

# **BAHIR DAR UNIVERSITY**

## **BAHIR DAR INSTITUTE OF TECHNOLOGY FACULY OF COMPUTING**

---

**OPERATING SYSTEM AND SYSTEM PROGRAMING**

**INDIVIDUAL ASSIGNMNT**

**NAME-HELEN MISGANAW SOLOMON**

**ID NO-1601739**

**SECTION B**



To : Lecturer Wendmu Baye

## Introduction

The operating system I am going to install is PUPPY LINUX and the system call id chmod()-change the permission of the file.

An operating system is a software that manages computer hardware and software with a different important functions . for example

- Memory management
- resource management
- input /output management
- network enabling
- provide security

From many operating system one is puppy linux Os

Puppy Linux is a lightweight , fast ,and flexible Linux distribution designed to run efficiently in on older hardware and low-resource system.

### History of puppy Linux

#### A .Origins (2003-2005)

Puppy Linux was created by an Australian software developer **Barry Kauler** , in 2003. The earliest version were based on Vector Linux and Muppy Linux.

The main goal was to make a small , fast Linux distro that could run entirely in RAM and also allowing good performance on older computers.

#### B .Early development(2005-2008)

1. Puppy 1.0(2005): small and users JWM as it's default desktop.
2. Live CD/USB Focus : puppy was designed to run as a live system without installation , saving user data in a persistent file.

3. Community growth : A dedicated forum became the hub for puppy development and user support.

### **C. Puppy Linux variants(2008-2013)**

1. Barry Kauler's Retirement(2013) :
2. Forking and pups : different teams create their own versions .

### **D. Modern Puppy Linux(2014-present)**

1. puppy 6.X : introduced Ubuntu Xenial-based builds.
2. puppy 7. X : shifted to Debian .
3. Puppy 8.X : continued Ubuntu-based version.

## **Background**

puppy Linux is a lightweight ,open-source Linux distribution designed for speed ,simplicity ,and compatibility with older hardware.

Unlike traditional operating systems ,PUPPY Linux runs entirely in RAM(random access memory) , making it ideal for virtualization experiments.

## **Motivation**

Learn operating system in virtualized environment .

Explore puppy Linux's unique features .

Practice system administration tasks like file permissions (chmod).

## Objective

- 1 .Install puppy Linux.
2. Document the installation process with screenshots.
3. Analyze puppy Linux's file system support .
4. Troubleshoot common issues during installation.
5. Evalute advantage and disadvantage of puppy Linux.

## Requirements

### Hardware

#### Host machine

**CPU:** minimum x86 or AMD64 processor .

Recommended : 64-bit dual-core or better.

**RAM:** minimum 2GB for the host and virtualbox.

Recommended : 4GB+ if running other apps alongside virtualbox.

**Storage:** minimum 10GB free space for both puppy linux and virtualbox.

Recommended: 20GB or more for additional software.

#### **Virtual machine**

**RAM allocation:** minimum 256MB because puppy linux can run with very little RAM.

Recommended: 512MB-1GB for better performance.

**Storage allocation:** minimum 5GB virtual disk .

Recommended: 10GB virtual disk for flexibility and persistence.

## **software**

### **Host OS**

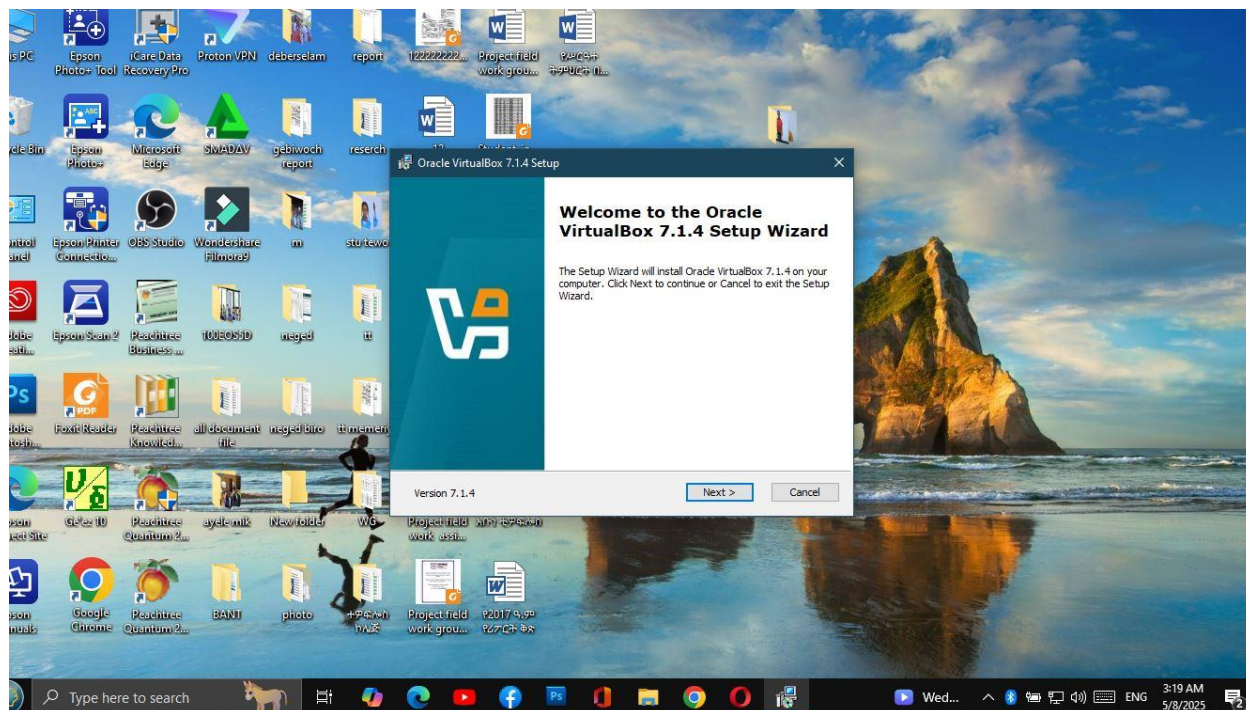
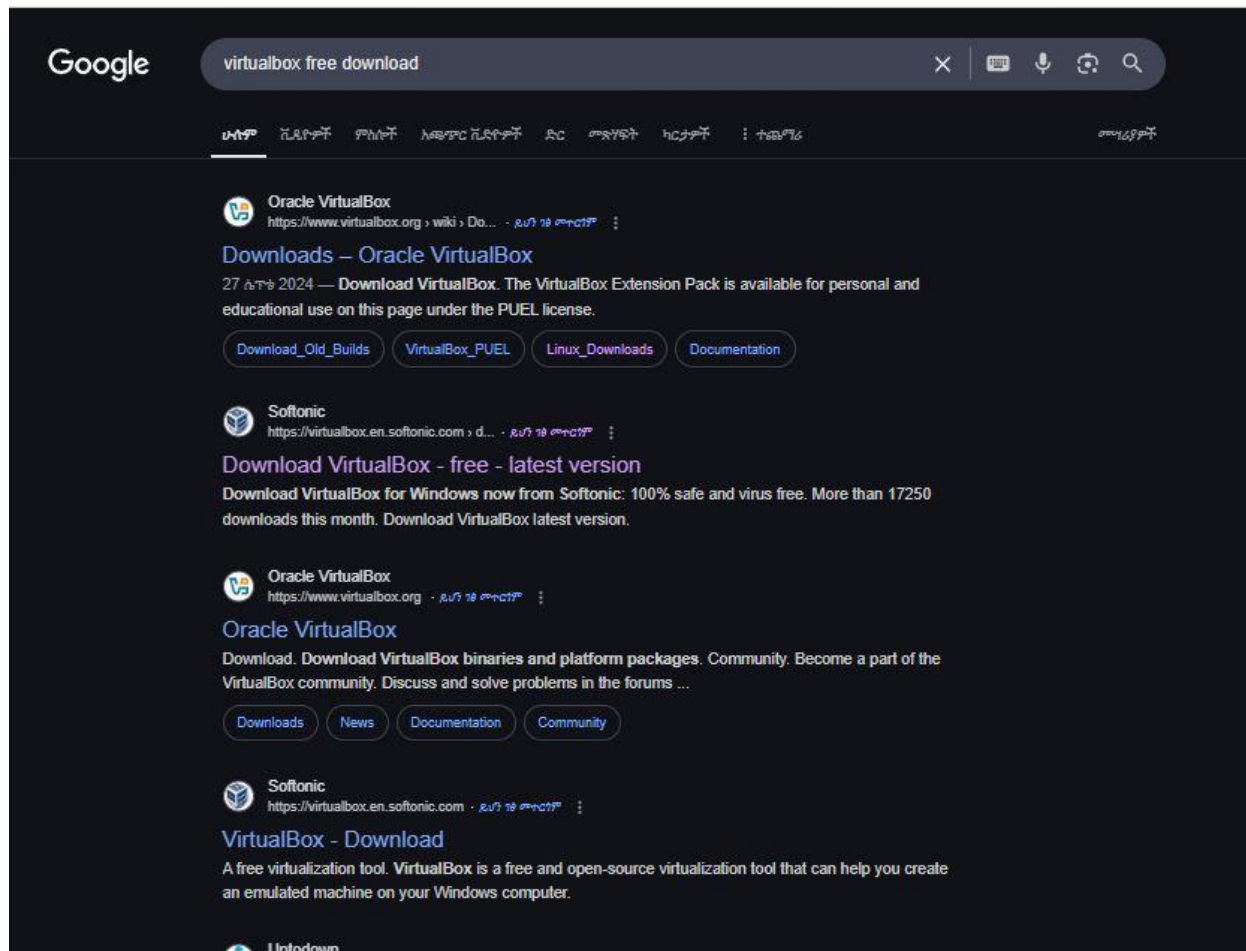
Supported operating system: windows , macOS ,Linux....

Virtualbox Version: latest stable release.

Puppy Linux ISO: ISO size typically 300MB-700MB very small download.

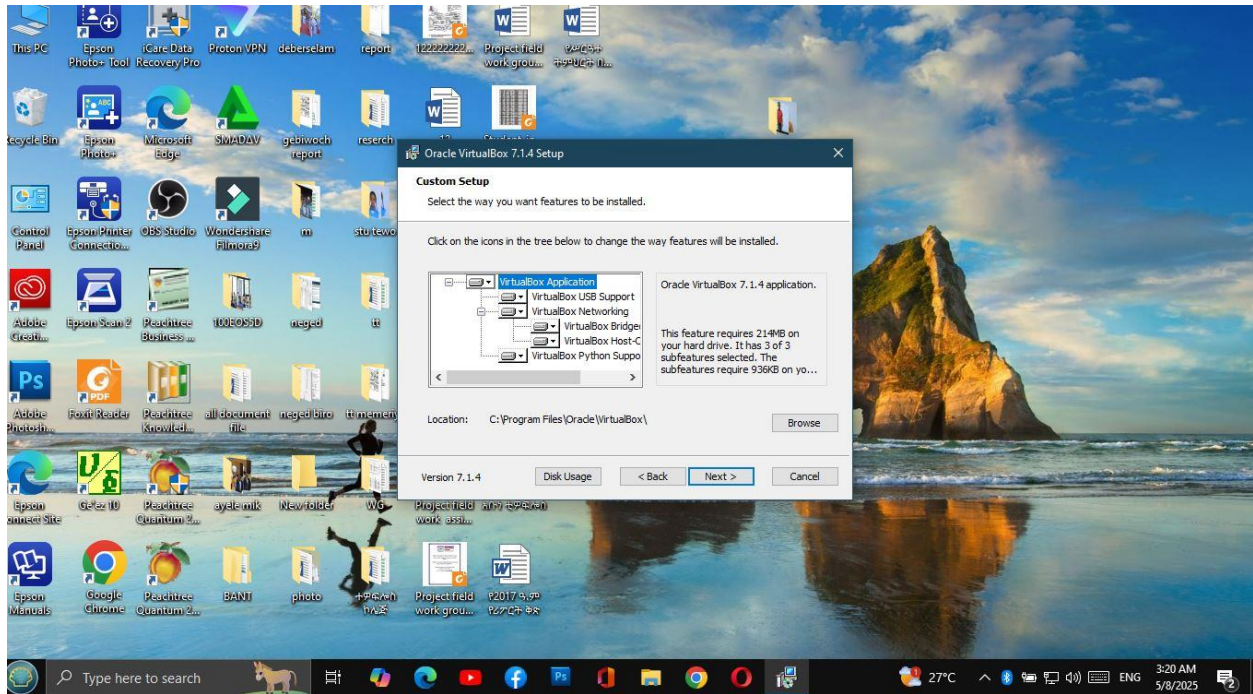
## **Installation steps**

Step 1: first go to our chrome and search virtual box to install virtual box and go to the home page and install the version you want and choose next .

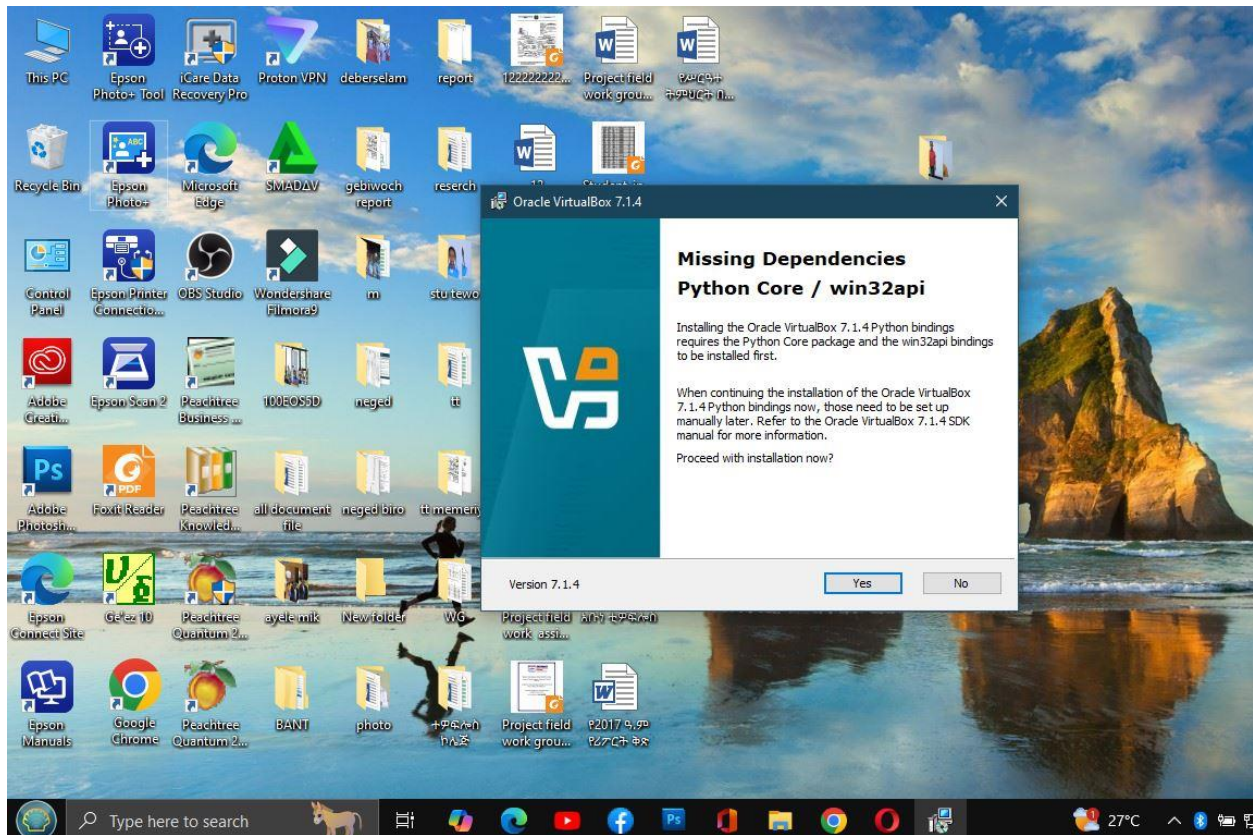




Step 2: we don't have to change anything so choose next.

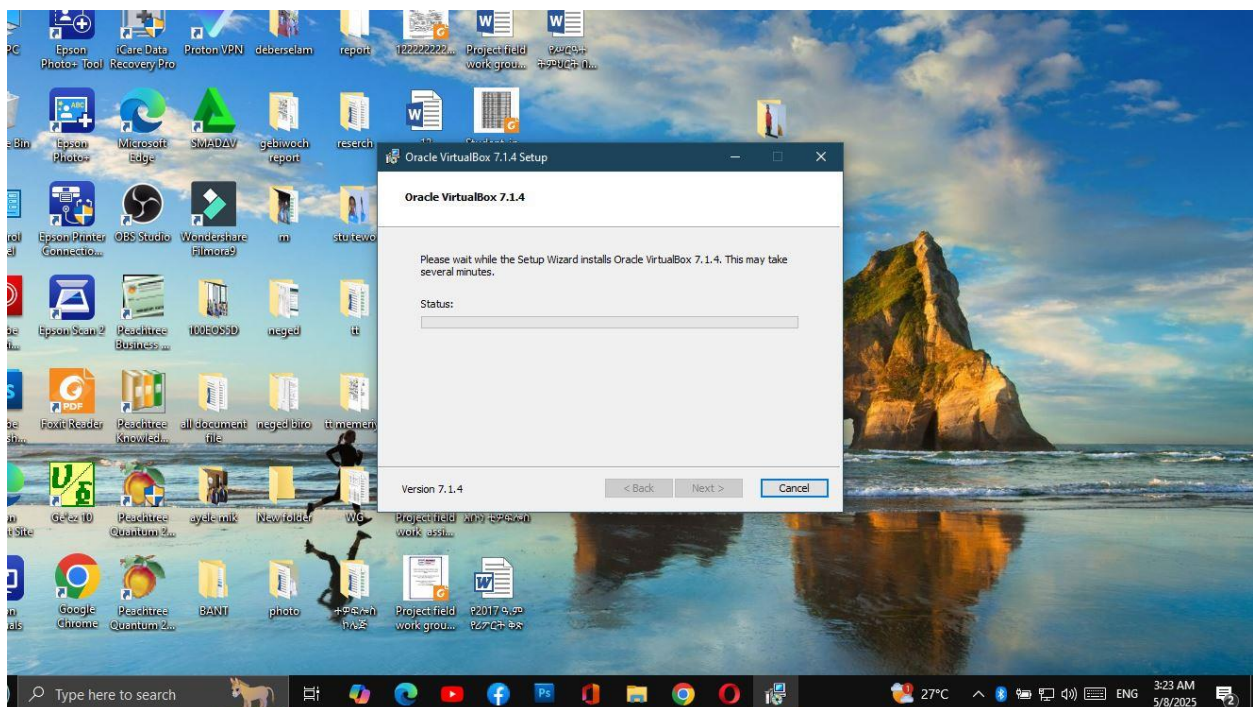
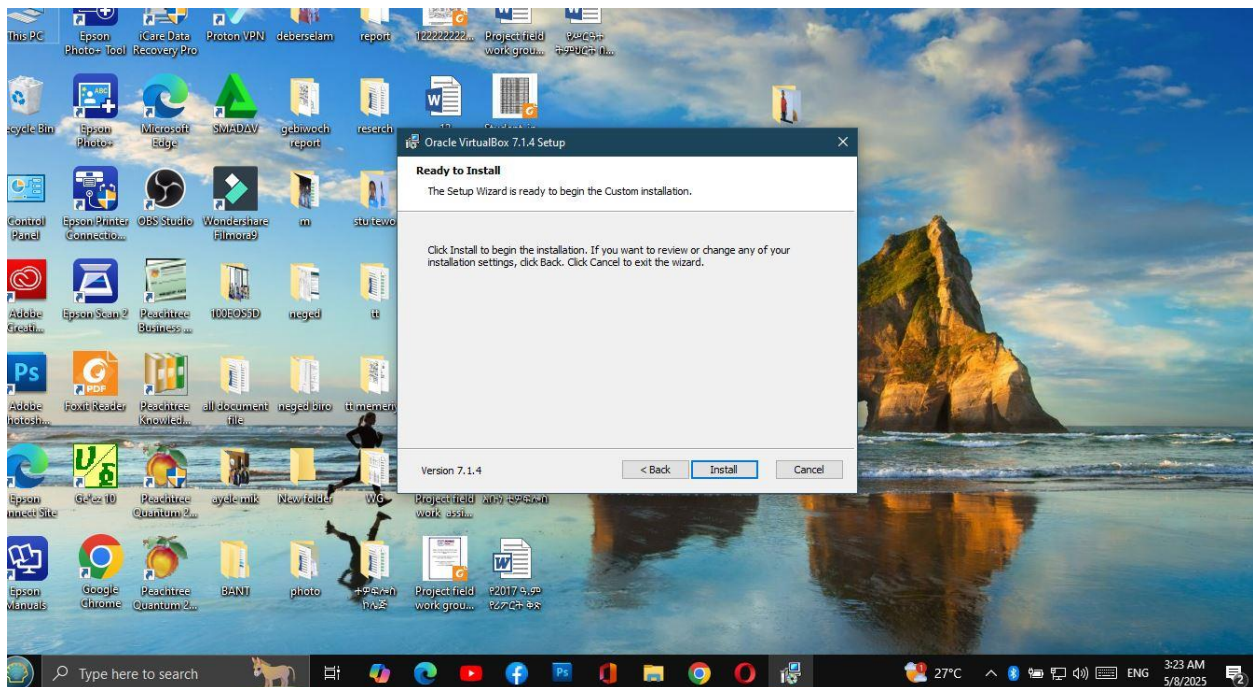


Step 3: the same process we will select yes now .

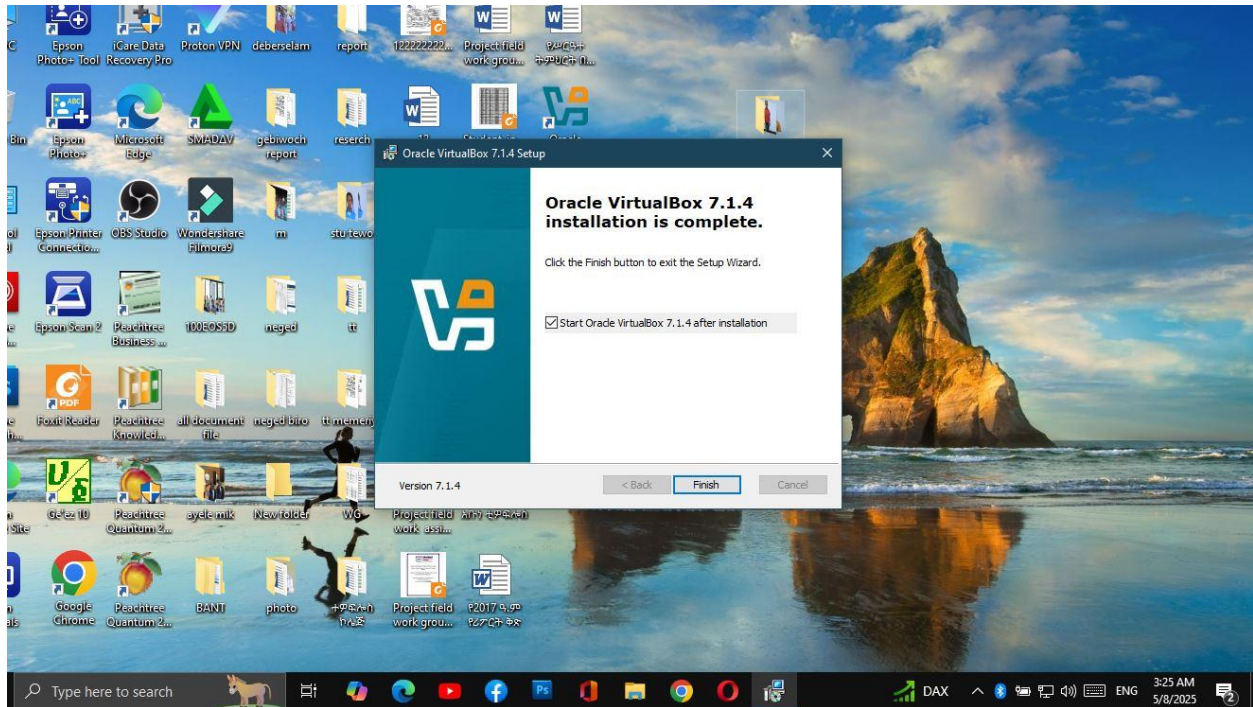


Step 4: now, we have done these steps so we will click install to begin the installation process. if we want to make any changes we can click back otherwise click install.

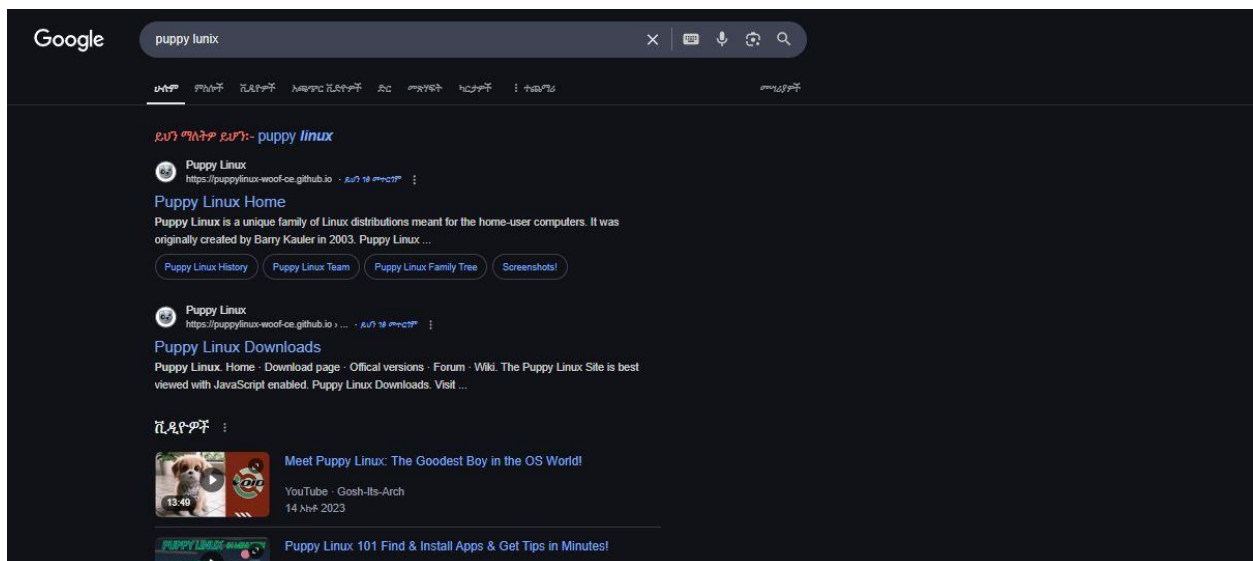




Step 5: now we finished installing so click finish and go to the next step.



Step 6: now that we install the virtual box the next will be installing ISO file of puppy linux by going in puppy linux official website



Step 7 :click the home page then we get the website for puppy linux



Step 8.chooses a version of ISO file of puppy Lunix.

Get the ISO or burn it to a CD/DVD using your favorite CD/DVD burner, or flash it using dd (windows version) to your USB flash drive, or visit our download page for more comprehensive information.

Compatibility *	Architecture	Latest Version	Download link
Ubuntu Focal 64	x86_64 64-bit	FossaPup64 9.5	<a href="#">Main - Mirror - Checksum</a>
Raspbian Buster	armhf 32-bit	Raspup 8.2.1	<a href="#">Main - Mirror - Checksum</a>
Ubuntu Bionic	x86 32-bit	BionicPup32 8.0	<a href="#">Main - Mirror - Checksum</a>
Ubuntu Bionic 64	x86_64 64-bit	BionicPup64 8.0	<a href="#">Main - Mirror - Checksum</a>
Ubuntu Xenial	x86 32-bit	XenialPup 7.5	<a href="#">Main - Mirror - Checksum</a>
Ubuntu Xenial 64	x86_64 64-bit	XenialPup64 7.5	<a href="#">Main - Mirror - Checksum</a>
Slackware 14.1	x86 32-bit	Slacko Puppy 6.3.2	<a href="#">Main - Mirror - Checksum</a>
Slackware64 14.1	x86_64 64-bit	Slacko64 Puppy 6.3.2	<a href="#">Main - Mirror - Checksum</a>
Ubuntu Trusty	x86 32-bit	Tahrpup 6.0.5	<a href="#">Main - Mirror - Checksum</a>
Ubuntu Trusty 64	x86_64 64-bit	Tahrpup64 6.0.5	<a href="#">Main - Mirror - Checksum</a>

\* Compatibility: A Puppylinux distribution can also be built and assembled using packages and components from another Linux distribution called in Puppy the "binary compatible" distribution. The choice of a binary compatible distribution determines the availability of additional packages, among other things.

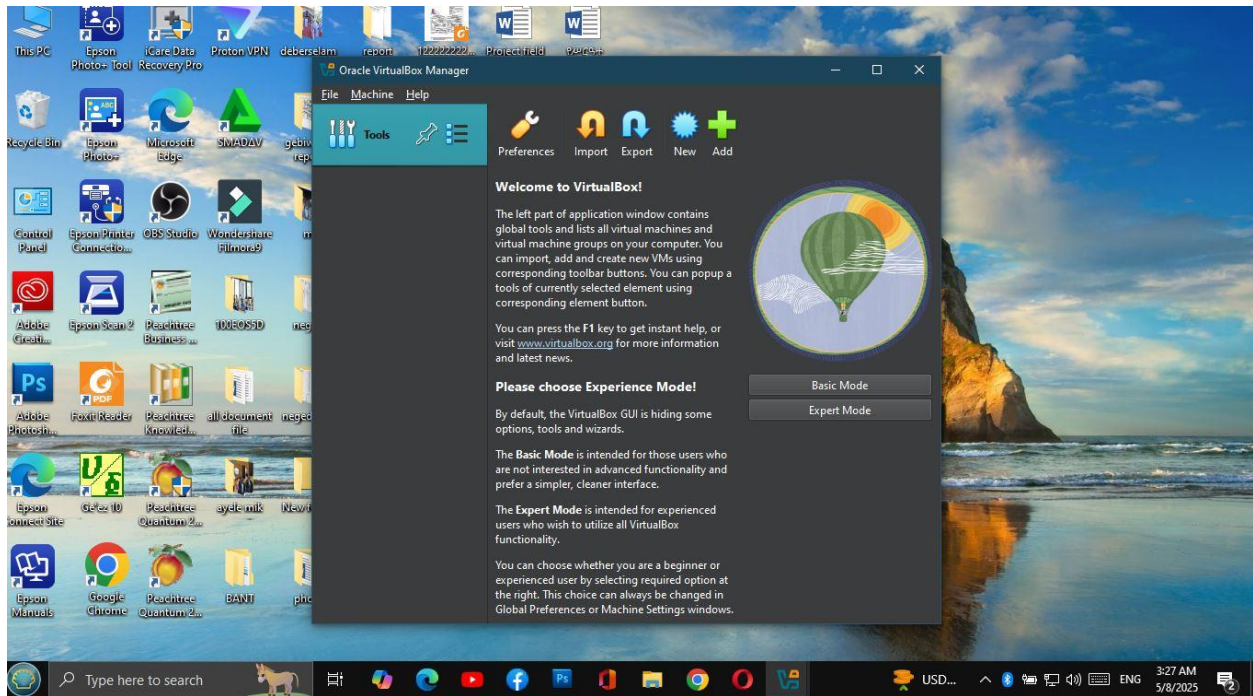
### Questions?

It has been said that the best experience of Puppy Linux is not from the software itself, but from the community that gathers around it. Whatever you have in mind - praises, curses, questions, suggestions, or just plain chit-chat, we welcome you to join us at [Puppy Linux Discussion Forum](#) or just check the [discussions](#) for the latest puppylinux woof builds.

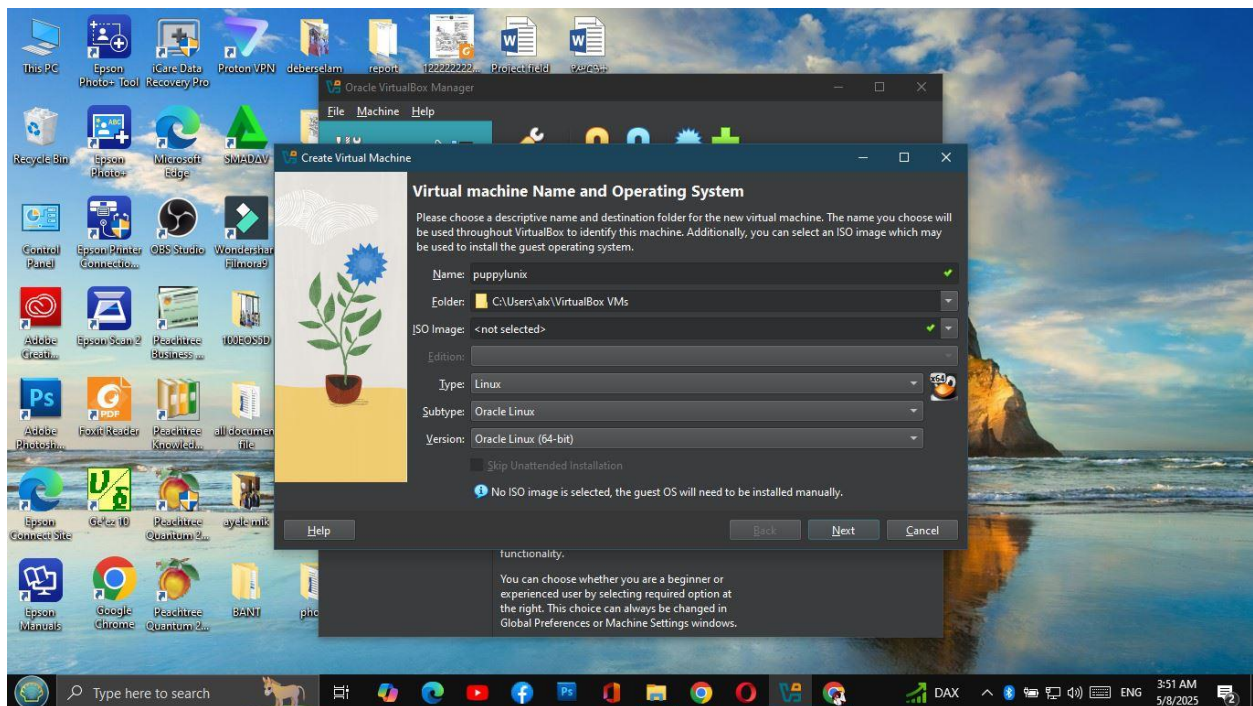
[I need more info before deciding to try ...](#)

Step 9:now , we have downloaded the ISO file so we go back to virtualbox ad create a new virtual machine .

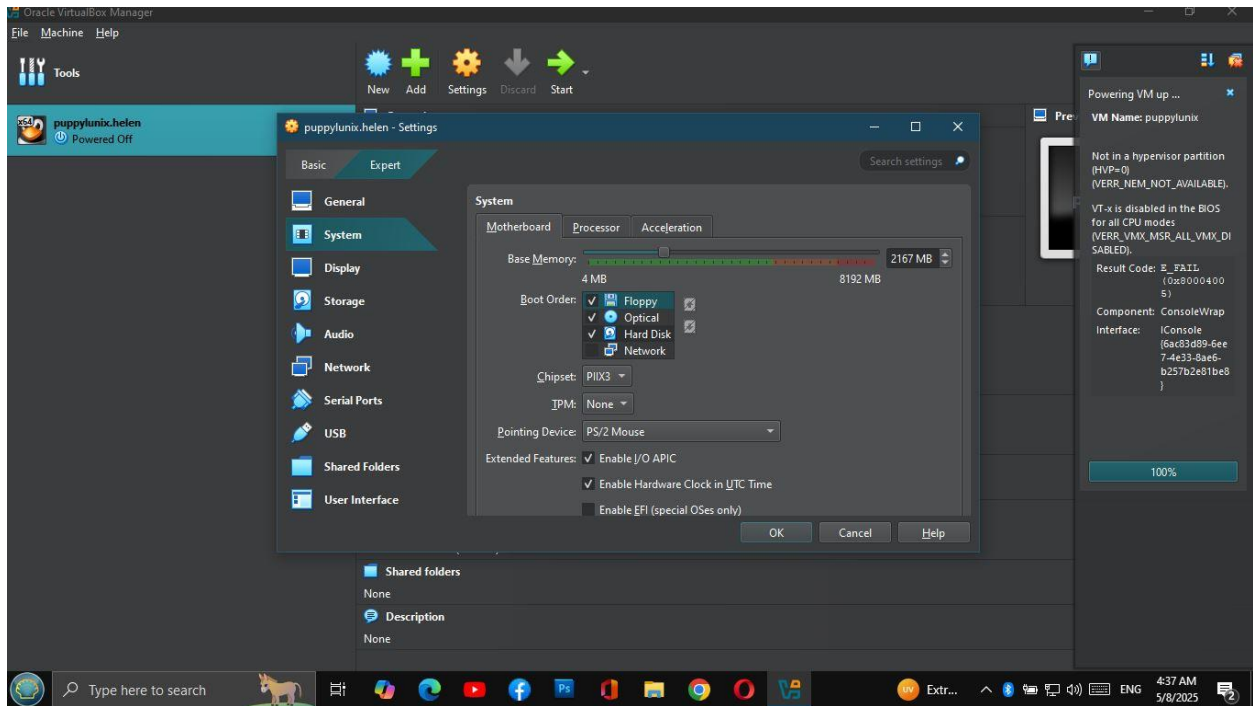




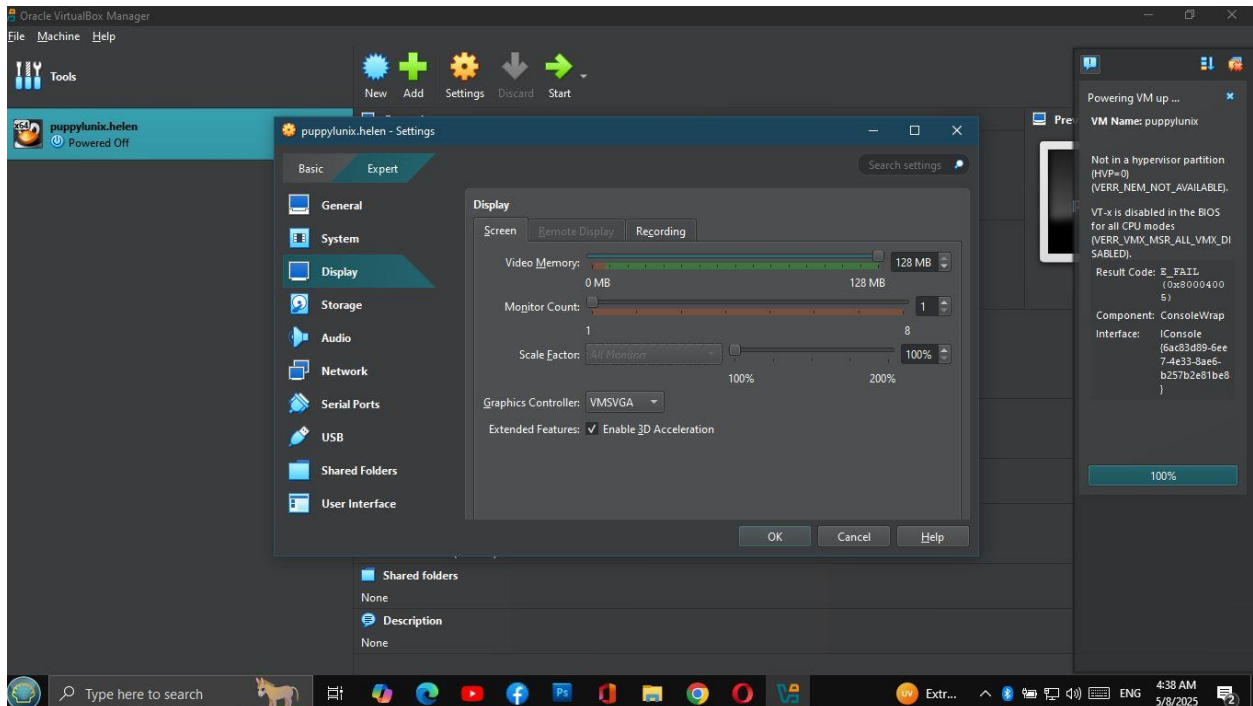
Step 10: now let's name our OS puppy linux attach the ISO file, type, version and click next.



Step 11: go to the system and change the base memory we shouldn't allocate more than 40 percent of our RAM of our PC because it will slow down the process.

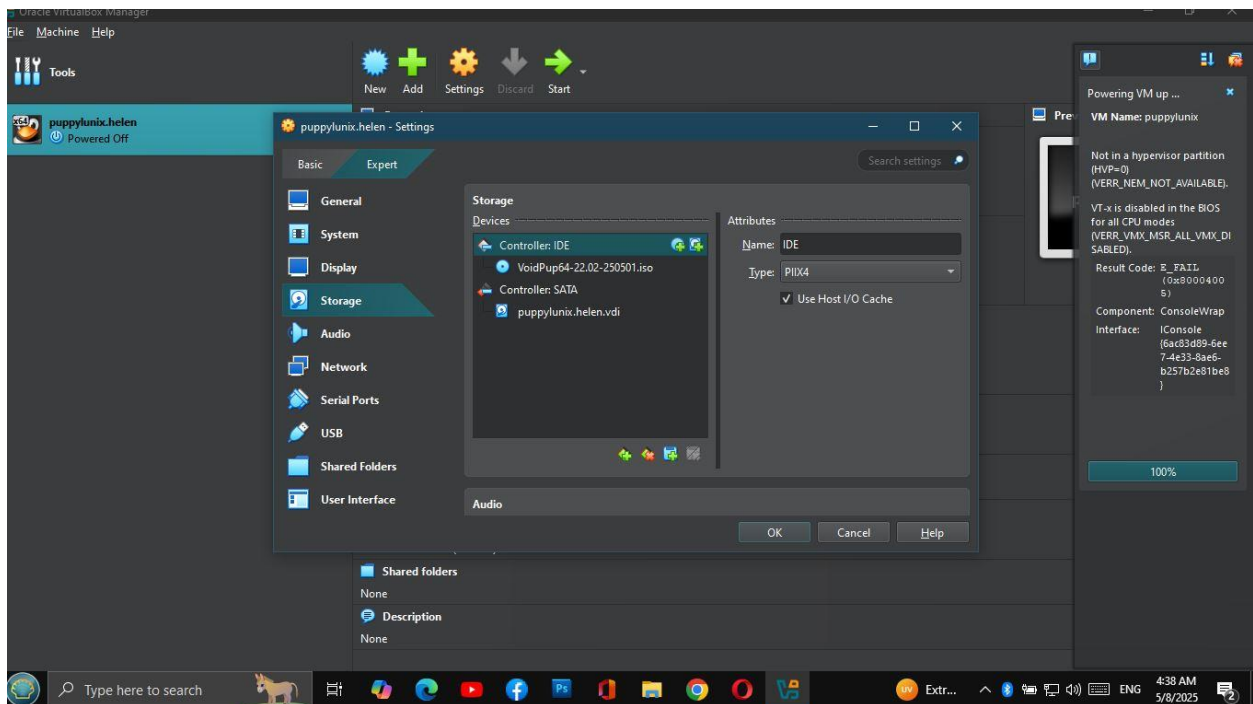


Step 12: go to display and make the video memory 128MB.

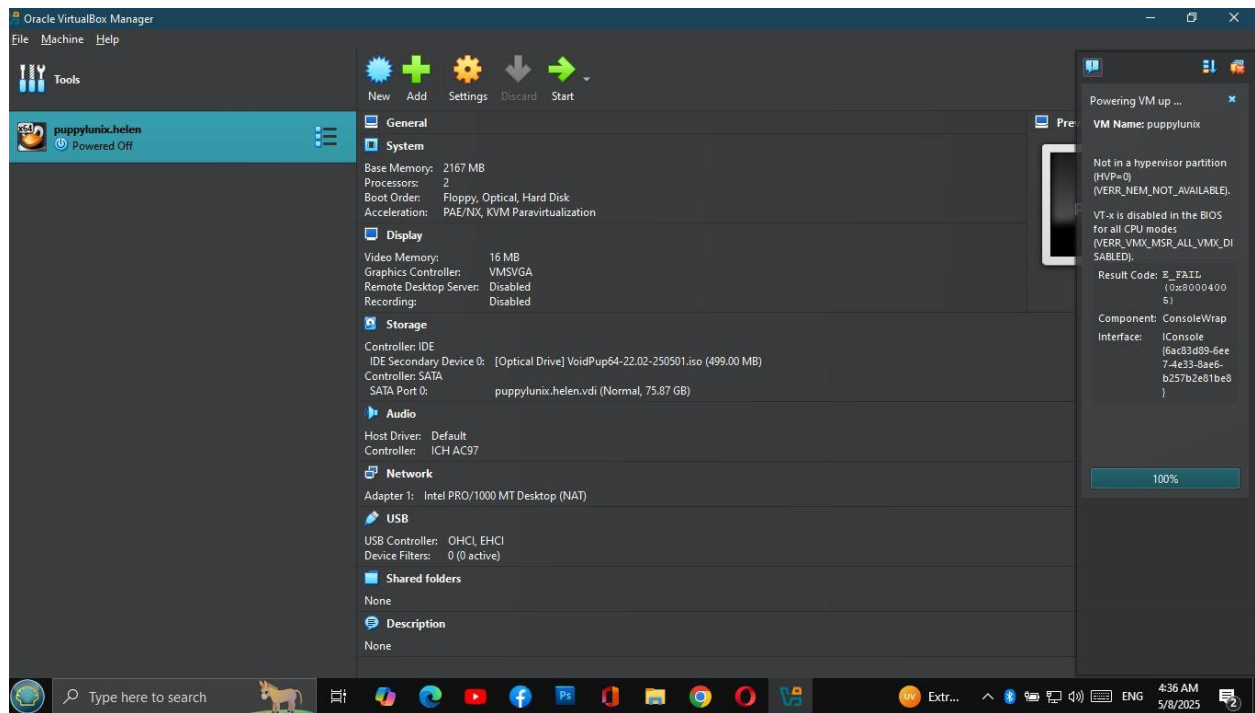




Step 13: now go to storage and click on controller IDE and insert the ISO file and click ok.



Step 14: finally ,we created our virtual machine named Puppylunix.helen.



Step 15: the last step is click start and start our machine and if we have any modifications we can make it ,once we have done installing puppy linux we can install the VirtualBox Guest additions for better integration with host and guest operating systems.

## Issues faced

The main problem I have been facing with was network (internet connection) the whole time .

## solution

The solution I take was using different network options ,changing the source network, places and also hardware.

## **File system support**

### **1.supported file system for installation and storage**

- a. **Primary file systems** :these are fully supported and work best for puppy linux installations.

#### **i.ext2(second extended file system)**

pros : it is simple ,fast , and lightweight ,works well got frugal installs and USB drives.

Cons : No journaling ,limited file size.

#### **ii.ext3(third extended file system)**

pros : adds journaling ,backward-compatible with ext2.

Cons : slightly slower than ext2 due to journaling .

#### **iii.ext4(fourth extended file system)**

pros : supports larger files and volumes, better performance with delayed allocation .

cons : overkill for very old hardware.

#### **iv.FAT32(file allocation table 32)**

pros /; universally compatible , works well for USB drives and shared storage.

Cons : maximum file size , no permissions or journaling.

#### **v. NTFS(new technology file system)**

**pros** : read/write support , works well for dual-boot with windows.

**Cons** : slower than ext4 on linux, not ideal for puppy's frugal install.

### **2.Secondary (limited support) file systems**

#### **vi. Btrfs(B-tree file system)**

pros : snapshots , compressions , and advanced features.

Cons : not all puppy versions support it out of the box, requires manual setup.

#### **vii. XFS(high-performance FS)**

pros : great for large files and high speed storage.

Cons : not ideal for small partitions.

### **viii.ReiserFS (legacy journaling FS)**

pros : good for handling many small files.

Cons : out dated , limited puppy support.

### **Viii.exFAT(extended FAT)**

Pros : no 4GB file limit.

Cons : requires exfat-fuse

## **advantage**

puppy linux is

**1.Externly lightweight** : is designed to runs smoothly on very old PCs and small low space hardware . due to it's small size it boots quickly .

**2. Fast performance:** Loads entirely into RAM ,making it blazing fast even in slow hardware.

**3. portable and can run without installation:** Can boot from USB ,CD ,or even SD card.

Doesn't need a hard drive (runs in live mode). It support's Frugal installs which means it doesn't require a full hard drive installation.

**4.user friendly for beginners**

Simple interface

Comes with essential apps pre-installed

**5.customizable and Modular** :\_users can easily add or remove software using SFS file or PET package.

**6.Good for security and privacy** : since it can run without a hard drive it leaves no traces on the host machine.

## Disadvantage

1. **Not ideal for modern or high-end systems** : while great for old PCs ,puppy Linux doesn't take full advantage of modern hardware.
2. **Limited software availability** : it uses it's PET manager and SFS files this means that not all standard Linux software. some applications require manual installation or may run slower.
3. **Persistence can be tricky** :it doesn't save changes between reboots unless manually configured. Users must create a save file to retain settings ,which can be confusing for some ,
4. **No official corporate backing** : unlike Ubuntu or Fedora ,puppy Linux is community-driven .this means slower update for security patches and fewer official hardware compatibility guarentees.
5. **Limited Multitasking and Heavy workloads** : due to lightweight nature it struggles with heavy multitasking . Not suitable for servers or professionals workstation tasks.

## Conclusion

Puppy Linux is a unique and highly efficient Linux distribution designed for speed, portability ,and minimal hardware requirement .one of the best choice for



reviving old computers and also have different benefits such as –portability and flexibility ,privacy and security etc.

It is a remarkable distribution that excels in specific scenarios particularly reviving old hardware , portable computing ,and system repairs. However ,it's limitations in software availability ,modern hardware support is not a general purpose for most users.

## What, why, and how virtualization in modern operating system

### What is virtualization

A virtualization is the process of creating a software based abstraction layer that decouples physical hardware resources to run multiple isolated operating systems in a single physical machine .it allows a single physical computer to run multiple operating systems or virtual machines simultaneously. This is achieved through a software layer called a hypervisor , which manages the allocation of hardware resource to each VM.

### Key Types

1. **Full virtualization** : the guest OS runs unmodified . it allows multiple guests operating system to run currently on a single physical machine without modification ,utilize a hypervisor to simulate the underlying hardware.  
Example : running window 10 inside a VM on a Linux host ,virtual box....

### Key concepts:

**a. Hypervisor** : a software layer that manages virtual machines and there resources . in full virtualization , hypervisor fully emulates the hardware.

**b. Guest operating system:** operating system that run within a virtual machine , unaware of virtualization .

**c. Hardware Emulation :** The hypervisor simulates the physical hardware , including COU , memory , and peripherals to provide a virtual environment for the guest OS.

**2. para-virtualization :** in modern operating system para virtualization is a virtualization technique where the guest operating system is aware of it's virtualized environment and directly communicate with the hypervisor through special cell **hypercalls** .

The direct interaction improves performance and efficiency compared to full virtualization which relies on emulating hardware instructions.

#### **Key aspects**

- a. Hypercall :** instead of relying on hardware emulation ,the guest OS uses hypercall to request service from hypervisor .
- b. improved performance :** By bypassing hardware emulation for certain operation , paravirtualization can reduce overhead and improve performance .
- c. modified OS :**it requires modifications to the guest OS kernel , such as replacing certain instruction with hypercalls.

**3.Hardware-Assisted Virtualization :** also known as native or accelerated virtualization ,uses COU level support to performance of running multiple virtual machines on a single physical machine .

Uses CPU extensions for better efficiency . example : modern hypervisors like KVM ,Hyper-V.

**4.containerization:** a form of virtualization that packages applications and their dependency in to a self-contained unit called **container** .it is a lightweight virtualization sharing the host kernel .

## Why virtualization

Virtualization is a fundamental technology in modern operating system because it enables efficient resource utilization , flexibility , isolation , and scalability .

### Some reasons why virtualization is essential

- a. **Efficient hardware utilization** : virtualization allows multiple virtual machines or containers to share the same hardware instead of dedicating one physical machine to a single OS.
- b. **Isolation and security** : each container operates independently , preventing crashes in one from affecting others this means that crashes in one container will not affect the others.
- c. **Cost saving**: as we described before it shares a single hardware for multiple virtual machines so it reduces the need for additional physical hardware ,lowering power , cooling , and maintenance costs.
- d. **Testing and development flexibility** : developers can test software in different OS environment without needing multiple device .
- e. **Legacy software support** : older applications that don't work on modern OSes can run in a virtualized environment .
- f. **Scalability and cloud computing** : virtualization is the backbone of cloud computing ,allowing rapid deployment and scaling of containers .
- g. **Disaster recovery and snapshots** : it can be backed up ,cloned ,or restored instantly .

Generally ,virtualization enhances performance , security ,cost-efficiency and flexibility in modern computing . it powers everything from personal development setups to large-scale cloud infrastructure .

## How virtualization

Virtualization in modern operating system works by using a hypervisor , a software layer , to abstract the physical hardware and create virtual machine that run as if they were on separate physical computer .

Let's now try to explain this more

**1. Hypervisor** : it is the core of virtualization .it act as an intermediary between the physical hardware and the virtual machines.

It manages the allocation of resources to the different virtual machines .

**2.virtual machines** : are soft based emulation of physical computers. Each VM has it's own virtual hardware allocated from the host machine .

**3.How it works** : The hypervisor creates and manages the virtual machines, allocating resources as needed .

When a virtual machine requests access to hardware resources , the hypervisor handles the request and manages the interaction with the underling physical hardware .

This allows virtual machines to operate independently and as if they are running on separate physical machines, even through they share the same underling hardware.

**Steps how virtualization works :**

**1.preparing the system** : before virtualization , ensure your system meets the requirement ,including enabling process virtualization in the BLOS/UEFI settings.

**2.installing the hypervisor** : install the hypervisor software on your host operating system .

**3.creating and importing Vms** : create a new VM , defines its virtual hardware and import a guest operating system.

**4.starting the VM** :start the virtual machine ,and the guest operating system will begin to boot and operate within the VM .

**5.using the Vm :** we can use the virtual machine as we would a regular computer ,with it's own operating system and application running isolated from the host .

**6. managing Resources :** The hypervisor manages the resource allocated to each VM ,ensuring that they don't conflict and that the host system remains stable.

**7. shutting down the VM :** when we finish with VM ,shout it down through the hypervisor interface.

## Implementing system call chmod()

Chmod() system call is used to change the file mode of a specified file or directory . the purpose is to control access to files and directories by defining who can read , write , or execute a file.

To implement chmod() system call we follow several steps

### Step 1. Understanding chmod():

The chmod() change the permission of a file.it takes two arguments one is the pathname of the file and the second is the new permissions to be set.these permissions are typically specified using octal notation.

### Step 2.setting up your development environment :

- a. Install necessary tools : we must insure that we have the document tools installed , including the C compiler , make utility and kernel headers for puppy Linux.
- b. Get the kernel source : download the puppy Linux kernel source code . we can found it on puppy linux website.

### Step 3. Locate the system call :



- a. Identify the system call table : open the kernel source code and locate the system call table .
- b. Find implementation : the actual implementation is of chmod() is found in files related to file management.

#### **Step 4. Modify or implement :**

- a. Understanding existing code : understand the existing code to handle cases like updates or changes.
- b. Implementing your changes : if we are modifying the existing implementation , ensure you maintain it's structure while adding your logic.

let us take example

```
#include <linux/fs.h>
```

```
#include<linux/uaccess.h>
```

```
#include<linux/namei.h>
```

```
Long sys_chmod(const char __user*filename,umode_t mode) {
```

```
    Struct path path ;
```

```
    Struct dentry*dentry;
```

```
    Struct inode*inode;
```

```
    Int error;
```

```
    error=kern_path(filename,LOOKUP_FOLLOW,&path);
```

```
    if(error)
```

```
        return error;
```

```
    dentry = path.dentry;
```

```
    inode = dentry->d_inode;
```

```

//check the permission

If(!capable(CAP_FOWNER))

    Return-EPERM;

//update inode permissions

Inode->i_mode &=S_IRWXU;

Inode->i_mode|=(mode & S_IRWXU);

Mark_inode_dirty(inode);

Return 0;

}

```

**Step 5.update system call table:** if we have added a new system call instead of modifying an existing one ,we will need to update the syscall table to include your new function.

**Step 6.Rebuild the kernel :**compile your change and reboot into your new kernel.

**Step 7: testing your implementation :**

Write test case: create a simple c program to test your chmod().

Check for errors ; ensure your implementation handles errors correctly and behaves as expected.

Generally implementing the chmod() system call in puppy linux involves understanding kernel development and system call in linux.

**THANK YOU!!!**