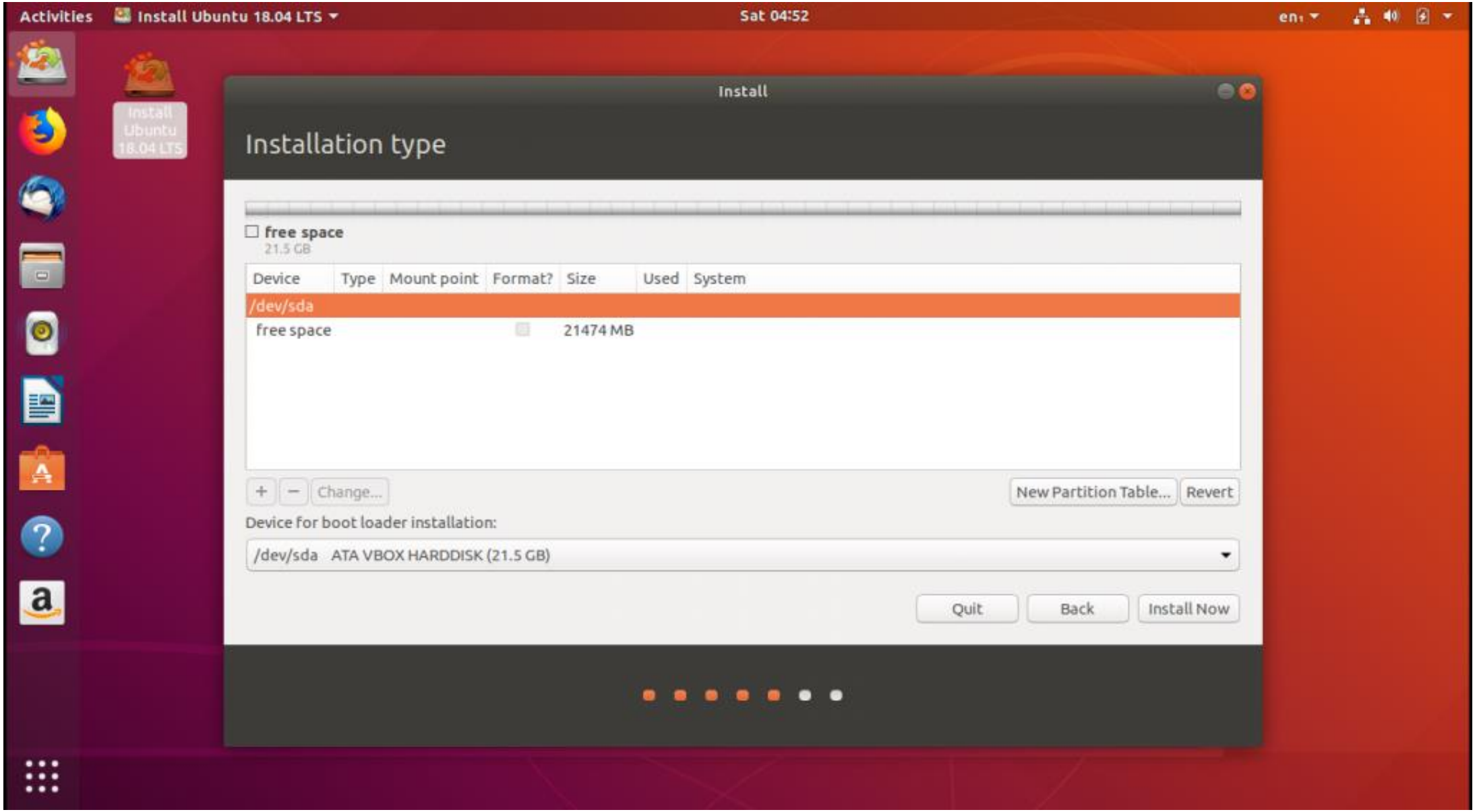
Installation Type: 'Something Else' is selected

Once You reach the Step ask you ‘How do you want to install Ubuntu’. Select ‘Something Else’ and continue. In the next screen you will see a window showing all the hard disk partitions on your machine. You need to find from among all the partitions, the one you unallocated in the 1st step, it will be labelled as free space in this window and it’s space should roughly match 102400 Mb. Highlight this ‘free space’ by left clicking it and then select the small ‘+’ sign below, this will pop a dialog box for the creation of a new Partition.

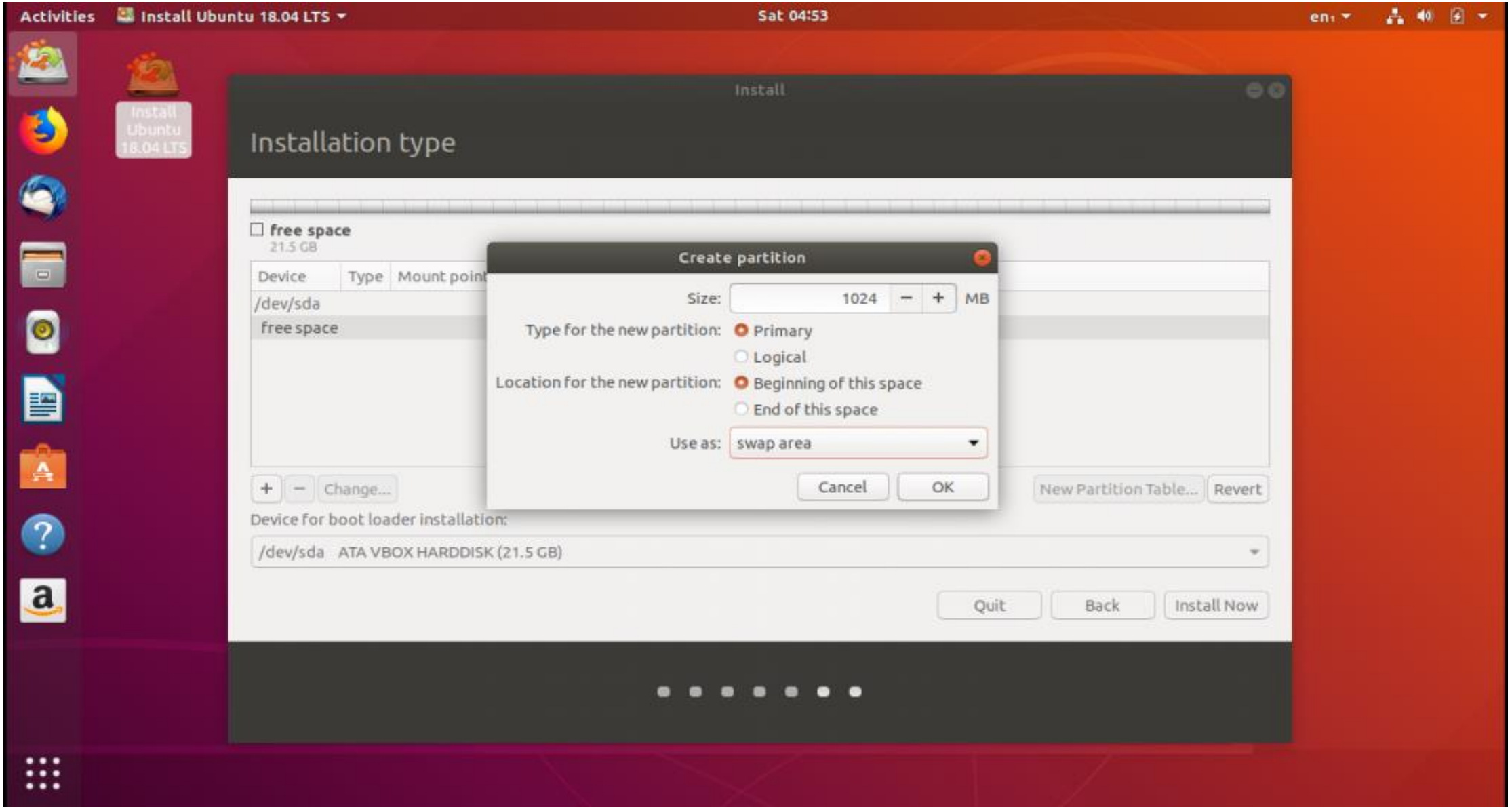


Modifying the Partition Table

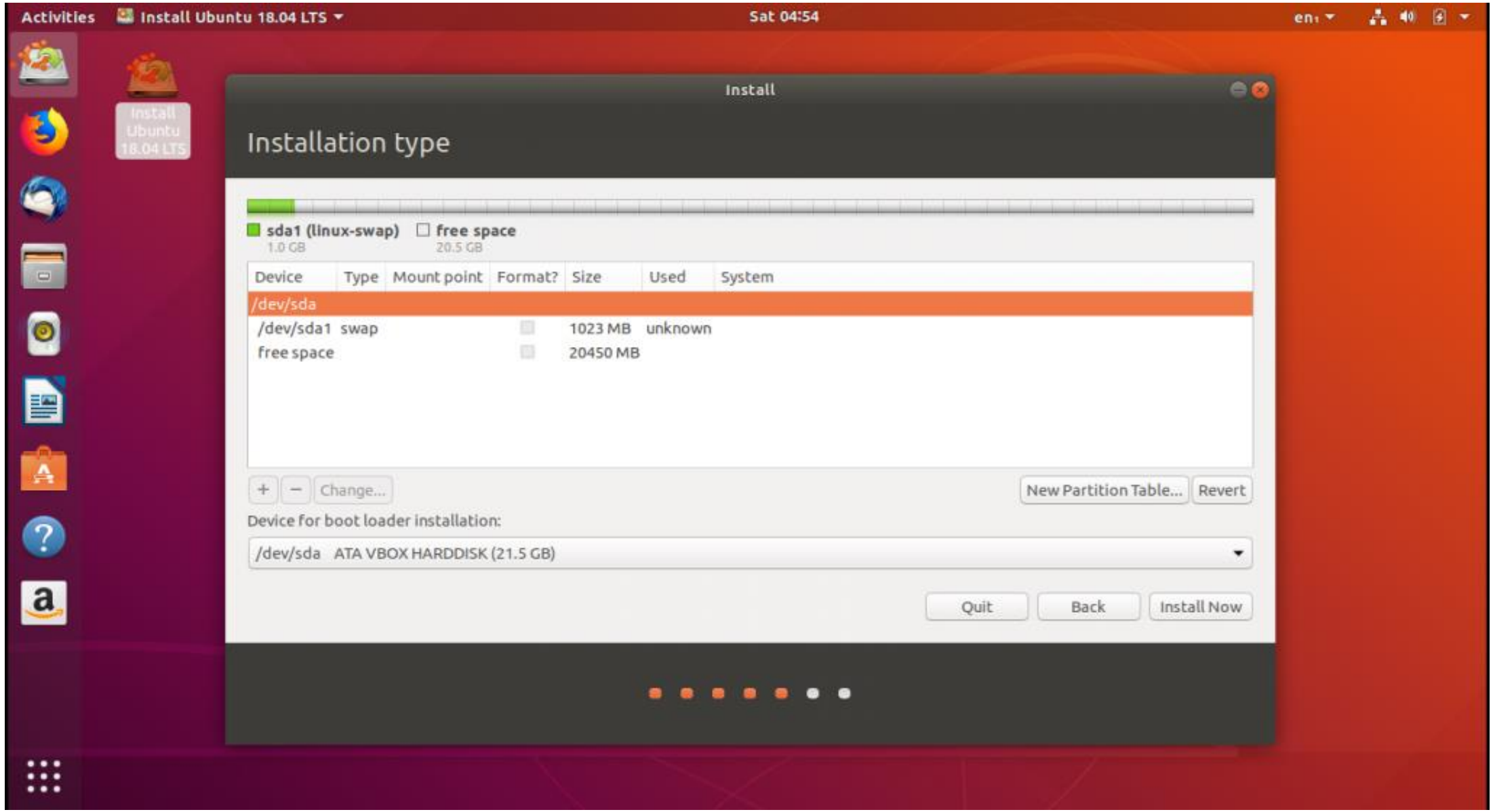
Now since i already have Linux installed on my system you see that there are 2 partitions of types ‘swap’ and ‘ext4’ these two are the Linux Partitions. But if you didn’t have Linux installed previously you would only find a portion labeled as free space as mentioned above. 1st we will be creating a ‘swap area’ we highlight the ‘free space’ open the new partition creation window and enter the amount of space for swap as equal to the amount of RAM you have. then select the ‘New Partition type’ as ‘Primary’ and finally select ‘Use as’ ‘swap area’. Now as i already mentioned since i already have Linux installed, from now on i am going to continue the installation process in a virtual machine and hence the exact partition table and partition sizes may not match.



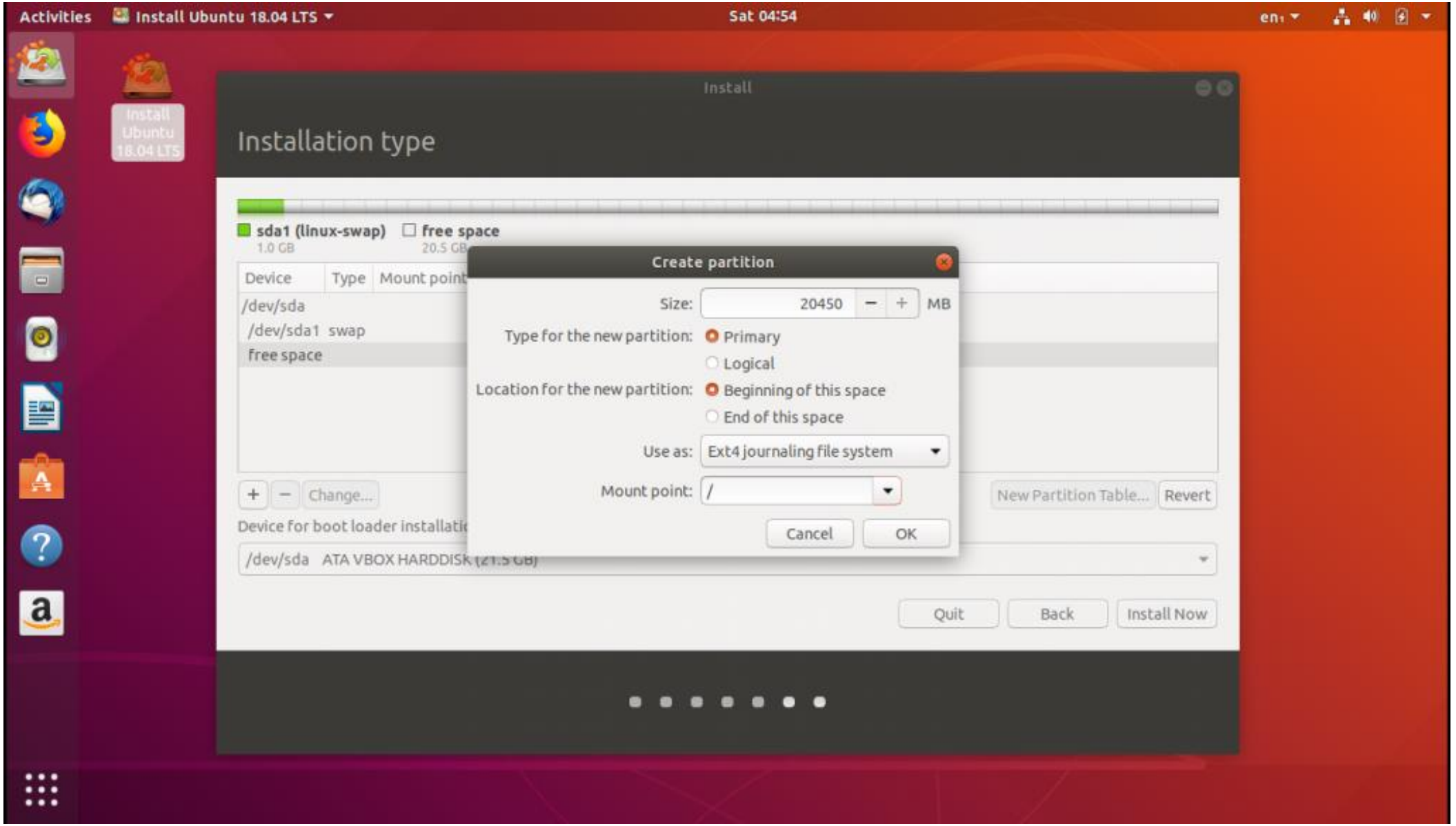
Free Space (Exact Partition Sizes would not match)



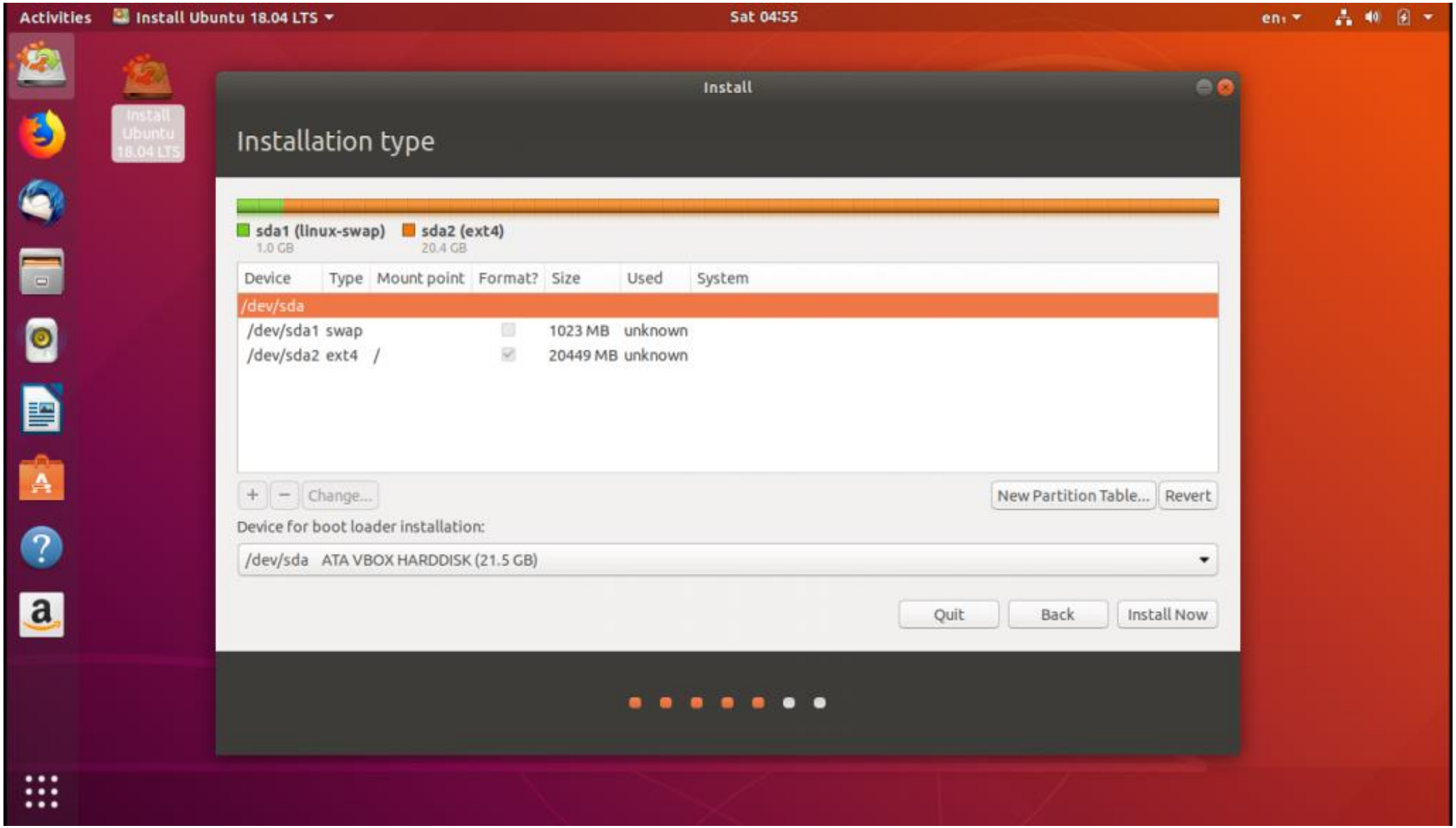
Creating swap area



'swap area' created

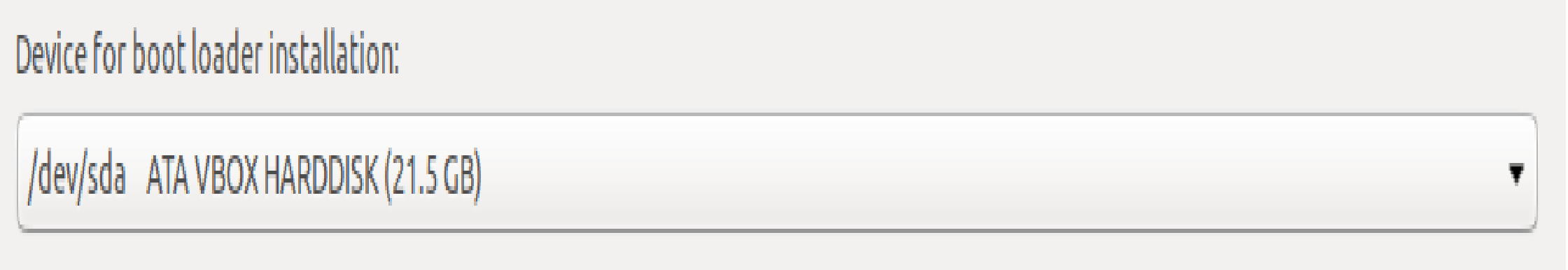


Creating Root File System Partition



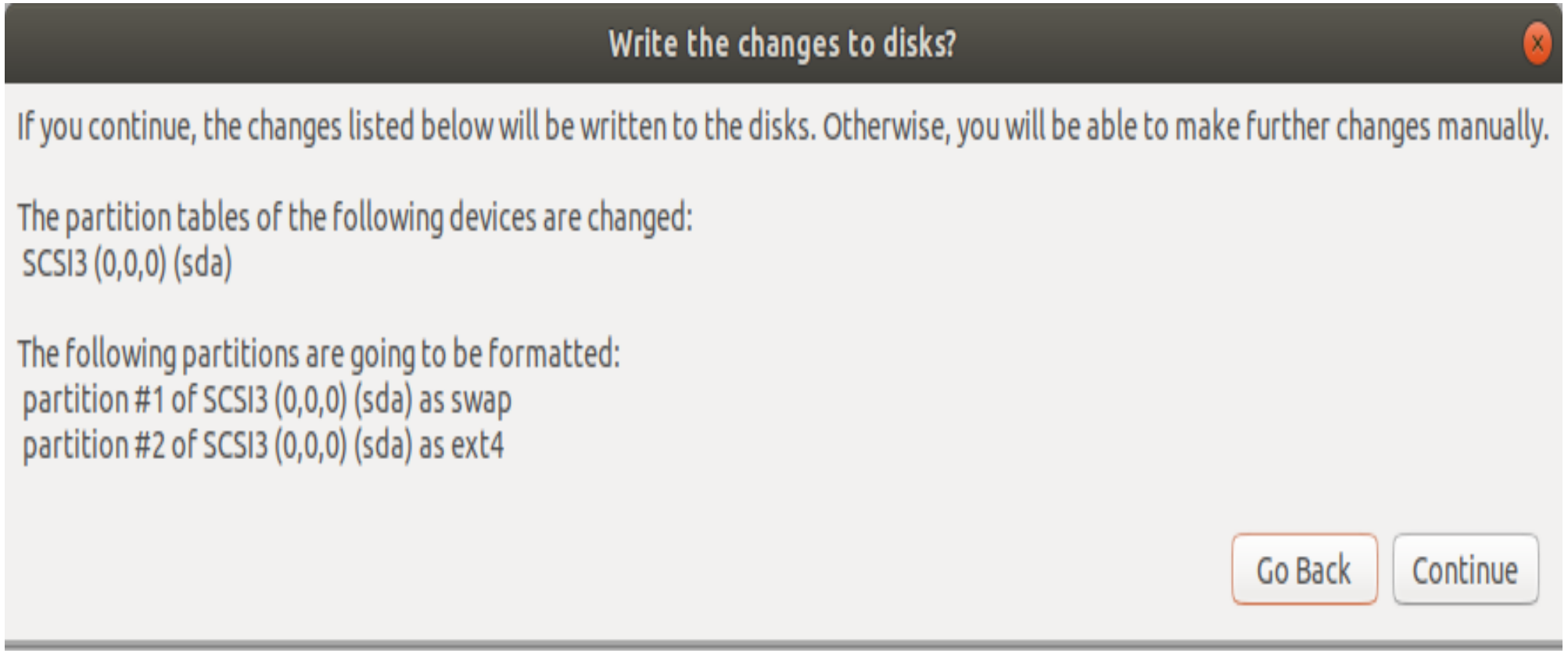
Partition Table Created

Also make sure your ‘Device for Bootloader Installation’ is set to your internal Hard Disk Drive. (Where Windows is installed and Linux is going to be installed) in most cases this is ‘/dev/sda’



Device for Bootloader Installation

After this is created you press ‘Install Now’ and confirm you changes to the Partition Table as shown below:

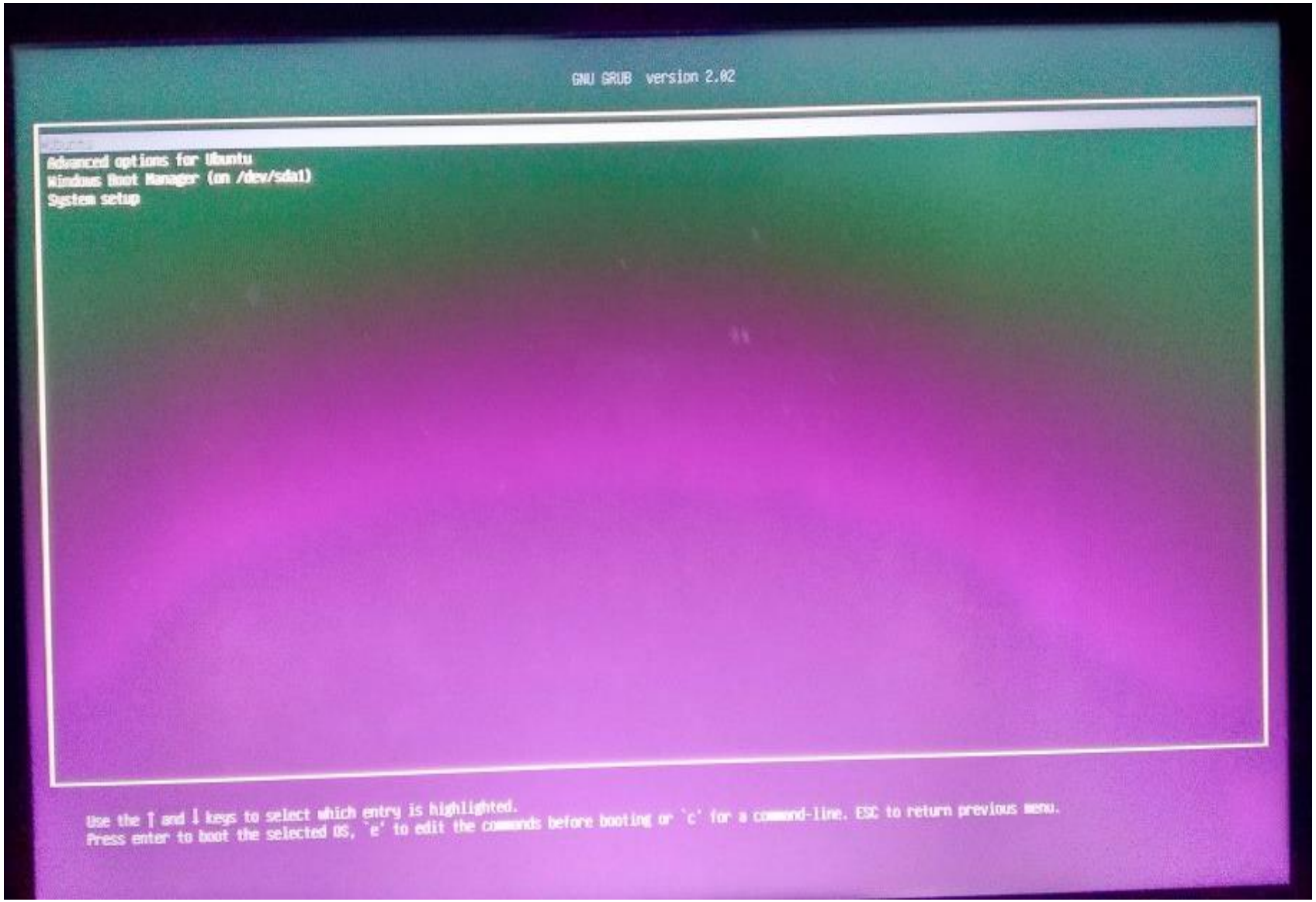


Confirmation of Modifications (Make sure both the partitions you created out of the free space are listed)

After all this is done you simply need to proceed as follows till the installation starts:

Selecting Your Timezone

After this you simply need to continue and give your login credentials (Username, Password) and then the installation begins. After the installation completes you can shut down your computer, and remove the bootable USB drive and the next time turn on your machine you should be presented by a grub window as shown below, letting you choose between Linux (Ubuntu, 1st Option) and Windows (Windows Boot Manager). And there you have your own dual booted system.



The Grub Menu

THANK YOU

hitbullseye