



BahirDar University

Institute of Technology

Faculty of Computing

Department of Software Engineering
Operating System and System Programming
Individual Assignment

Name: Belen Kassa

ID: BDU1600770

Section: B

Submitted to : Mr. Wendmu Baye

Submission Date : 16/08/2017 E.C

OS: PUREOS

Virtualization Tool: Oracle VM VirtualBox

1. Installation of Operating System in Virtual Environment

A Introduction.

PureOS is a privacy-focused, secure Linux distribution developed by **Purism**. It's based on **Debian testing**, combining **rolling and point release** models for a balance of freshness and stability.

It runs entirely on **free and open-source software**, and is approved by the **Free Software Foundation**. With **GNOME or KDE Plasma** desktops, **GNOME Web** as the browser, and **DuckDuckGo** as the default search engine, PureOS is built to protect your privacy and give you full control over your system.

Perfect for anyone who values **freedom, security, and ethical tech**.

B Objectives

The primary objective of installing **PureOS** in **Oracle VirtualBox** is to create a **secure, isolated environment** for exploring its privacy-focused features without affecting the host system. This method is ideal for testing, experimentation, and ensuring security.

1 Isolation and Security

PureOS focuses on privacy and free software. VirtualBox provides an additional layer of **isolation**, preventing any interactions between PureOS and the host system, thus protecting sensitive data.

2 Testing and Experimentation

VirtualBox allows users to test **PureOS** safely, experiment with different configurations, and run applications without the risk of impacting the main system. It's perfect for exploring new features or conducting software trials.

3 Portability and Backup

Virtual machines are easily backed up, copied, and moved. This ensures your PureOS setup and its configurations are safe and can be restored or transferred as needed.

4 Multiple Operating Systems

VirtualBox lets you run **PureOS** alongside your host OS, allowing easy switching between systems without rebooting. This makes it simple to use both OSs simultaneously for different tasks.

C. Requirements

Hardware:

- CPU: Intel Core i5 / AMD equivalent
- RAM: 4 GB or more
- Disk: 20 GB or more
- BIOS: Virtualization enabled (VT-x/AMD-V)

Software:

- Oracle VM VirtualBox
- PureOS ISO (download from <https://pureos.net>)
- Host OS: Windows 10+

D. Installation Steps

1. Download and Install Oracle VirtualBox

Visit the official [Oracle VirtualBox website](https://www.oracle.com/virtualbox/evaluation/).

Download the version for your operating system (Windows/Linux/macOS).

Install it by following the setup instructions.

2. Download the PureOS ISO File

Go to the [PureOS official website](https://pureos.net).

Click "Download PureOS"

Choose the Live IOS Version(usually ends with.ios)

3. Create a New Virtual Machine in VirtualBox

Open VirtualBox, click "**New**".

Name your VM (e.g., "PureOS")

select **Type: Linux**

Version: Debian (64-bit).

Click **Next**.

4. Allocate RAM

Assign at least **2048 MB (2 GB)** of RAM for smoother performance.

Click **Next**.

5. Create a Virtual Hard Disk

Select "**Create a virtual hard disk now**" → click **Create**.

Choose **VDI (VirtualBox Disk Image)** → click **Next**.

Select **Dynamically allocated** → click **Next**.

Set the disk size to at least **20 GB** → click **Create**.

6. Mount the PureOS ISO File

Select your new VM and click **Settings** → go to the **Storage** tab.

Under **Controller: IDE**, click the **Empty** disk icon.

On the right, click the **disk icon**, then **Choose a disk file**.

Browse to your downloaded **PureOS ISO** file and select it → click **OK**.

7. Start the Virtual Machine

Click **Start** to launch the VM.

The PureOS live installer will load from the ISO file.

8. Install PureOS

1. Launch the Installer:

- Double-click the “Install PureOS” icon on the desktop to open the installer.

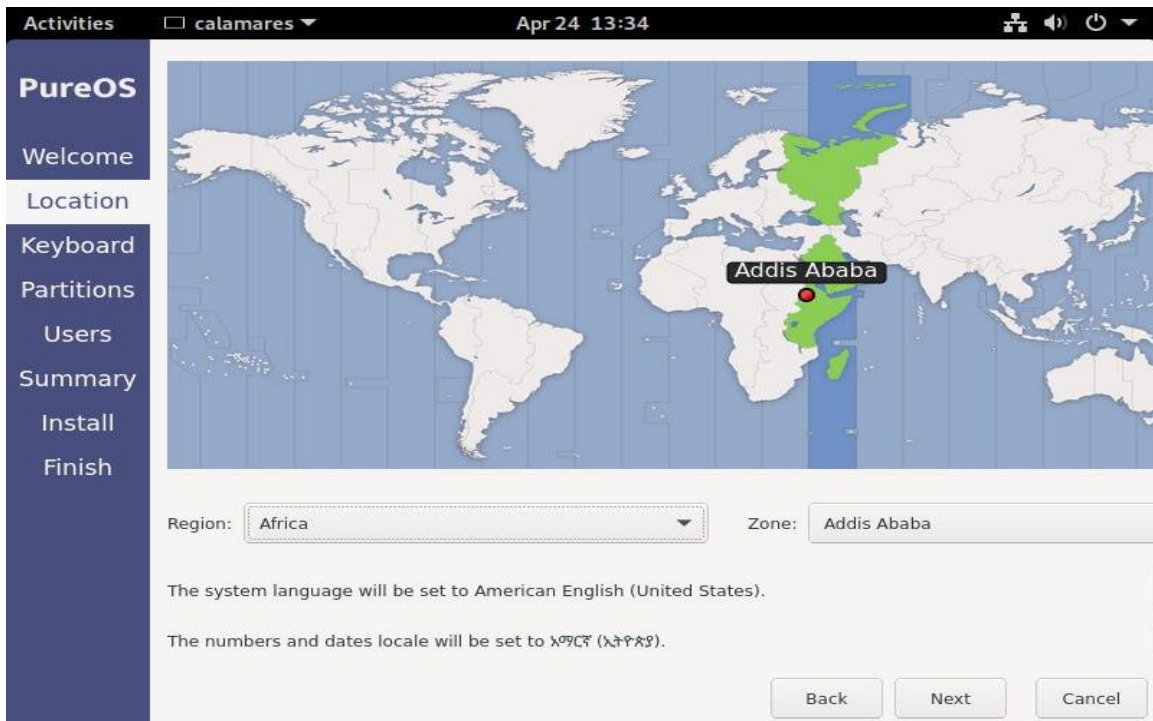
2. Select Language:

- Choose your preferred language. Click Next.



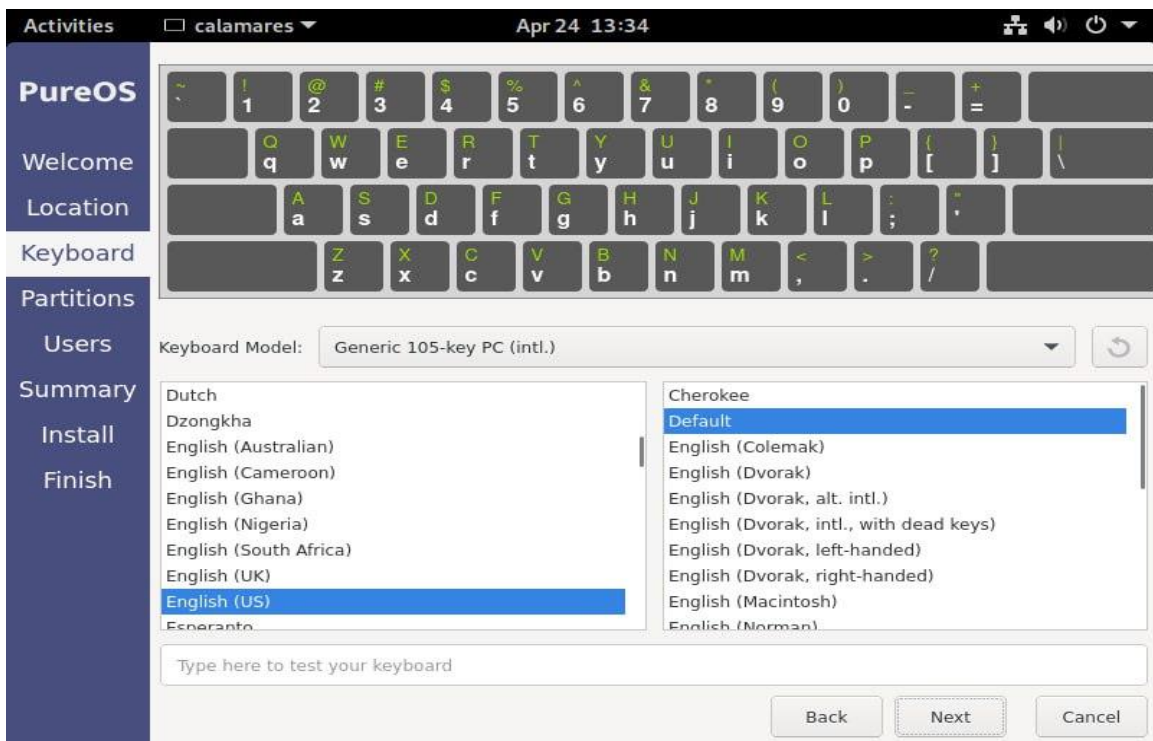
3. Choose Your Location / Time Zone:

- Select your country and time zone. Click Next.



4. Configure Keyboard Layout:

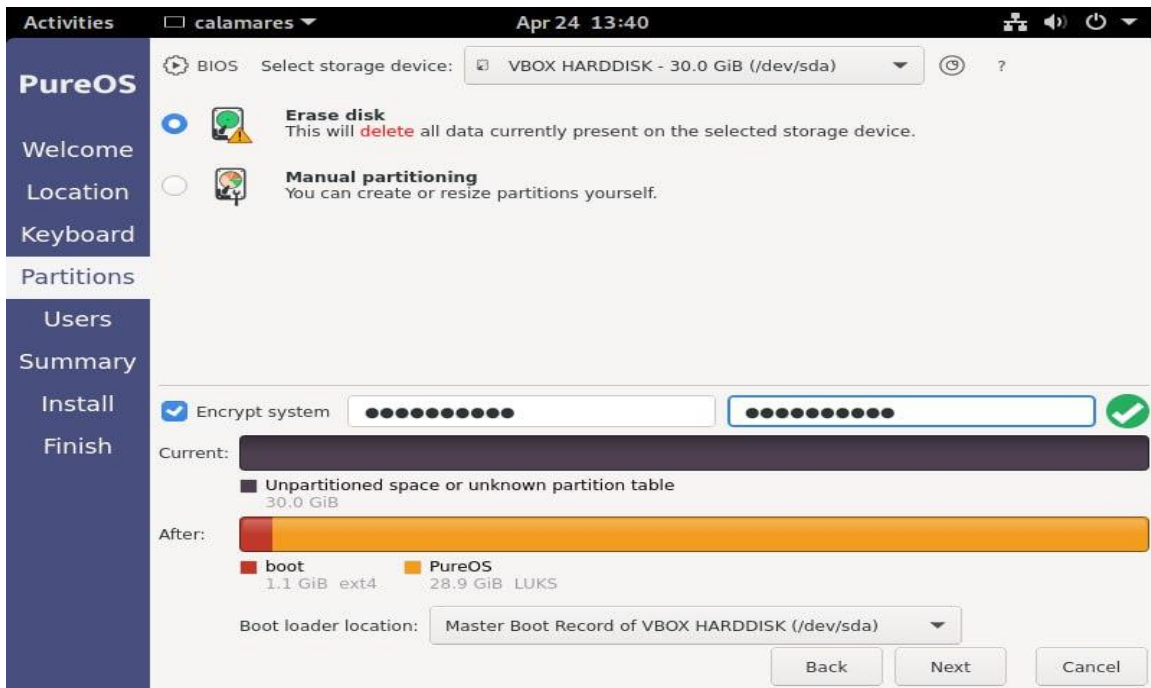
- Choose your keyboard layout and test it. Click Next.



5. Set Up Disk Partitioning:

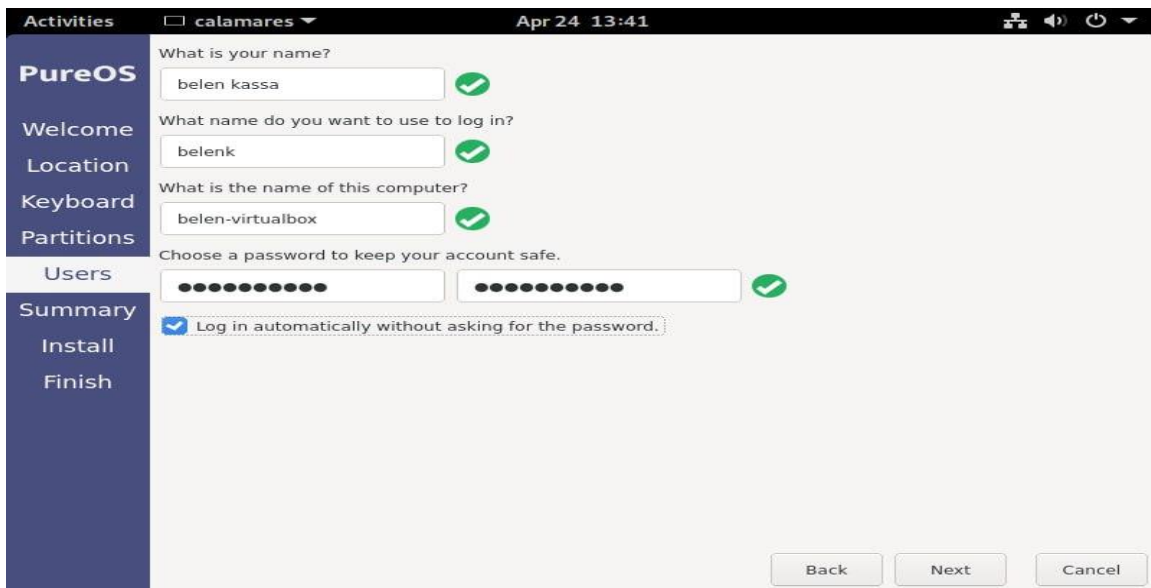
- Choose "Erase disk" for automatic partitioning (recommended).

- Or select manual partitioning for custom setup. Click Next.



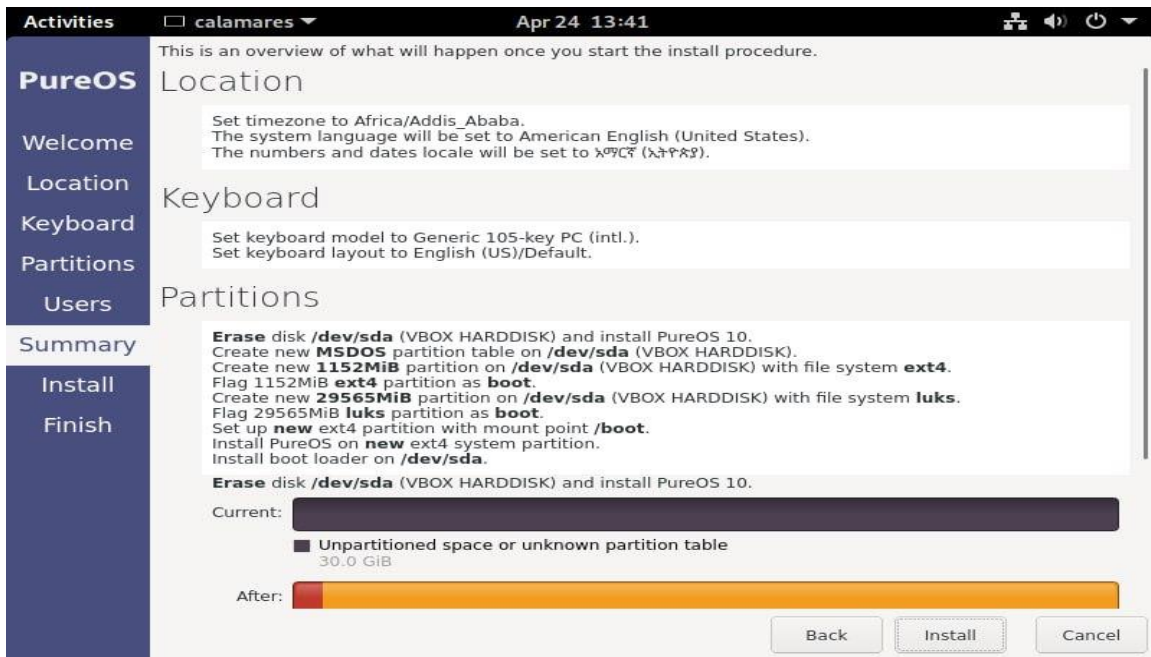
6. Create a User Account:

- Enter full name, username, password, and hostname. Choose login options. Click Next.



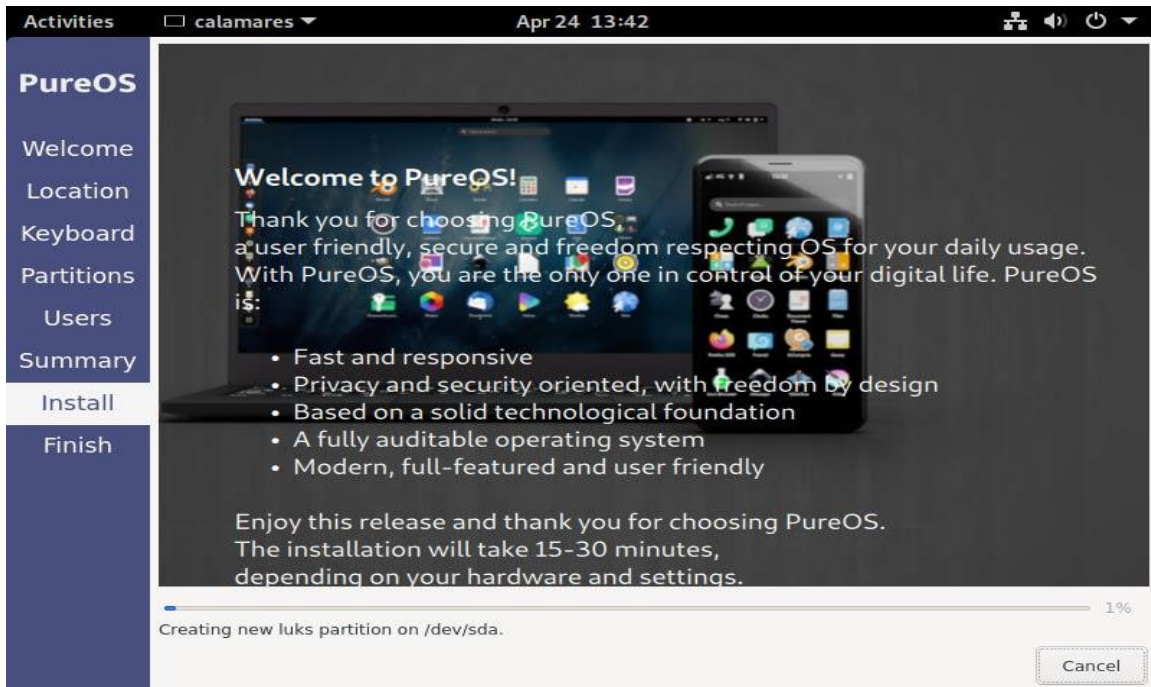
7. Confirm Settings:

- Review the summary of your configuration. Click Install.



8. Installation Progress:

- Wait as files and packages are installed.



9. Restart the System:

- Click "Restart Now."
- Remove the ISO from the virtual drive to avoid booting back into the live session.

Installation Complete!

You now have a fully functional PureOS virtual machine running in VirtualBox. You can begin exploring its features and working on your assignments.

9. Reboot After Installation

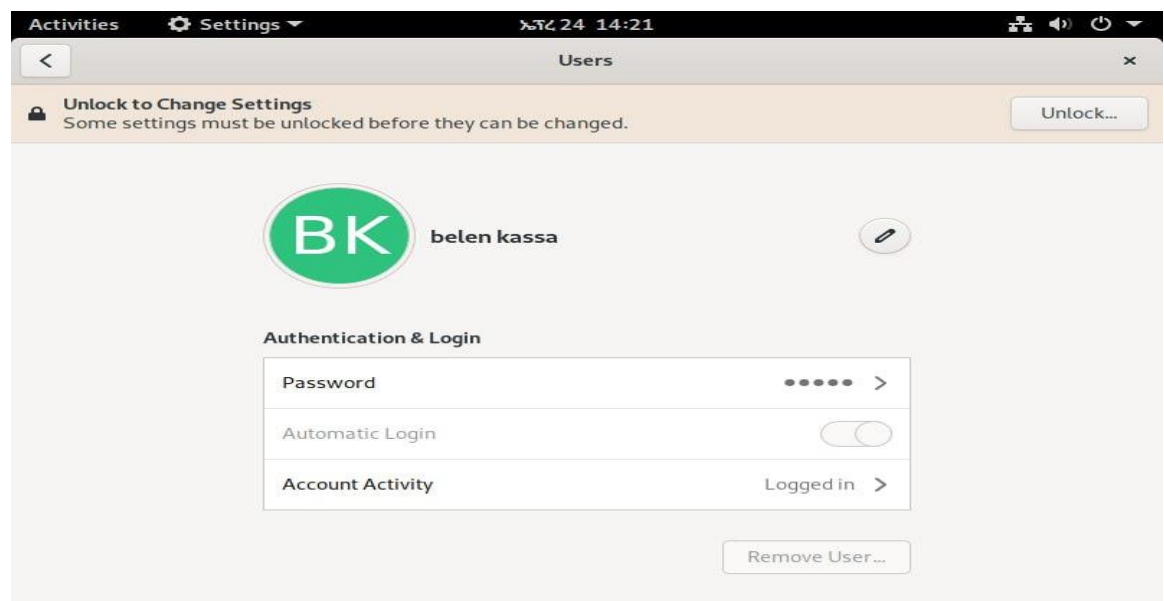
Once installation is complete, **reboot** the system.

If prompted, remove the ISO from virtual media (go to Settings → Storage → remove ISO).

Restart the VM to boot into your new PureOS system.

10. Login and Start Using PureOS

Enter your username and password to access PureOS.



You can now explore its features in a **safe and isolated environment!**

E. Issues & Solutions

Common issues and their solution

1 . VM not booting from ISO

-Check ISO is attached, set Optical Drive first

2. Black screen or freeze

-Enable 3D Acceleration, increase video memory

3. No hard disk found Create and attach virtual hard disk

-Create and attach virtual hard disk

4 .System is slow

-Give more RAM & CPUs, enable virtualization in BIOS

5 .No internet

- Set network to NAT or Bridged

6. Screen won't resize

-Change resolution in settings or install Guest Additions

7. Installation crashes

-Re-download ISO, check RAM and disk size

Problems I Faced and Their Solutions

Issue 1: Virtualization Not Enabled (VT-x Error)

When starting the VM, I got an error: 'VT-x is disabled in the BIOS'.

Solution: I entered the BIOS setup (F10 on HP), enabled Intel Virtualization Technology (VT-x), saved and restarted. This allowed the VM to run successfully.

Issue 2: Installer Not Advancing (Next Button Not Working)

During partitioning, the 'Next' button was inactive.

Solution: I had selected encryption but left the passphrase blank. I unchecked 'Encrypt system' and reselected 'Erase Disk'. The Next button worked after that.

Issue 3: PureOS Kept Booting into Installer

After installation, the VM booted into the installer again.

Solution: I removed the ISO from VirtualBox (Settings > Storage > Remove Disk from Virtual Drive). Then it booted into the installed OS.

Issue 4: Fullscreen Mode Hiding Menu Bar

I couldn't exit the VM or see the menu.

Solution: Pressed Right Ctrl + F to exit fullscreen. The menu reappeared and I continued setup.

F. Filesystem Support in PureOS

PureOS supports a variety of Linux file systems, with **ext4** being the default for installation. .
Other supported file systems include:

ext3 and **ext2** – older versions of ext4

Btrfs – for advanced features like snapshots and compression

XFS – high-performance file system for large files

FAT32/exFAT/NTFS – for compatibility with USB drives and Windows systems (read/write support)

Why ext4?

Because It is stable, reliable, and well-supported

- Journaling protects against crashes.
- Better performance than ext3.

-Reliable for SSD/HDD storage.

G. Advantages and Disadvantages

ADVANTAGES

Privacy-Focused: Built with privacy and security at its core, using open-source software to protect user data.

Free and Open-Source: Only includes free software, ensuring transparency and freedom.

Built-in Security: Uses privacy tools like **DuckDuckGo** for search and **HTTPS Everywhere** for encrypted web connections.

Debian-Based: Inherits the stability and vast software library of Debian.

User-Friendly Interface: Default desktop environments (GNOME or KDE) are easy to use and customize.

DISADVANTAGES

Software Compatibility: Some proprietary software or hardware may not work out of the box.

Hardware Support: Limited support for some proprietary hardware, especially with newer devices.

Learning Curve: May be challenging for beginners who are new to Linux.

Limited Gaming Support: Not as widely supported for gaming compared to other distributions like Ubuntu.

Performance: While generally stable, it can sometimes run slower on older or less powerful hardware.

[H. Conclusion](#)

Installing **PureOS** in **Oracle VirtualBox** is an effective way to experience the privacy-focused features of PureOS without modifying your primary operating system. It offers a safe and isolated environment for testing, learning, and running applications. While there may be minor challenges, such as adjusting hardware settings and ensuring network connectivity, the process is straightforward and provides a secure setup. This method is perfect for users who want to try out PureOS's privacy features in a flexible, non-invasive way.

Installing PureOS virtually deepens understanding of Linux OS structure, filesystem behavior, and system configuration.

2.Virtualization in Modern Operating Systems

Virtualization in Modern Operating Systems - PureOS Context

What is Virtualization?

Virtualization is a method that allows multiple operating systems to run on a single physical computer. Tools like Oracle VirtualBox or VMware enable the creation of virtual machines (VMs), which behave like real computers. PureOS, for instance, can be installed and operated within such an environment.

Why Use Virtualization (with PureOS)?

- Enables testing of PureOS without altering the host system
- Reduces hardware requirements and associated costs
- Provides a safe and isolated environment for experimentation
- Supports easy backup and restoration through snapshots
- Allows running legacy or unsupported software safely

How It Works

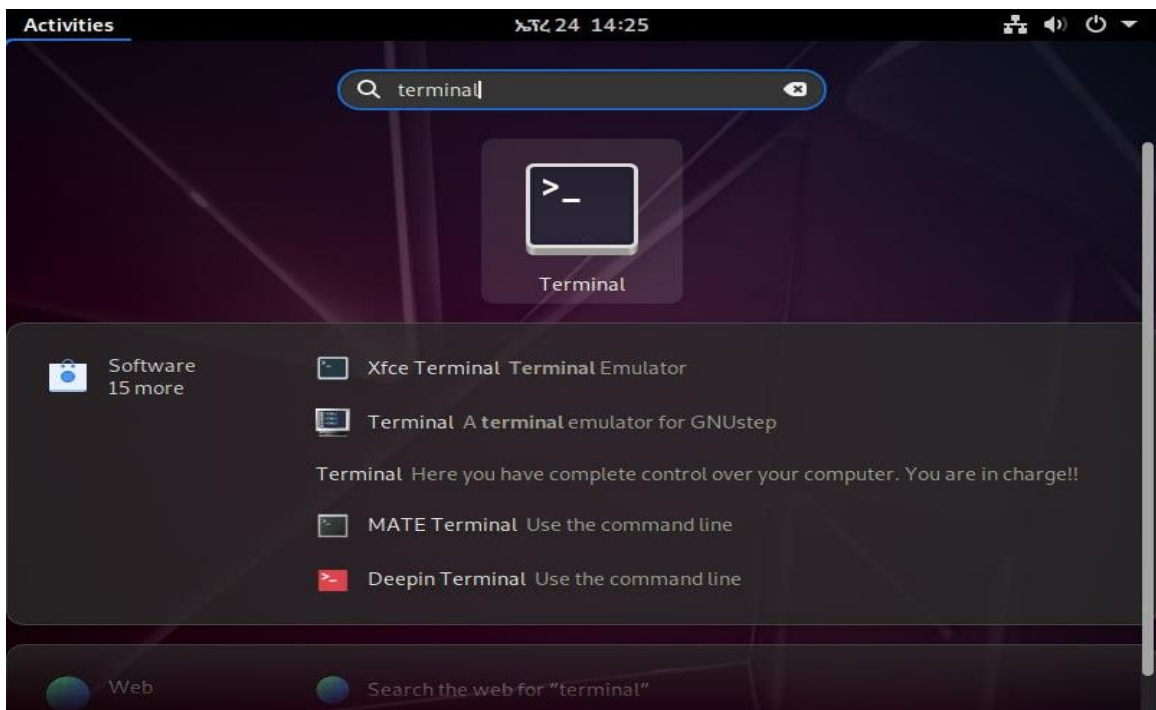
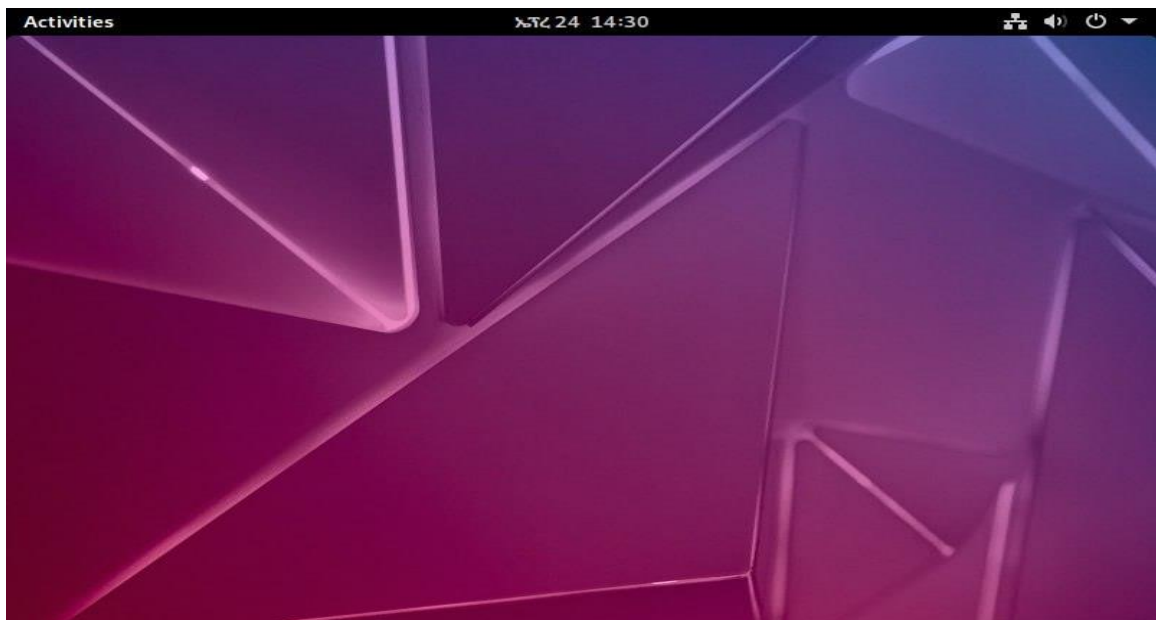
Virtualization is managed by a hypervisor such as Oracle VirtualBox, which allocates hardware resources (CPU, memory, etc.) to virtual machines. After downloading the PureOS ISO file, it is installed on the virtual machine, which functions independently within the host system.

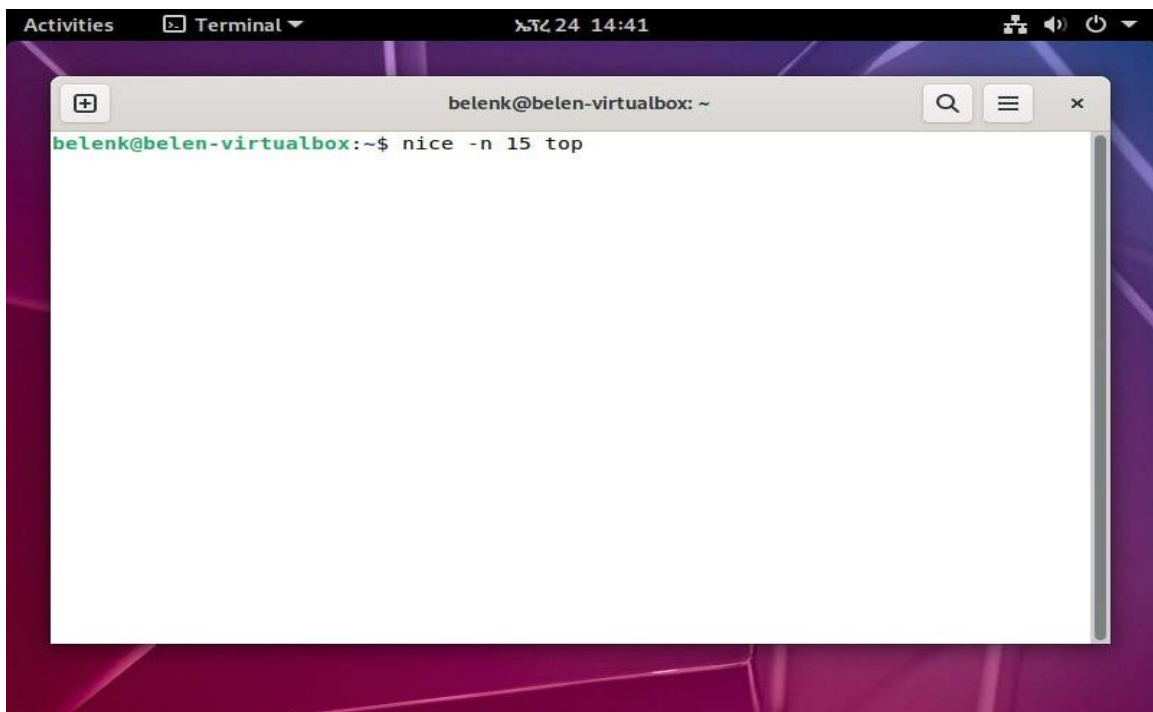
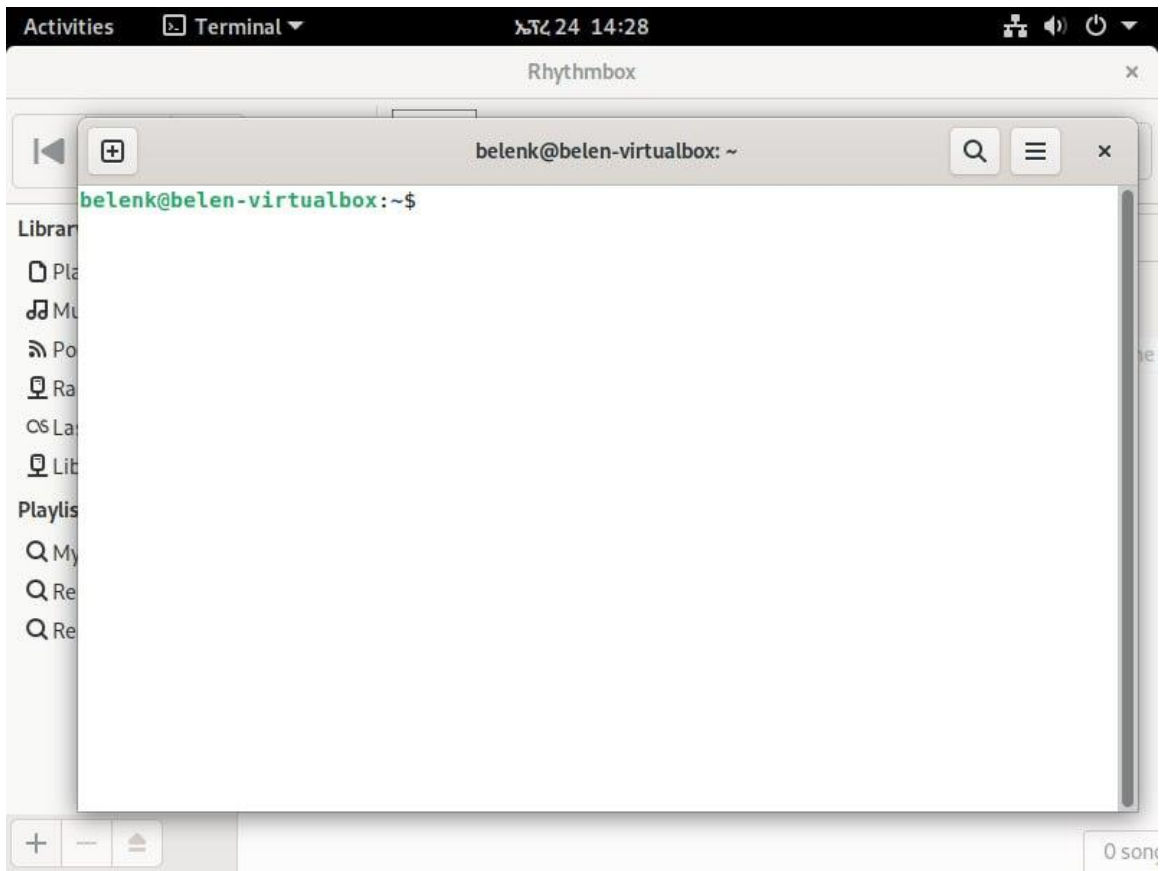
4. System Call Implementation with nice()

The `nice()` system call in Linux and UNIX-based operating systems is used to set the scheduling priority of a process. It allows a process to voluntarily reduce its CPU priority, making it more 'nice' to other processes.

To demonstrate this, I opened the Terminal and executed the following command:

```
nice -n 15 top
```





This launched the 'top' process with a niceness value of 15, indicating lower priority.


```
Activities Terminal 24 14:41

belenk@belen-virtualbox: ~
belenk@belen-virtualbox:~$ nice -n 15 top

top - 14:41:55 up 24 min, 1 user, load average: 0.15, 0.22, 0.25
Tasks: 187 total, 2 running, 185 sleeping, 0 stopped, 0 zombie
%Cpu(s): 5.3 us, 1.8 sy, 0.0 ni, 86.5 id, 6.1 wa, 0.0 hi, 0.3 si, 0.0 st
MiB Mem : 3871.5 total, 1059.1 free, 1395.5 used, 1417.0 buff/cache
MiB Swap: 3871.0 total, 3871.0 free, 0.0 used, 2219.7 avail Mem

  PID USER   PR    NI  VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
 1137 belenk  20     0 4169056 278876 126068 S  10.8   7.0   0:38.53 gnome-she+
 4010 belenk  20     0 557660 47532 37692 S   5.1   1.2   0:00.58 gnome-ter+
 140 root    20     0      0      0      0 I   0.3   0.0   0:00.36 kworker/u+
2054 belenk  20     0 86.3g 151516 87120 S   0.3   3.8   0:01.50 yelp
2072 belenk  20     0 87.1g 272080 137788 S   0.3   6.9   0:07.89 WebKitWeb+
   1 root    20     0 164200 10408 7628 S   0.0   0.3   0:01.38 systemd
   2 root    20     0      0      0      0 S   0.0   0.0   0:00.00 kthreadd
   3 root     0  -20      0      0      0 I   0.0   0.0   0:00.00 rcu_gp
   4 root     0  -20      0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
   5 root    20     0      0      0      0 R   0.0   0.0   0:00.43 kworker/0+
   6 root     0  -20      0      0      0 I   0.0   0.0   0:00.00 kworker/0+
   7 root    20     0      0      0      0 I   0.0   0.0   0:01.01 kworker/u+
   8 root     0  -20      0      0      0 I   0.0   0.0   0:00.00 mm_percpu+
   9 root    20     0      0      0      0 S   0.0   0.0   0:00.00 rcu_tasks+
  10 root    20     0      0      0      0 S   0.0   0.0   0:00.00 rcu_tasks+

Activities Terminal 24 14:42

belenk@belen-virtualbox: ~
top - 14:42:04 up 24 min, 1 user, load average: 0.13, 0.21, 0.25
Tasks: 187 total, 1 running, 186 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2.7 us, 0.2 sy, 0.0 ni, 97.0 id, 0.0 wa, 0.0 hi, 0.2 si, 0.0 st
MiB Mem : 3871.5 total, 1059.1 free, 1395.5 used, 1417.0 buff/cache
MiB Swap: 3871.0 total, 3871.0 free, 0.0 used, 2219.7 avail Mem

  PID USER   PR    NI  VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
 1137 belenk  20     0 4169072 278884 126068 S   6.2   7.0   0:38.87 gnome-she+
 4010 belenk  20     0 557660 47532 37692 S   1.6   1.2   0:00.70 gnome-ter+
2072 belenk  20     0 87.1g 272080 137788 S   0.3   6.9   0:07.93 WebKitWeb+
 4142 belenk  35    15 10504  3932  3448 R   0.3   0.1   0:00.05 top
   1 root    20     0 164200 10408 7628 S   0.0   0.3   0:01.38 systemd
   2 root    20     0      0      0      0 S   0.0   0.0   0:00.00 kthreadd
   3 root     0  -20      0      0      0 I   0.0   0.0   0:00.00 rcu_gp
   4 root     0  -20      0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
   5 root    20     0      0      0      0 I   0.0   0.0   0:00.43 kworker/0+
   6 root     0  -20      0      0      0 I   0.0   0.0   0:00.00 kworker/0+
   7 root    20     0      0      0      0 I   0.0   0.0   0:01.02 kworker/u+
   8 root     0  -20      0      0      0 I   0.0   0.0   0:00.00 mm_percpu+
   9 root    20     0      0      0      0 S   0.0   0.0   0:00.00 rcu_tasks+
  10 root    20     0      0      0      0 S   0.0   0.0   0:00.00 rcu_tasks+
  11 root    20     0      0      0      0 S   0.0   0.0   0:00.09 ksoftirqd+
  12 root    20     0      0      0      0 I   0.0   0.0   0:00.22 rcu_sched
  13 root    rt     0      0      0      0 S   0.0   0.0   0:00.00 migration+
```

This showed the top process with a nice value of 15. This confirmed the use of the nice() system call.

