**BAHIRDAR UNIVERSITY**

**BAHIR DAR INSTITUTE OF TECHNOLOGY**

**(BiT)**

**FACUITY OF computing**

Operating system and system programming

Individual Assignment

Name: Tsehay Adugna

Section: A

ID Number :1602632

Title: Raspberry pi OS

Submission

date:16/08/2017

Submitted to:Lec Wendmu B

# Table of Contents

# Introduction

## Background

The Beginnings of the Raspberry Pi The Raspberry Pi was developed in the United Kingdom by the Raspberry Pi Foundation along with Broadcom. The original goal of the Raspberry Pi project was to provide an inexpensive tool for teaching basic computer science in schools and in developing countries. When the first model was released in 2012, it quickly gained popularity. In fact, it was much more popular than the Raspberry Pi Foundation expected and saw a large number of sales even outside its target market for uses such as robotics and others.

What sets the Raspberry Pi apart from regular computers? Typically, a computer is made up of a main system board, called a motherboard or logic board, and several other components plugged into it. You don't see all of these components, because they're enclosed in a case. The Pi, on the other hand, is a single-board computer, or SBC. An SBC is a complete computer built onto one circuit board. All the components needed for a fully functional computer, including the processor, memory, video chipset, storage, and so forth are built onto the SBC. This usually allows it to be much more compact and often less expensive. These microcomputers are designed to run any ARM- based Linux distribution for an operating system.

Since the first iteration, the Raspberry Pi Model B, more than 40 million boards have been sold. Today, millions use the Raspberry Pi for anything from learning programming from the ground up to serving as a fully fledged desktop PC. Of course, there are many uses in between.

Raspberry Pi OS is a free, Debian-based operating system optimised for the Raspberry Pi hardware. Raspberry Pi OS supports over 35,000 Debian packages. We recommend Raspberry Pi 0S for most Raspberry Pi use cases. Because Raspberry Pi OS is derived from Debian, it follows staggered version of the Debian release Cycle. Releases happen roughly every 2 years. The latest version of Raspberry Pi OS is based on Debian Bookworm. The previous version was based on Debian Bulseye

Raspberry Pi OS (formerly known as Raspbian) is the official operating system for the Raspberry Pi, a series of small, affordable single-board computers developed by the Raspberry Pi Foundation. It is optimized for the Raspberry Pi hardware and is the most widely used OS for these devices. Here's an overview of Raspberry Pi OS:

## Motivation

The primary motivation behind the Raspberry Pi was to revive interest in computer science and programming, especially among young students. In the early 2000s, the creators noticed that fewer students were applying to study computer science, and those who did lacked hands-on programming experience. Computers had become more like consumer devices, with limited opportunities for tinkering and experimentation. The Raspberry Pi Foundation, led by Eben Upton, set out to create an affordable, programmable computer that could help people, especially children, learn how computers work and how to write code.

Another motivation was to make computing accessible to everyone, regardless of their economic background. Traditional computers were often too expensive or complex for people in low-income or developing regions to use for learning and innovation. The Raspberry Pi, priced as low as $35, aimed to break down those barriers. Its small size, low power consumption, and low cost made it ideal for classrooms, makerspaces, and home use. It allowed people to explore not just software development but also hardware interfacing through GPIO pins, enabling a hands-on approach to learning.

Lastly, the Raspberry Pi was designed to foster innovation and creativity across various fields, not just education. While its core mission was academic, the Raspberry Pi quickly became a favorite among hobbyists, engineers, and developers for prototyping projects, automating tasks, and exploring technologies like IoT and robotics. The open-source nature of its software and the support of a global community helped it evolve into a powerful platform for experimentation, real-world applications, and even industrial use. This wide-ranging potential has helped sustain its growth and relevance far beyond its original educational purpose.

## Objective

The primary objective of the Raspberry Pi is to promote the teaching and learning of computer science and programming skills in an accessible and affordable way. By providing a low-cost, credit card-sized computer, the Raspberry Pi aims to enable students, educators, and hobbyists to experiment with computing and electronics without the financial barrier of expensive hardware. It supports various programming languages and educational tools, allowing users to develop practical skills in software development, hardware interaction, and system design.

Another key objective is to foster innovation and creativity across different fields by offering a versatile platform for building a wide range of projects. From home automation systems and IOT devices to media centers and robotics, the Raspberry Pi empowers users to prototype, test, and deploy real-world applications. Its open design, extensive documentation, and supportive community make it an ideal tool not only for learning but also for professional and experimental use, helping to bridge the gap between education and industry.

# Requirements

## Hardware

To operate effectively, the Raspberry Pi requires a basic set of hardware components. At its core is the Raspberry Pi board itself, which includes a processor, RAM, USB ports, HDMI output, GPIO pins, and network connectivity options. Additional essential hardware includes a microSD card (typically 8GB or more) that acts as the primary storage and operating system drive, a 5V micro-USB or USB-C power supply depending on the model, and peripheral devices such as a keyboard, mouse, and monitor for user interaction. For more advanced projects, hardware like sensors, cameras, displays, and breadboards can be connected through the GPIO pins to extend functionality.

## Software

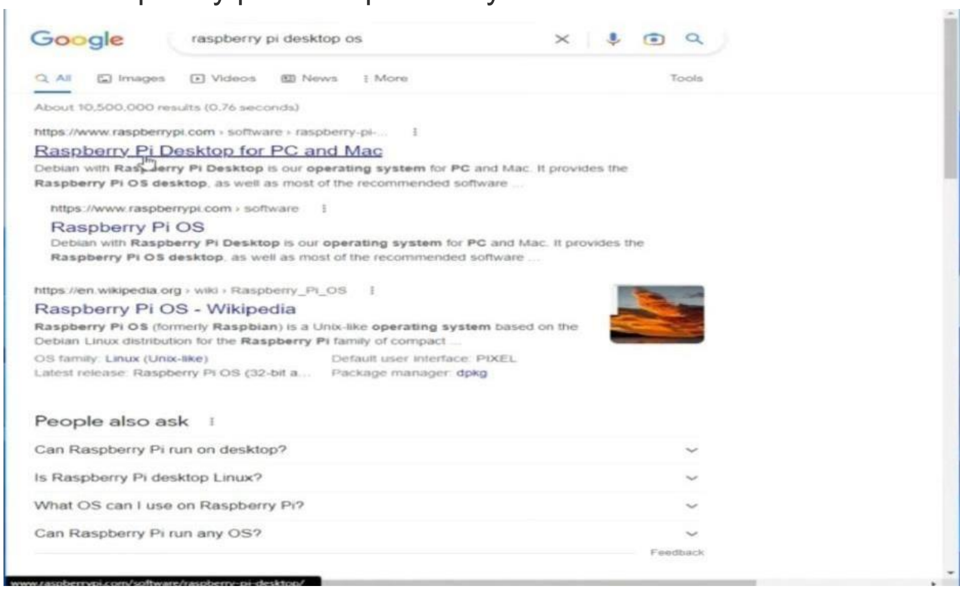
On the software side, the Raspberry Pi primarily runs a Linux-based operating system called Raspberry Pi OS (formerly Raspbian), which is optimized for the device. The OS includes a graphical user interface and comes preloaded with educational tools, programming environments (like Python, Scratch, and Thonny), and general-purpose software such as web browsers and office tools. Depending on the project, users can install other operating systems like Ubuntu, RetroPie for gaming, or specialized OS images for media centers and IoT. Software management is handled through tools like APT (Advanced Package Tool), and users can write their own programs or use preexisting libraries to develop applications.

# Installation steps of raspberry pi using virtual box

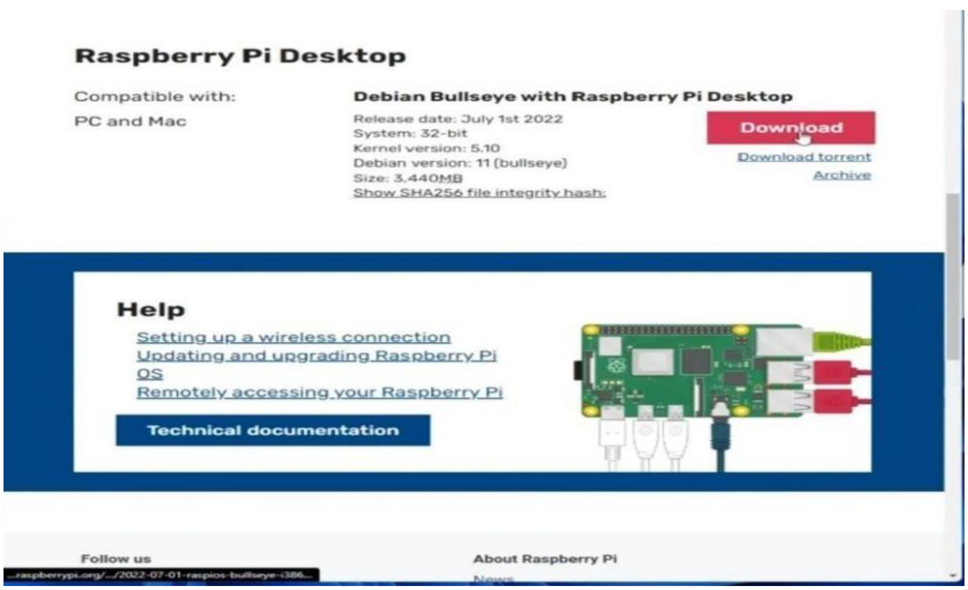
There are many operating systems you can use on a Raspberry Pi, but the Raspberry Pi hardware is ARM-based, and Virtual Box doesn't emulate ARM hardware. So to create a virtual machine you need to use an Raspberry Pi operating system that also provides a version that supports x86 hardware. The official operating system for the Raspberry Pi called Raspberry Pi OS. It's based on Debian Linux and you can download a version for PC and Mac. That's the version we'll use.

First download the virtual machine and the raspberry pi desktop on your computer. Once you have all the components you need, use the following steps to create the boot disk you will need to set up your Raspberry Pi.

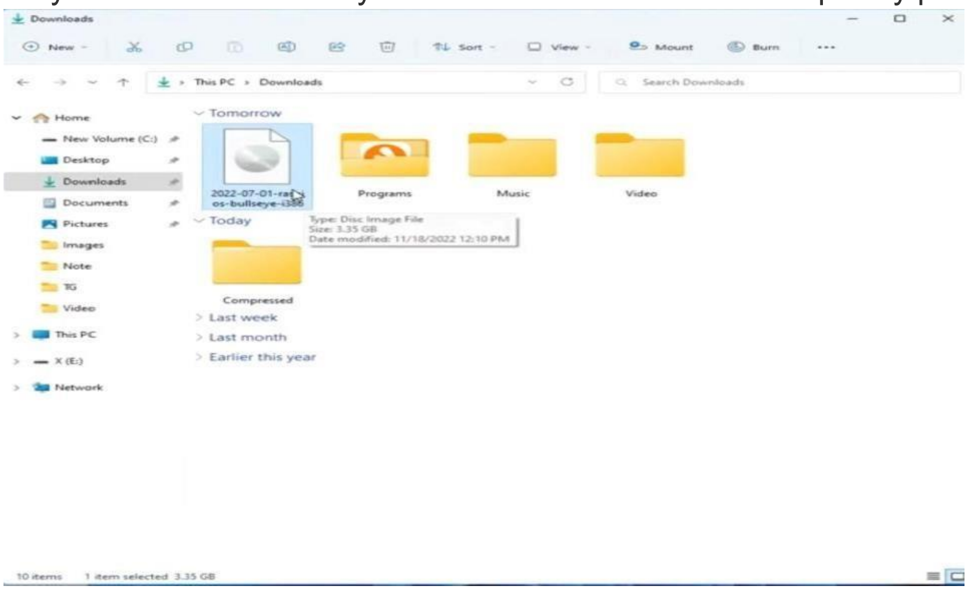
1.Search raspberry pi desktop Os on your browser



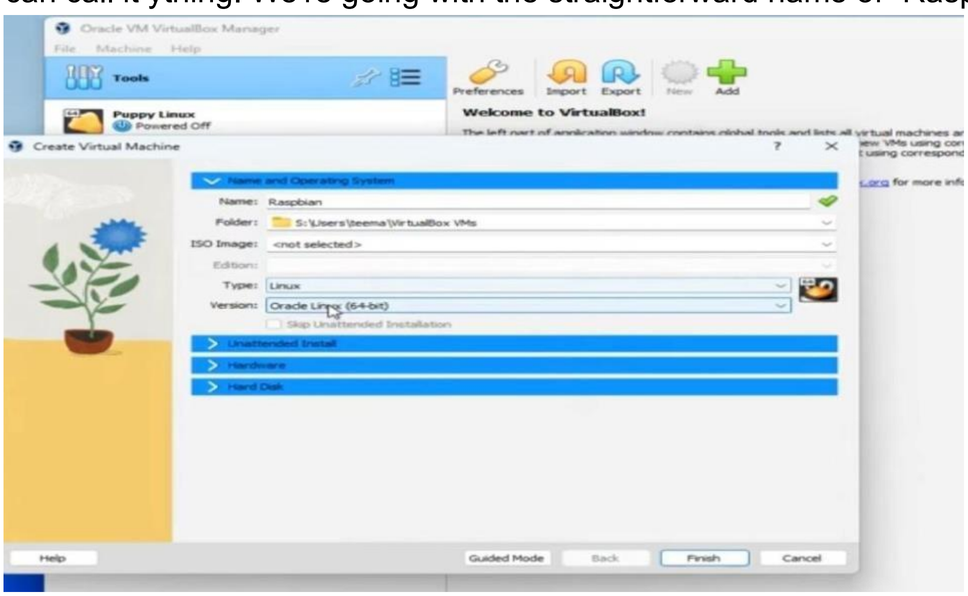

2.click the download button and it will start downloading



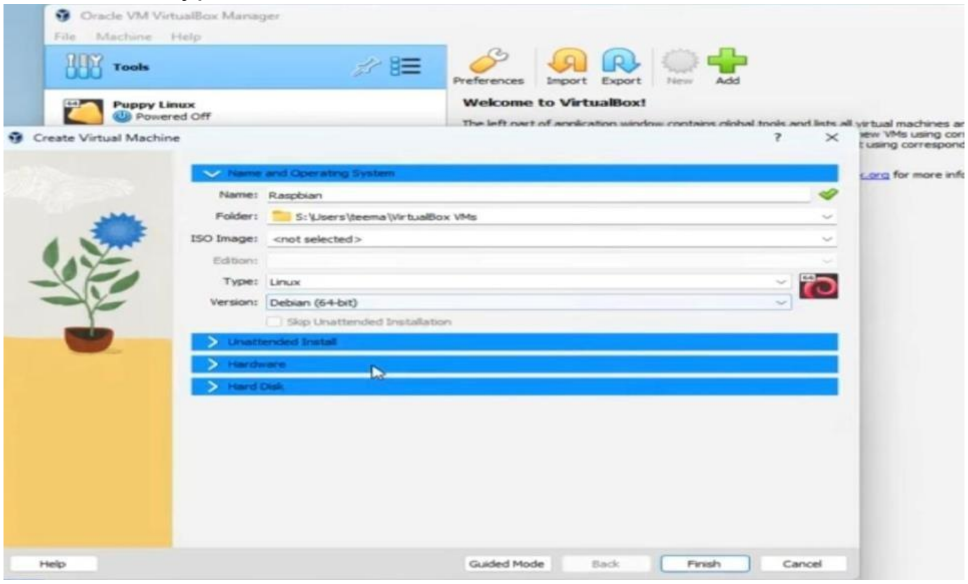
3.On your download folder you can see the downloaded raspberry pi Os



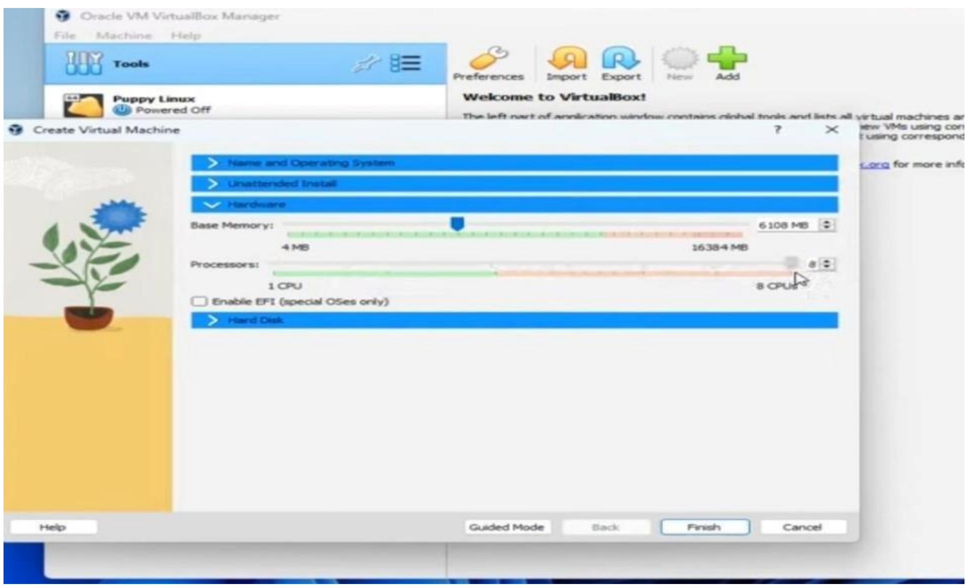

4.Start VirtualBox, and click "Machine" then "New." We need to give a name to the virtual machine. This is just a label that VirtualBox uses to identify the machine, so you can call it ything. We're going with the straightforward name of “Raspbian".



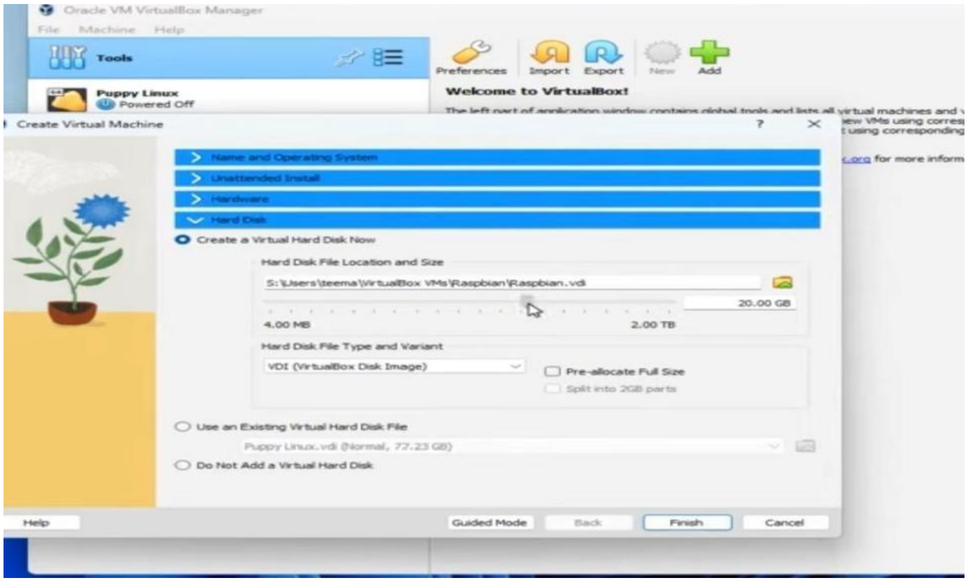
5.Now set the type to Linux and then let's set the version to Debian



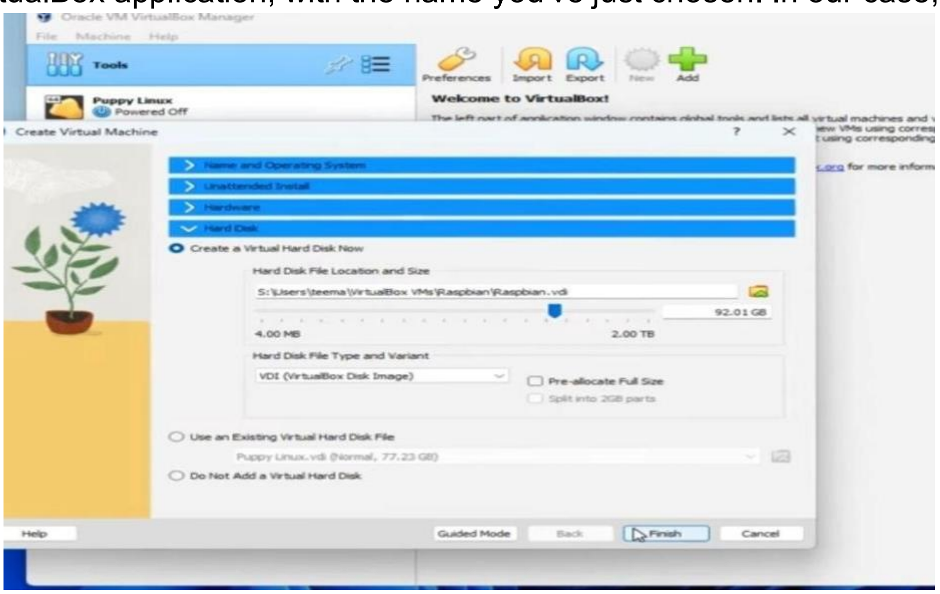

5.The “Hardware" page lets us set the RAM and number of CPU cores we're devoting to this virtual machine.

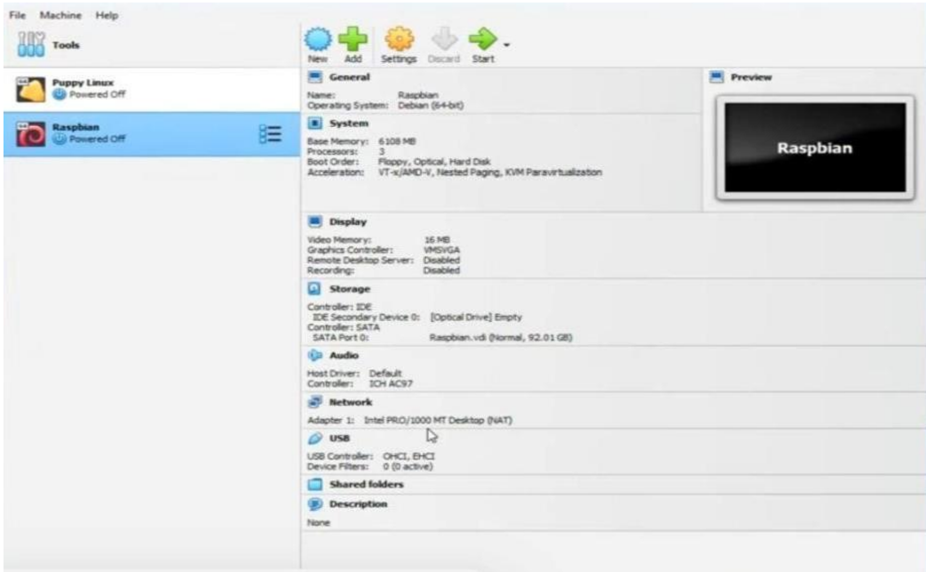



6.The "Virtual Hard Disk” page lets you select the size of the hard drive in our VirtualMachine. This will replicate the SD card that the Raspberry Pi uses for storage.

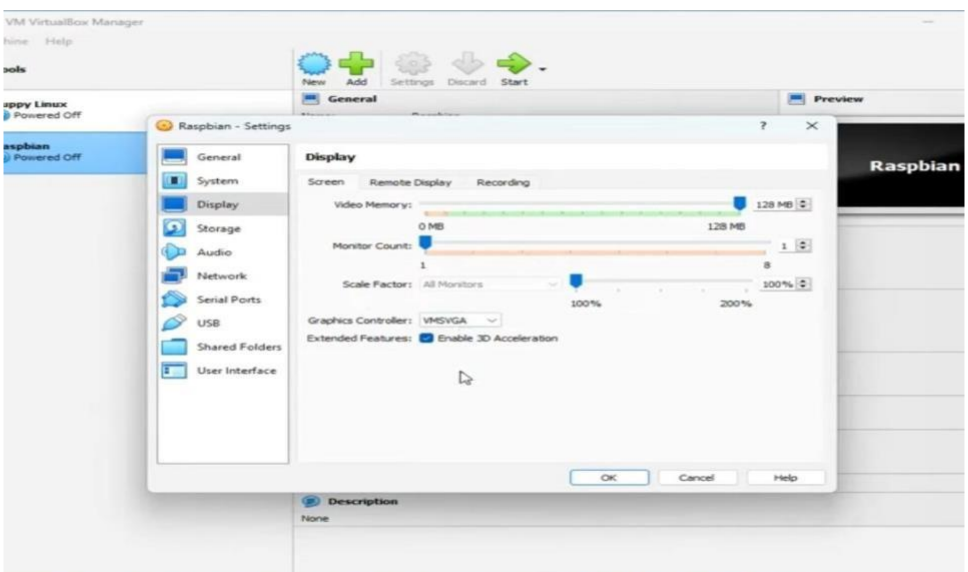



7.Click the "Finish" button to create your virtual machine. You'll see a new entry in the VirtualBox application, with the name you've just chosen. In our case, it was “rasbpeian”.

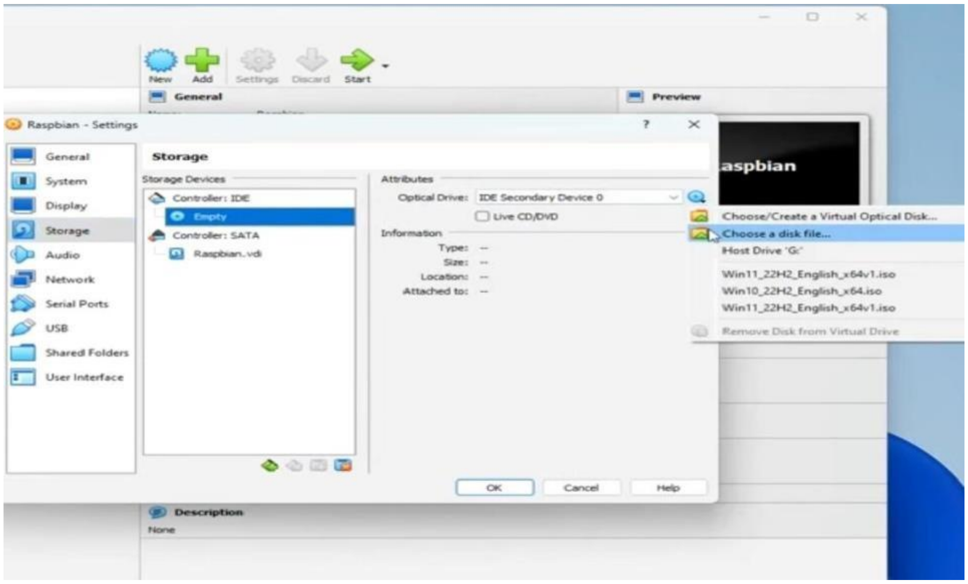




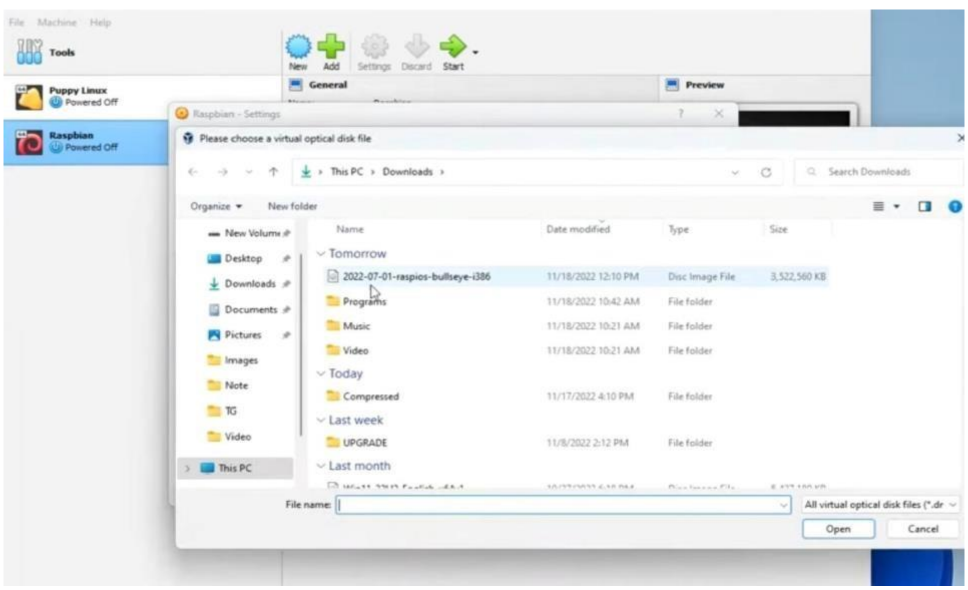

8.We're going to fine-tune some settings in our machine. Select your new virtual machine and click the "Settings" icon. Select "Display" in the sidebar, and drag the "Video Memory" slider all the way to the right.and then enable 3D acceleration



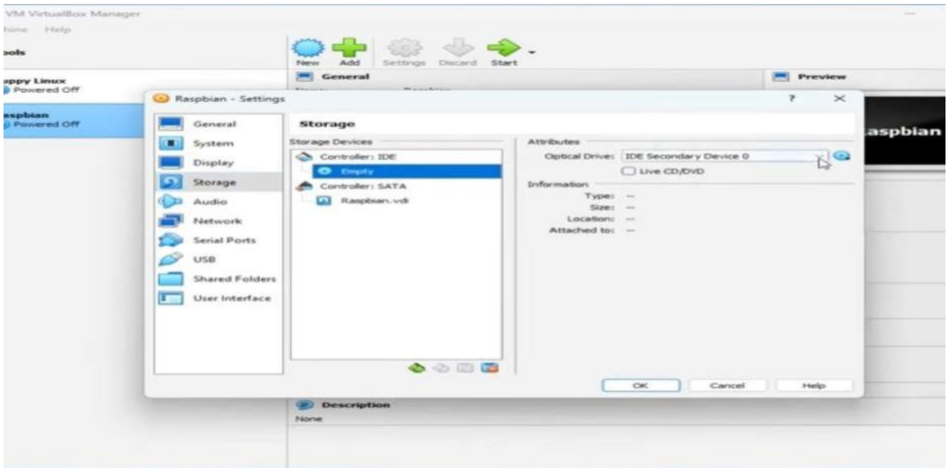

9.Now go to storage and click on empty on the controller. And click on disks icon and click on choose a disk file.



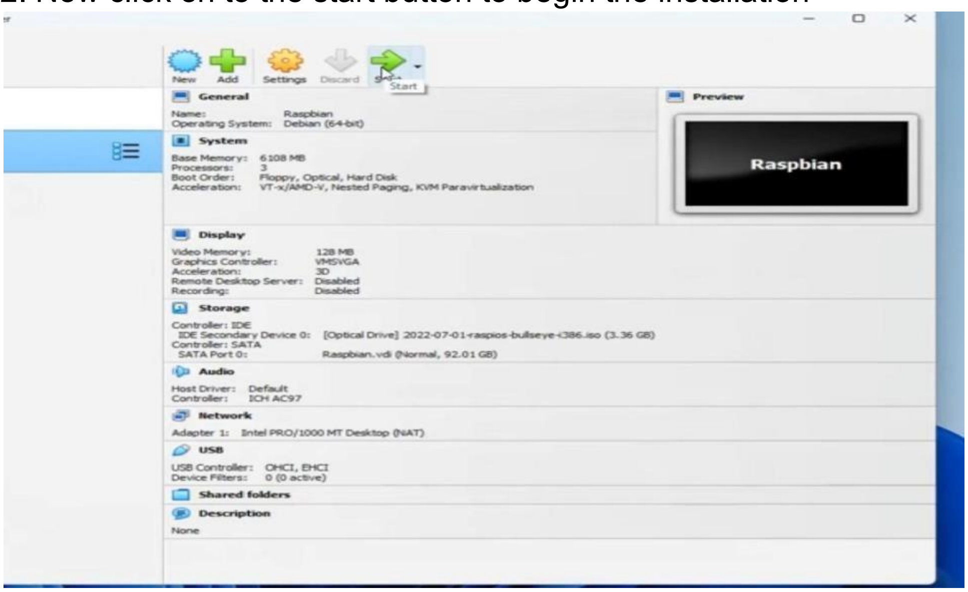

10.select the raspberry pi Os file. And click on open.



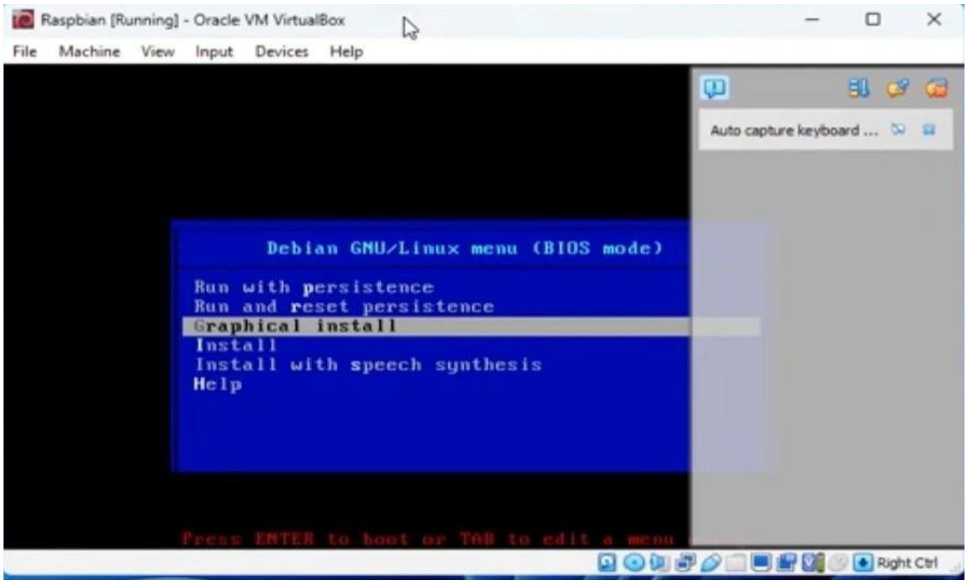
11.click on ok.



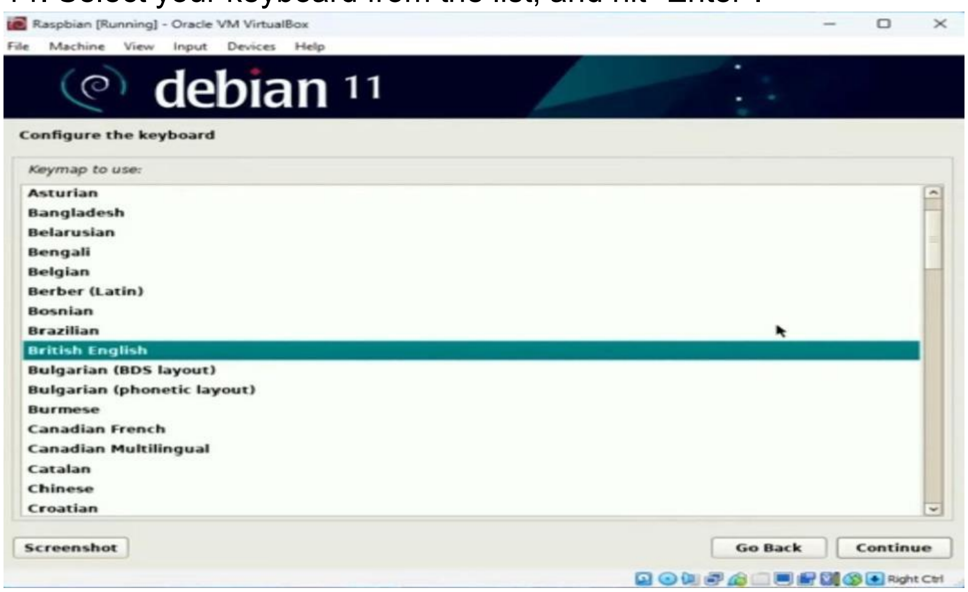

12. Now click on to the start button to begin the installation

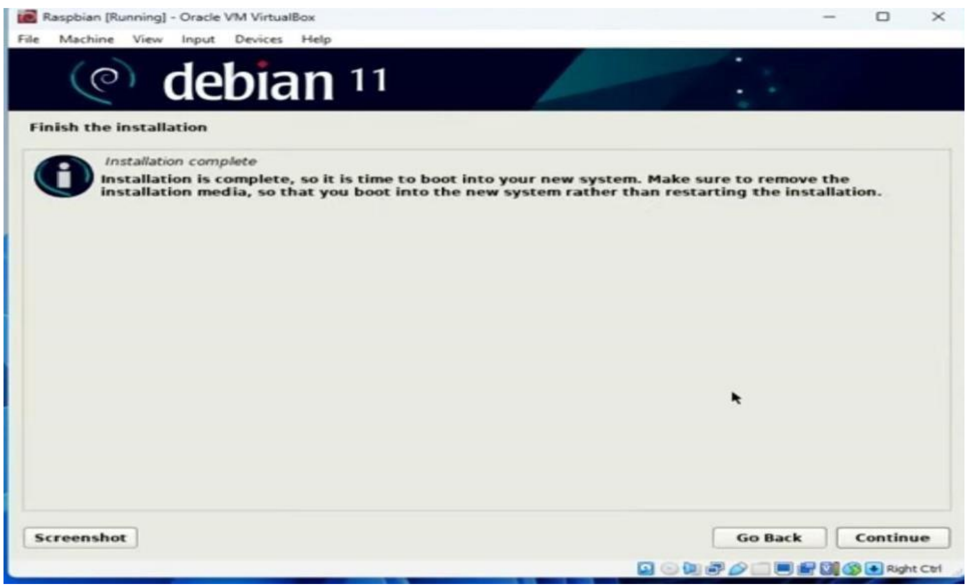



13.Select "Graphical Install" from the menu and hit "Enter."

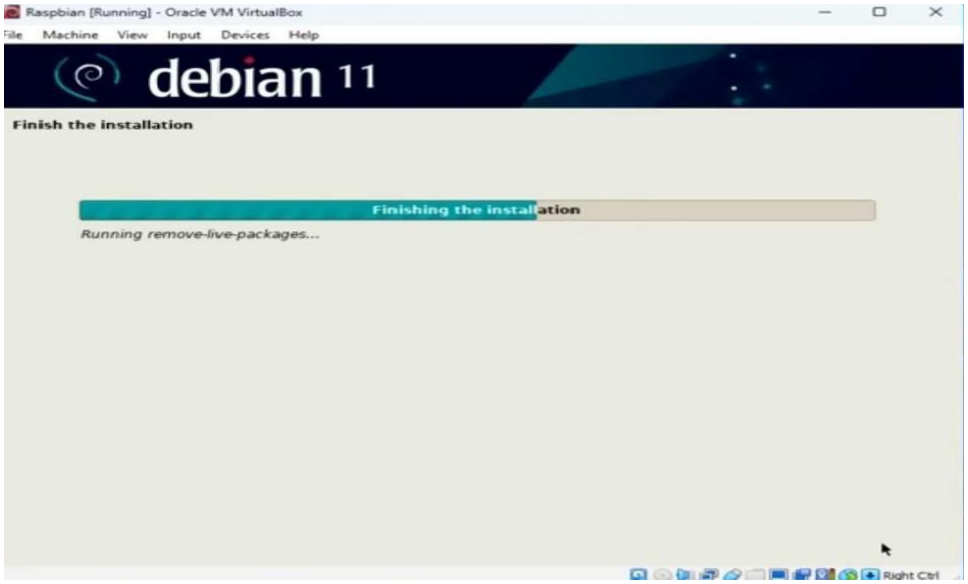



14. Select your keyboard from the list, and hit "Enter”.



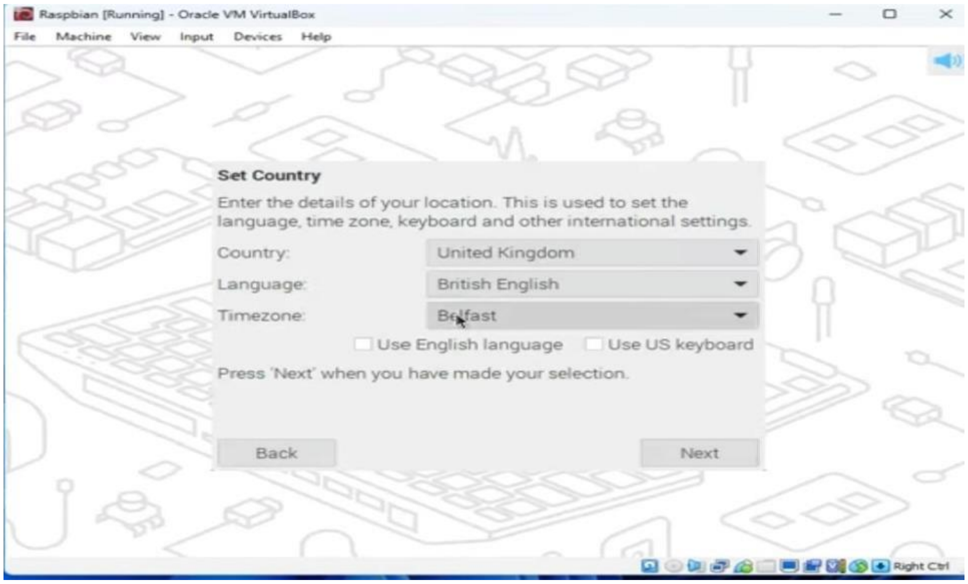




A progress bar will slowly creep along as the installation takes place.

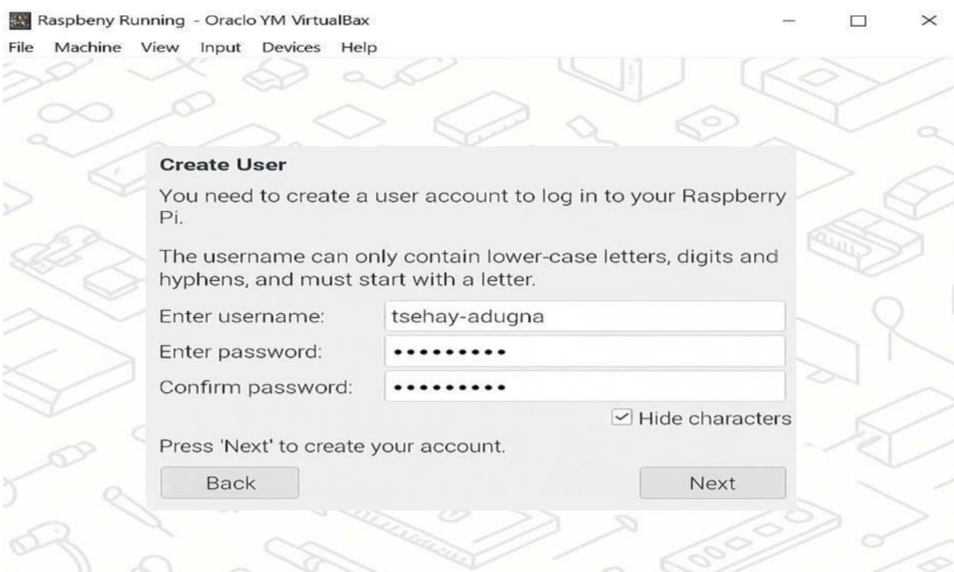



The first time you boot your virtual machine you'll have to provide some information to finalize
configuring your virtual Pi

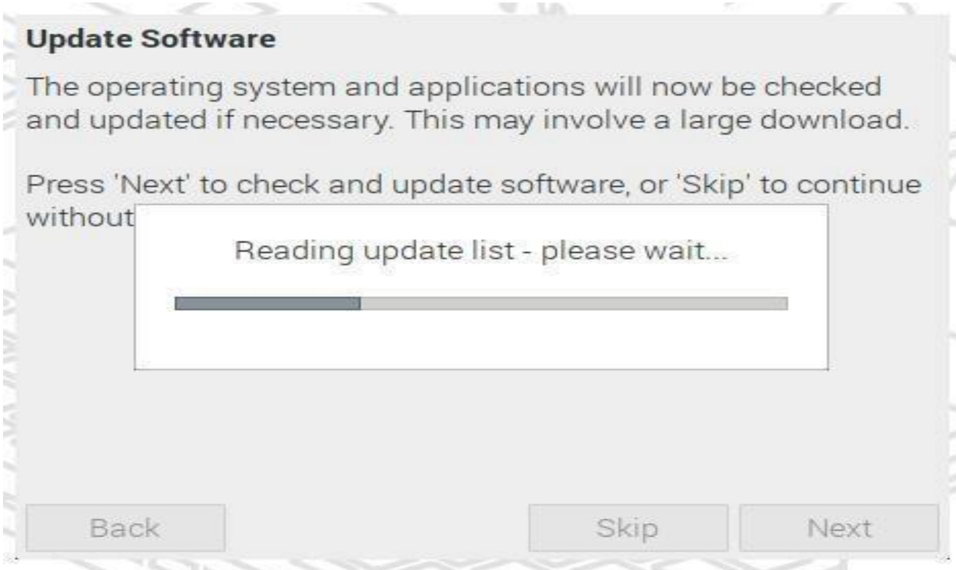

15.Click the "Next" button when you see the first setup screen. On the "Set Country" screen use the drop-down menus to select your country, language and time zone setting.



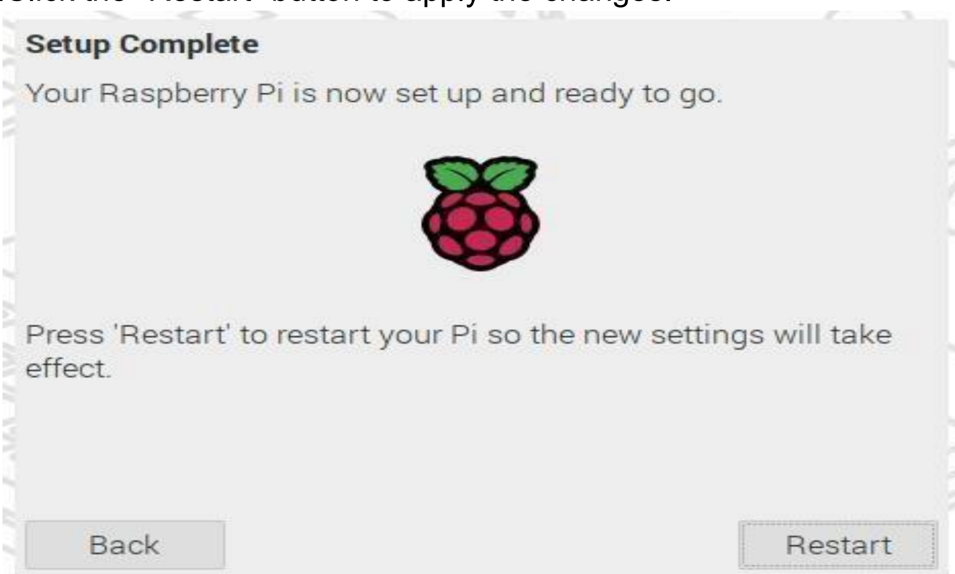

16.Click the "Next" button to proceed. On the "Create User" screen, enter the name of your user and create a password for them. Click the “Next" button to create their user account.



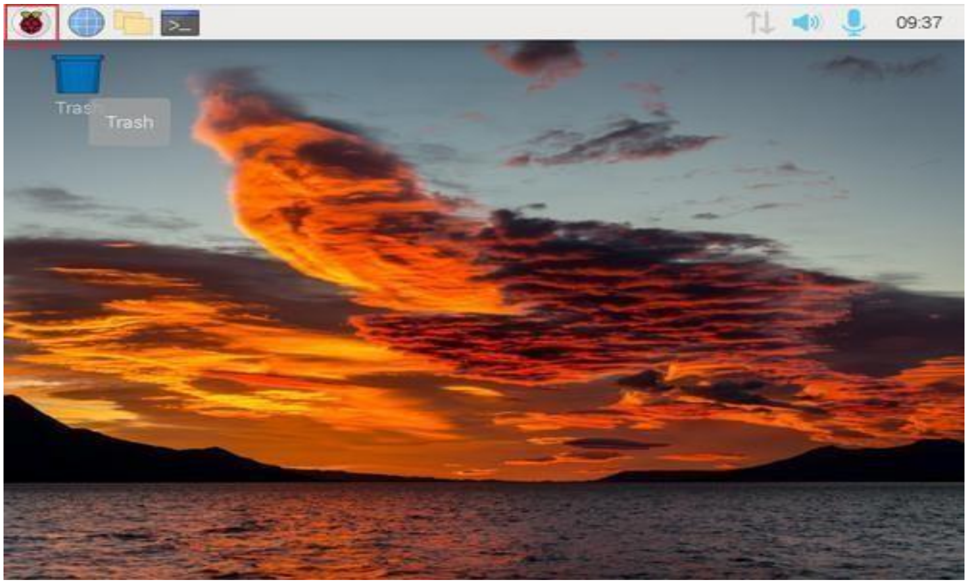

17.The "Update Software “screen checks your current install against the software repositories to make sure it is up to date. This takes a little while, but I recommend you bite the bullet and sit through it.

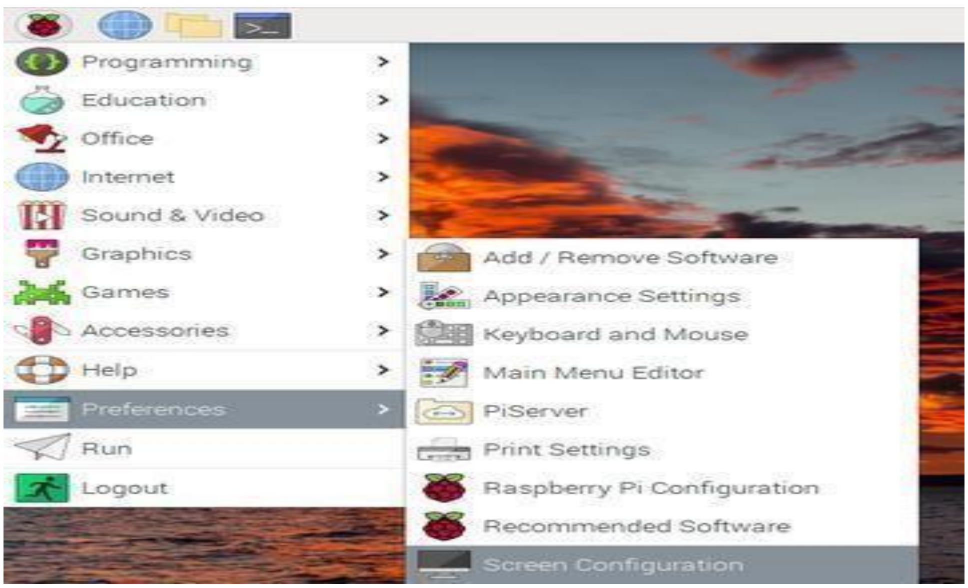



18.Click the "Restart" button to apply the changes.




19.Setting a more realistic resolution is easy. Click the Raspberry icon in the top left corner.


09:37
Trash


Programming
Education
Office
Internet
Sound & Video
Graphics
Games
Accessories
Help
Preferences
Run
Logout
Add / Remove Software
Appearance Settings
Keyboard and Mouse
Main Menu Editor
PiServer
Print Settings
Raspberry Pi Configuration
Recommended Software
Screen Configuration

## Issues(problem faced)

When setting up the Raspberry Pi operating system (typically Raspberry Pi OS), users may face several issues or problems, especially if they are new to Linux or working with hardware. Here are some common ones:

1. Booting Issues:

A frequent problem is the Raspberry Pi not booting up. This can be due to an improperly flashed microSD card, a corrupt OS image, or using a card that isn't formatted correctly. Incompatible or low-quality microSD cards can also cause the Pi to hang or fail to start.

2. Power Supply Problems:

Using a power supply that doesn’t provide enough voltage or current can result in unstable operation, random reboots, or boot failure. Some users may overlook this and try to use a phone charger, which might not deliver the needed 5V at 2.5A (or more, depending on the model).

3. Peripheral Compatibility:

Not all USB devices (like keyboards, Wi-Fi adapters, or external drives) are fully compatible outof-the-box with the Raspberry Pi. This can lead to input lag, device detection failures, or the OS not recognizing the hardware at all.

4. Display Configuration:

Sometimes, the HDMI display may not show output due to resolution mismatches or lack of HDMI signal detection. This might require manual changes to the config.txt file on the SD card to force HDMI output or adjust display settings.

5. Network Connectivity:

Initial setup may involve problems with connecting to Wi-Fi, especially if network credentials are entered incorrectly, or the Wi-Fi signal is weak. Some older models lack built-in Wi-Fi, requiring external adapters that need additional drivers.

6. Software Updates and Package Issues:

During or after setup, users might encounter errors while updating or installing packages due to broken repositories, missing dependencies, or internet issues. These errors can be confusing for beginners unfamiliar with Linux terminal commands.

7. User Permissions and Command Line Confusion:

Raspberry Pi OS is Linux-based, so users must deal with concepts like sudo privileges, file permissions, and command-line navigation, which can be intimidating for newcomers.

## Solutions

Here are practical solutions to the common problems encountered while setting up the Raspberry Pi operating system:

1. Booting Issues:

Ensure you download the correct and latest OS image from the official Raspberry Pi website.

Use a reliable flashing tool like Raspberry Pi Imager or balenaEtcher to write the image to the microSD card. Use a high-quality, Class 10 microSD card (8GB or larger) and format it with FAT32 before flashing. If it still doesn't boot, try a different microSD card or reflash the OS.

2. Power Supply Problems:

Always use an official Raspberry Pi power supply or one that delivers at least the required voltage and current (e.g., 5V/3A for Pi 4 and Pi 5). Avoid charging cables not designed for data and power; use proper USB-C or micro-USB cables.

3. Peripheral Compatibility:

Check the Raspberry Pi hardware compatibility list online before purchasing USB accessories. Use wired peripherals during setup to avoid wireless driver issues. Ensure any USB hubs are powered if you're connecting multiple devices.

4. Display Configuration

If the screen stays black, open the config.txt file on the SD card and uncomment or add:

hdmi_force_hotplug=1hdmi_group=1

hdmi_mode=16 (or suitable mode for your display).

Try a different HDMI cable or port ifissues persist.

5. Network Connectivity:

Double-check the Wi-Fi SSID and password before setup. Move the Raspberry Pi closer to the router or use an Ethernet cable for a more stable connection.For headless setups, add a file named wpa_supplicant.conf and an empty ssh file to the boot partition.

6. Software Updates and Package Issues:

Make sure your internet connection is stable. Run sudo apt update followed by sudo apt full-upgrade to ensure the system is fully up to date. If packages fail, check for typos, and use sudo apt --fix-broken install to resolve dependency issues.

7. User Permissions and Command Line Confusion:

Use sudo when installing or modifying system files. Learn basic Linux commands (like cd, ls, mkdir, nano) to navigate and edit files. Refer to the official Raspberry Pi documentation or forums for help. Following these solutions will ensure a smoother setup process and help you troubleshoot common issues confidently.

## File system support

The Raspberry Pi, particularly when running Raspberry Pi OS (a Linux-based system), supports many common filesystems, though the level of support and functionality can vary. Here's an overview of each:

- **ext4**: **Fully supported** and the default filesystem for Raspberry Pi OS. It's the most stable and optimized for performance on Linux.
- **FAT32**: **Fully supported**. Commonly used for boot partitions and USB drives. Great for compatibility with Windows and macOS, though it has a 4GB file size limit.
- **exFAT**: **Supported** with built-in support in recent Raspberry Pi OS versions. Suitable for flash drives and external hard drives needing to store files larger than 4GB.

- **NTFS**: **Supported**, but with limited write performance compared to Linux-native filesystems. Suitable for reading or occasionally writing to Windows-formatted drives.
- **Btrfs**: **Partially supported**. Can be used on the Pi but may require manual installation of tools (btrfs-progs). It offers advanced features like snapshots and compression but is more complex to manage.
- **ZFS**: **Not natively supported**. Requires third-party installation and kernel module compilation, which may not perform well on lower-end Raspberry Pi models due to ZFS's resource demands.

- **HFS+**: **Read-only support** by default for macOS-formatted drives. Read/write support is possible with extra configuration but not ideal for long-term use.
- **APFS**: **Not supported**. Apple's proprietary filesystem is not open-source, and Linux (including Raspberry Pi OS) has no reliable support for it.

In summary, the best-supported filesystems on Raspberry Pi are **ext4**, **FAT32**, **exFAT**, and **NTFS** (with some limitations). Others like **Btrfs** and **ZFS** can be used by advanced users, while **HFS+** and **APFS** are limited or unsupported.

## Why raspberry pi support this file systems

The reason for the varying levels of filesystem support on the Raspberry Pi primarily comes down to **compatibility with Linux**, **open-source availability**, and **hardware limitations**.

- **ext4** is the default because it is **natively supported by the Linux kernel**, stable, fast, and reliable. It's optimized for Linux systems like Raspberry Pi OS, offering excellent performance and journaling features that protect against data corruption—important for devices like the Raspberry Pi that may lose power unexpectedly.
- **FAT32** and **exFAT** are supported because they are **widely used for removable storage** and are well-documented. FAT32 is nearly universally compatible across operating systems but is outdated with limitations (like the 4GB file size cap). exFAT is newer and supports larger files, making it ideal for USB drives and SD cards used for media and backups. Raspberry Pi OS includes support for these filesystems to make external drive access simple and cross-platform.
- **NTFS** is supported because it's common in **Windows environments**, but it's not native to Linux. Linux uses a driver (like ntfs-3g) to read/write NTFS, but write performance and reliability are generally worse than with ext4. It's included for convenience rather than performance.

- **Btrfs** and **ZFS** are advanced filesystems designed for features like **snapshotting, compression, and error correction**, which are great on servers or enterprise systems. However, they are **resource-intensive**, making them less ideal for the Raspberry Pi's lower-end hardware (especially older models). They're also not included by default, requiring manual setup.
- **HFS+** and **APFS** are **Apple's proprietary filesystems**. HFS+ has some limited support in Linux (often read-only), while APFS is closed-source and **not supported** natively in Linux at all. Because Apple doesn't provide open specs or support, Linux developers can't easily implement full functionality.

In short, the Raspberry Pi favors filesystems that are **open-source, Linux-native, and resourceefficient**, while still offering some level of support for external drives commonly used in Windows and macOS environments for flexibility.

## Advantages of Raspberry pi

The Raspberry Pi offers several advantages that make it a popular choice for education, prototyping, and hobbyist projects. Its most significant strengths include its **affordability**, **compact size**, and **versatility**. It allows users to explore programming, electronics, and network applications without needing expensive hardware. With built-in support for GPIO pins, USB, HDMI, and networking, it can be used for a wide range of applications—from learning to code and building smart home systems to running lightweight servers and media centers. Additionally, it has a strong global community and extensive online resources, making it beginner-friendly and highly supported.

## Disadvantage of Raspberry pi

The Raspberry Pi also comes with some disadvantages. Its **limited processing power and memory**, especially in older models, can restrict its use for heavy computing tasks or professional-grade software. Storage is typically based on microSD cards, which are slower and less reliable than SSDs or hard drives. While it's great for learning and small-scale projects, the Raspberry Pi may not be suitable for performance-intensive applications like video editing, large databases, or high-traffic servers. Also, since it relies on Linux-based systems, users unfamiliar with Linux may face a learning curve when setting up and troubleshooting.

The Raspberry Pi is an excellent educational and prototyping tool, offering affordable computing with flexible usage. From beginner coding projects to complex IoT systems, it serves as a gateway into the world of electronics and software development.

## Future outlook and recommendations

The future outlook for the Raspberry Pi is promising, as it continues to evolve with more powerful hardware and broader use cases in education, industry, and DIY communities. With the release of newer models like the Raspberry Pi 5, which offer faster processors, better graphics support, and expanded memory options, the platform is becoming increasingly capable of handling complex tasks. It is expected to play a growing role in fields like **edge computing**, **IoT**, **robotics**, and **AI**, especially as demand for low-cost, energy-efficient devices rises. Additionally, its educational mission remains strong, with ongoing efforts to make computing accessible worldwide, especially in under-resourced areas.

As a recommendation, users and developers should continue to explore the Raspberry Pi not just as a learning tool, but as a **viable platform for prototyping and small-scale deployment**. Educators can leverage it to teach core concepts in computer science and electronics, while hobbyists and professionals can use it to build smart devices, automate tasks, or test software. For long-term success, it is advisable to stay engaged with the Raspberry Pi community, keep software up to date, and explore integrations with emerging technologies such as machine learning and cloud computing to unlock its full potential.

# Virtualization in Modern Operating Systems

## What is Virtualization?

Virtualization is the creation of a virtual (rather than physical) version of something—like an operating system, a server, storage device, or network resource. It allows multiple operating systems and applications to run on a single physical machine, using virtual machines (VMs).

Virtualization is a technology that allows you to create virtual, simulated environments from a single, physical machine. Through this process, IT professionals can make use out of their previous investments and optimize a physical nmachine's full capacity by distributing resources that are traditionally bound to hardware across many different environments.Used for decades, virtualization is a powerful technology within IT infrastructure that can be used to increase efficiency, retain flexibility, and improve scalability. Because multiple operating systems can share the same physical hardware, virtualization can improve resource use, reduce costs associated with physical maintenance, and boost security through isolated systems.

## Why virtualization?

Virtualization allows hardware systems to function at their highest capacity. With virtualization, multiple operating systems can run alongside each other and share the same virtualized hardware resources for optimized efficiency. Teams can make more use of their computing resources to support important applications and workloads. Some benefits of virtualization include:

Server consolidation: By virtualizing servers, many virtual servers can be placed on each physical server to improve hardware utilization. Server consolidation leads to improved resource utilization when resources are allocated to where they are needed because a host machine can be divided into multiple VMs. This approach takes full advantage of the hardware's capacity. You can maximize space, power consumption, and maintenance by hosting multiple VMs on a single piece of physical hardware.

Cost savings: Improved hardware utilization can mean savings on additional physical resources, like hard drives or hard disks, as well as reducing the need for power, space, and cooling in the datacenter.

Isolated environments: Because they're separated from the rest of a system, VMs won't interfere with what's running on the host hardware, and they are a good option for testing new applications or setting up a production environment.

Faster application migration: Administrators no longer have to wait for every application to be certified on new hardware. Because VM configurations are defined by software, VMs can be quickly created, removed, cloned, and migrated. You can control a VM remotely, and you can automate the management of VMs.

Efficient environments: During regression tests, teams can create or copy a test environment, eliminating the need for dedicated testing hardware or redundant development servers. With the right training and knowledge, teams can optimize environments to gain greater capabilities and density.

Disaster recovery: VMs provide additional disaster recovery options by enabling failover that could previously only be achieved through additional hardware. Disaster recovery options reduce the time it takes to repair and set up the impacted server, leading to greater adaptability.

## How does virtualization work?

Virtualization depends on 2 important concepts: virtual machines and hypervisors.

## Virtual machines

A virtual machine (VM) is a computing environment that functions as an isolated system with its own CPU, operating system (OS), memory, network interface, and storage, created from a pool of hardware resources. A VM can be defined by a single data file. As an isolated environment, it can be moved from 1 computer to another, opened in either, and be expected to work the same. Virtualization allows virtual machines with multiple different operating systems to run simultaneously on a single physical device- like running a MacOS or Windows environnment on a Linux° system. Each operating system runs in the same way an OS or application normally would on the host hardware, so the end user's experience is nearly identical to a real-time operating system experience running on a physical machine.

## Hypervisors

Sometimes called a virtual machine monitor (VMM), a hypervisor is software that separates a system's physical resources and divides those resources so that virtual environments can use them as needed.A hypervisor takes physical resources (such as CPU, memory, and storage) from the hardware and allocates them to multiple VMs at once, enabling the creation of new VMs and the management of existing ones. Hypervisors can sit on top of an operating system (like on a laptop) or be installed directly onto hardware (like a server). The physical hardware, when used as a hypervisor, is called the host, while the many VMs that use its resources are guests.When the virtual environment is running and user or program issues an instruction that requires additional resources from the physical environment, the hypervisor relays the request to the physical system and stores the changes in a cache-which all happens at close to native speed.

There are 2 different types of hypervisors that allow virtualization to happen based on need.

**Type 1:** Also referred to as a native or bare- metal hypervisor, it runs directly on the host's hardware to manage guest operating systems. It takes the place of a host operating system, and VM resources are scheduled directly to the hardware by the hypervisor. This type of hypervisor is most common in an enterprise datacenter or other server-based environments.

**Type 2**: Also known as a hosted hypervisor, it runs on a conventional operating system as a software layer or application. It works by abstracting guest operating systems from the host operating system. VM resources are scheduled against a host operating system, which is then executed against the hardware. This type is better for individual users who want to run multiple operating systems on a personal computer.

## References

1.Robu. (2021, October 12). 5 Pros and 5 Cons of Raspberry Pi. Robu.in. https://robu.in/5-pros-and-5-cons-of-raspberry-pi/

2.RaspberryTips. (n.d.). Raspberry Pi Pros and Cons: Is it Worth it?. RaspberryTips. https://raspberrytips.com/raspberry-pi-pros-and-cons/

3.Simplilearn. (2023, September 19). What Is Raspberry Pi: Types, Applications, and Advantages. https://www.simplilearn.com/tutorials/programmingtutorial/what-israspberry-pi

4.Red Hat. (n.d.). What is virtualization? Red Hat. https://www.redhat.com/en/topics/virtualization/what-is-virtualization