Department software engineering

Course OSSP

Individual assignment

Name Zufan Genetu Haregewoien

Id 1602926

Sec B

Submission date 16/08/17 E.C

Submitted to Lec.Wendimu Baye

Elementary os

a.Introduction

Background

This document details the process of installing elementary os 6 within a virtual machine environment.virtualization allows running multiple operating systems concurrently on a single physical machine, offering benefits such as isolated testing environments,resource efficiency,and the ability to experiment with different operating systems without affecting the primary system.this documentation aims to provide a comprehensive,step by step guide,complete with screenshots,for successfully installing elementary os within a VM.

Elementary OS is a Linux distribution based on Ubuntu LTS (Long Term Support). It is known for its custom desktop environment called Pantheon, which emphasizes user experience, simplicity, and a cohesive design aesthetic, often drawing comparisons to macOS. Virtualization software, such as Oracle VM VirtualBox or VMware Workstation, allows users to run one or more operating systems simultaneously on a single physical machine. This is achieved by creating virtual machines (VMs), which emulate computer hardware in software.

Virtualization technology has become an essential aspect of modern computing, allowing users to run multiple operating systems on a single physical machine. This capability is

particularly useful for developers, testers, and system administrators who need to create isolated environments for testing applications or configurations.

Motivation

Elementary OS is a user-friendly Linux distribution based on Ubuntu, known for its aesthetic appeal and ease of use. This document aims to guide users through the installation of Elementary OS in a virtual environment using tools such as VMware Workstation or Oracle VM VirtualBox.

b. Objectives

To successfully install the latest stable version of Elementary OS within an Oracle VM VirtualBox virtual machine.

To understand the basic steps involved in creating and configuring a VM for a Linux distribution.

To provide a step-by-step guide for installing Elementary OS in a virtual environment.

To highlight potential issues during installation and their solutions.

To discuss filesystem support relevant to Elementary OS.

To evaluate the advantages and disadvantages of using Elementary OS in a virtualized environment.

c. Requirements

i. Hardware

CPU: 64-bit processor with hardware virtualization support (Intel VT-x or AMD-V). This must be enabled in the system's BIOS/UEFI settings. (Most modern CPUs support this).

RAM: Minimum 4 GB RAM (8 GB+ recommended for smoother performance). Elementary OS itself needs resources, plus your host OS and the VM overhead.

Disk Space: Minimum 30 GB free hard disk space (SSD recommended for better performance). This space is for the virtual hard disk file.

Display: A display capable of at least 1024x768 resolution.

ii. Software

VMware Workstation or Oracle VM VirtualBox installed on the host system.

Elementary OS ISO file (latest version).

Optional: Guest Additions or VMware Tools for enhanced performance.

d. Installation Steps

i. Step-by-Step Installation

1.Download virtualbox software and install it

2. Download Elementary OS:

Go to the Elementary OS website and download the latest ISO file.

3. Open Virtualization Software:

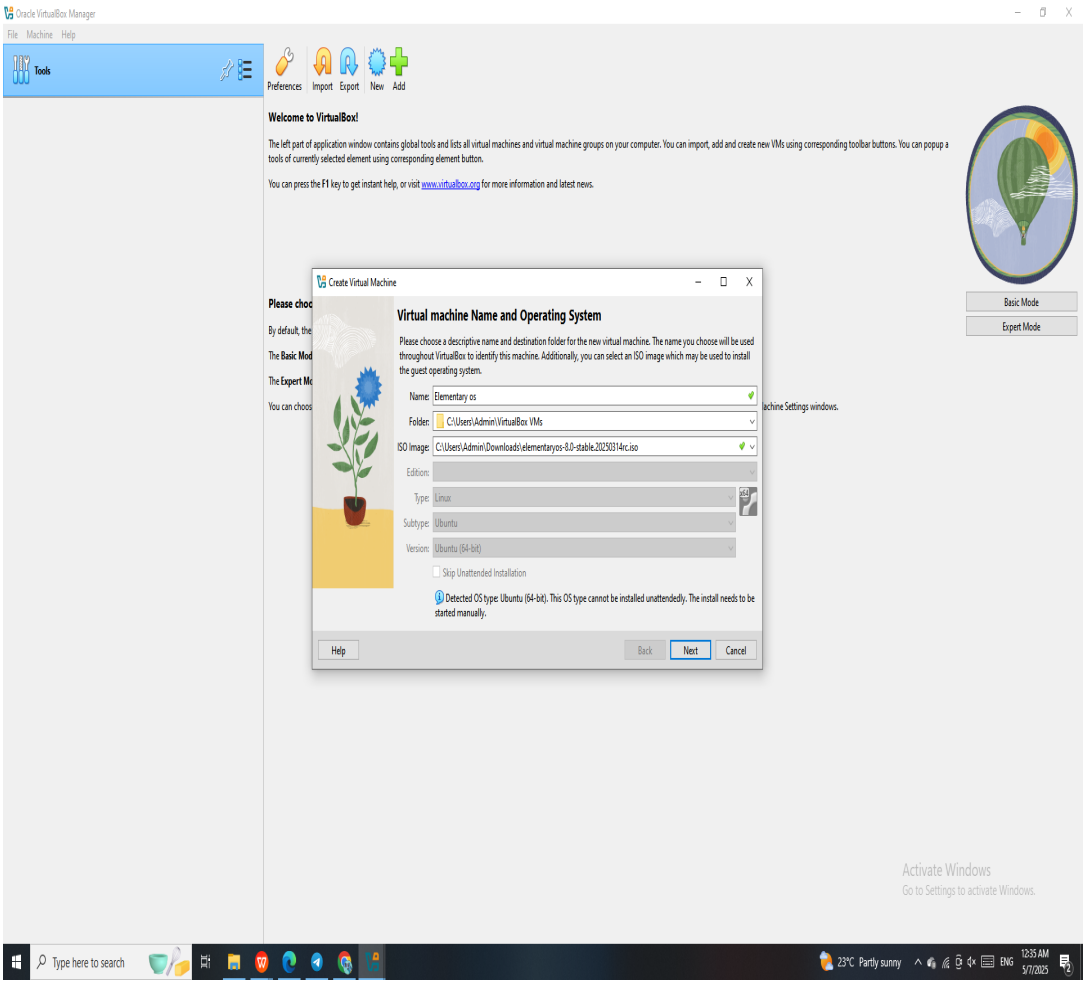
Launch Oracle VM VirtualBox.

4. Create a New Virtual Machine:

Click on "Create New Virtual Machine" and select the option to use an ISO file.

Choose the downloaded Elementary OS ISO.

Create New Virtual Machine



5.Create virtual hard disk

Choose "create a virtual hard disk now."

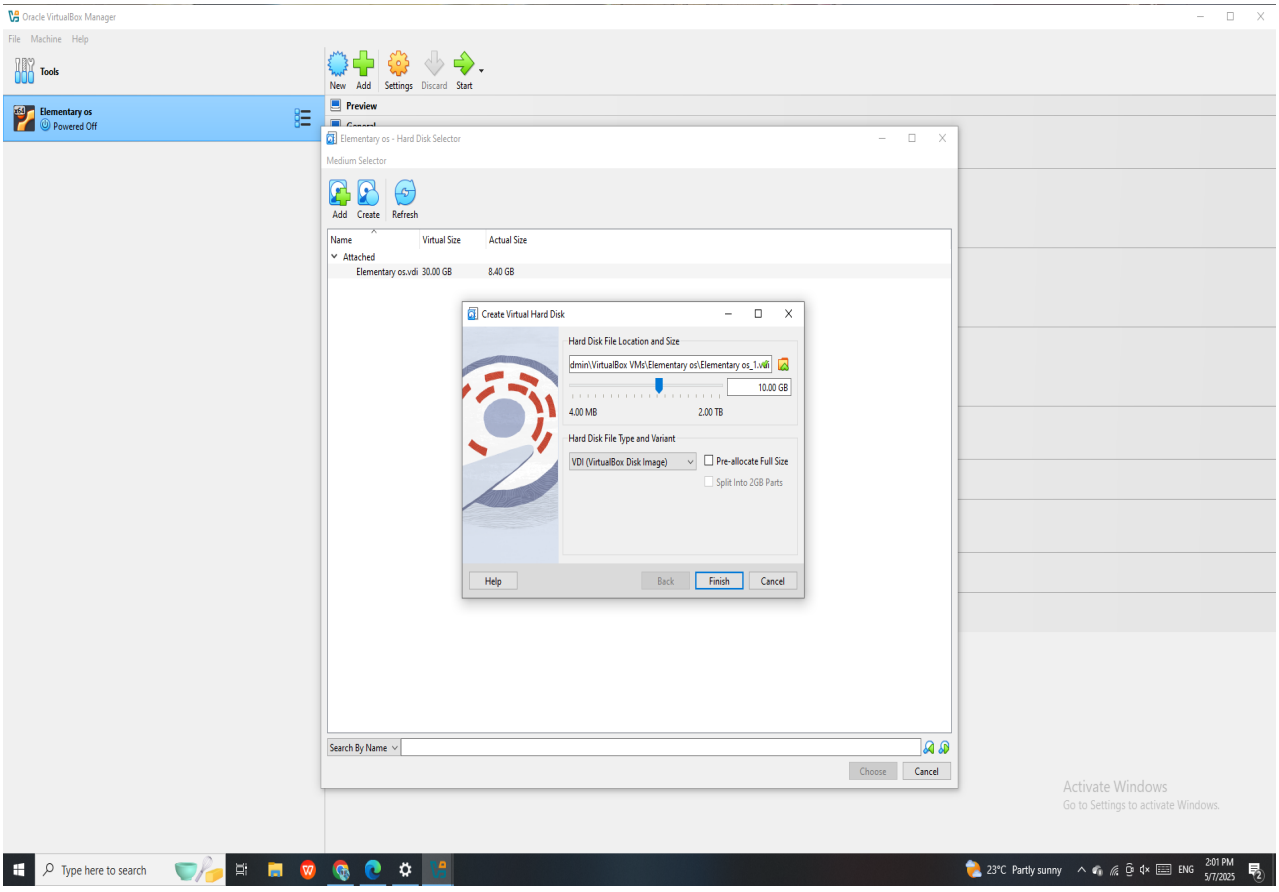
Click " create"

Select VDI(virtualbox disk image)

click "next"

set size: 15-20 GB

Click "create"

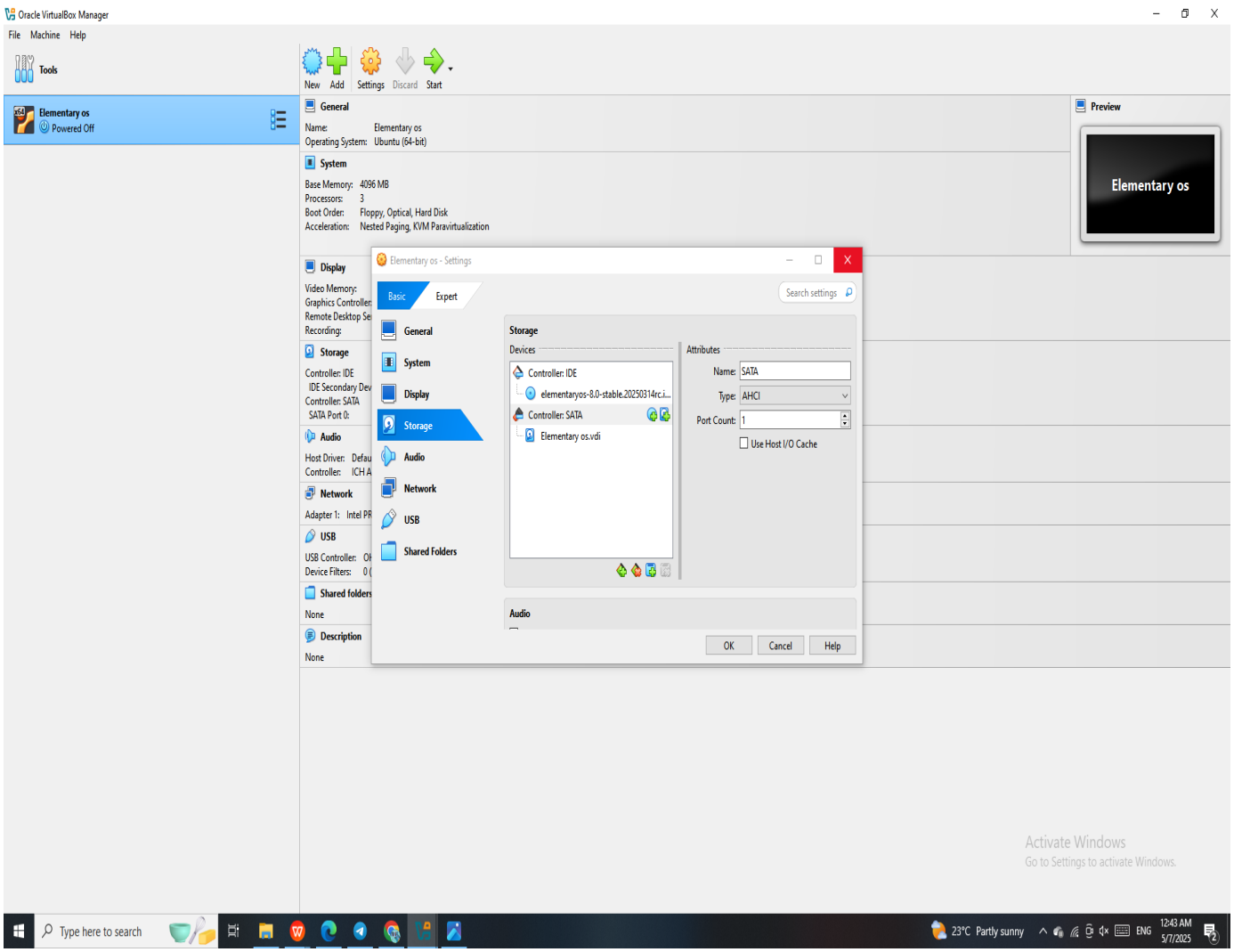


6.Configure VM Settings:

Allocate at least 2 GB of RAM and 20 GB of disk space for the virtual machine.

Configure the network settings as needed (NAT or Bridged).

VM Settings



7. Start the Virtual Machine:

Power on the virtual machine, and it will boot from the ISO file.

8. Install Elementary OS:

Follow the on-screen instructions to install Elementary OS.

When prompted to create an account, use our full name.

9. Complete Installation:

Once installation is complete, restart the virtual machine and remove the ISO from the virtual drive.

To create an account

My name: Zufan Genetu Haregewoien.

My computer's name: A name for the VM on the network (e.g., "elementary-vm").

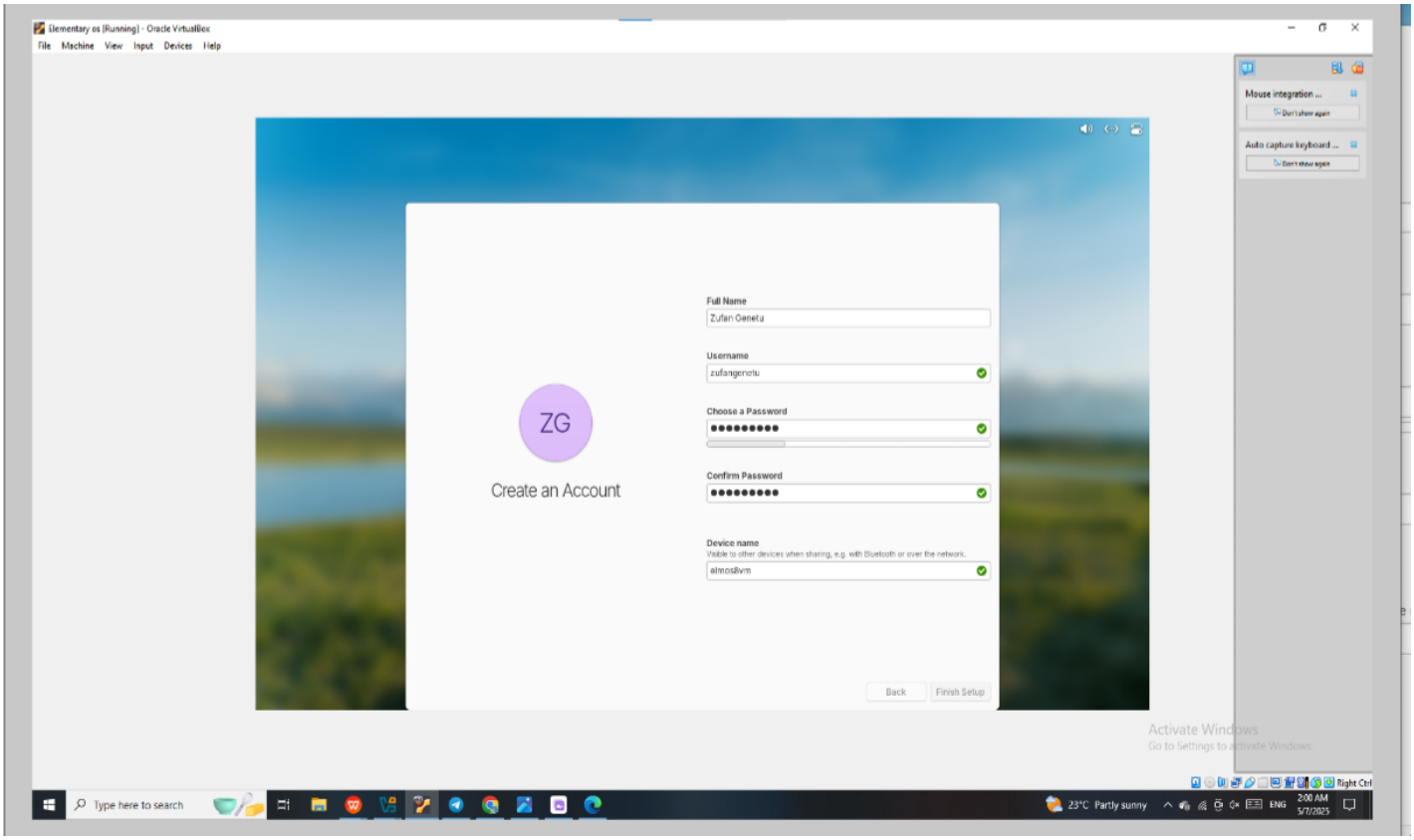
Pick a username: A short, lowercase name for logging in (e.g., "

My full name" or "firstname").

Choose a password: Create and confirm a strong password.

Choose whether to log in automatically or require the password. Requiring the password is more secure.

Click "Continue".



Installation Progress: Wait for the installation process to complete. This may take some time.

Installation Complete: Once finished, we will be prompted to restart the computer. Click "Restart Now".

Remove Installation Medium: VirtualBox should automatically eject the virtual ISO. If you see a message like "Please remove the installation medium, then press ENTER", just press Enter inside the VM window. If it tries to boot into the installer again, power off the VM, go to VM Settings -> Storage, select the Optical Drive, and ensure it's empty. Then start the VM again.

First Boot and Login: The VM will reboot into your newly installed

Elementary OS. Log in using the username and password you created.

Install VirtualBox Guest Additions (Crucial!): For proper screen resolution, mouse integration, copy/paste between host and guest, and shared folders:

In the VirtualBox VM window menu, go to Devices -> Insert Guest Additions CD image....

A virtual CD will appear on the Elementary OS desktop or in the File Manager.

Elementary OS might prompt you automatically to run the software. If so, click "Run" and enter your password.

If not prompted automatically, open the Terminal (Applications -> Terminal).

Navigate to the mounted CD directory. It's usually under /media/My username/VBox_GAs_x.x.xx (use ls /media/My username/ to find the exact name).

Run the installer with sudo ./VBoxLinuxAdditions.run

Enter your password when prompted and wait for the installation to complete.

Once finished, restart the Elementary OS VM (sudo reboot in terminal or use the power menu).

e. Issues (Problems Faced)

Problem 1: VM fails to start, complaining about VT-x/AMD-V not

being available or disabled.

Problem 2: Graphics are slow, screen resolution is stuck at a low value (e.g., 800x600 or 1024x768), mouse pointer is jerky or not captured/released correctly.

Problem 3: Installation freezes or fails partway through.

Problem 4: No network connectivity inside the VM.

Problem 5: After installation and clicking "Restart", the system boots back into the installer instead of the installed OS.

f. Solution

Solution 1: Reboot the host machine and enter the BIOS/UEFI settings (usually by pressing DEL, F2, F10, or F12 during boot). Find the CPU settings and enable "Intel Virtualization Technology (VT-x)" or "AMD-V" / "SVM Mode". Save changes and exit.

Solution 2: This is almost always because VirtualBox Guest Additions are not installed or not working correctly. Follow step d.18 carefully to install or reinstall them, then restart the VM. Also, ensure 3D acceleration is enabled in VM display settings if needed (but try disabling it if problems persist after installing Guest Additions).

Solution 3: Ensure the downloaded ISO image is not corrupt (verify checksum if possible). Allocate more RAM or CPU cores to the VM if possible. Ensure "Enable EFI" is checked in System settings. Try the installation again. Check VirtualBox logs for specific errors.

Solution 4: Check the VM's Network settings (VM selected -> Settings -> Network). For simple internet access, "NAT" is usually the easiest default. Ensure the "Cable Connected" box is checked. If you need the VM to be directly on your local network (like another physical machine), use "Bridged Adapter" and select your host's active network card.

Solution 5: The virtual installation ISO is still attached to the virtual optical drive. Power off the VM. Go to VM Settings -> Storage -> Select the Optical Drive under the Controller: IDE or SATA -> Click the disc icon on the right -> Choose "Remove Disk from Virtual Drive". Start the VM again.

g. Filesystem Support

Elementary OS primarily supports the following filesystems:

ext4: Default filesystem for Linux distributions due to its robustness and performance.

• **Btrfs:** Offers advanced features like snapshots and subvolumes, but may not be as stable as ext4 for all use cases.

Other filesystems like NTFS, FAT32, exFAT are supported but are typically used for interoperability with Windows systems rather than as primary filesystems.

h. Advantages and Disadvantages

Advantages

Isolation: No risk to the host OS.

Testing/Learning: Easy way to explore Linux and Elementary OS features.

Snapshotting: VM software allows taking snapshots to revert to previous states easily.

Portability: Virtual disk files can often be moved between host computers (with the same VM software).

Resource Control: Easily define how much RAM/CPU the OS gets.

Elementary OS Specific: Experience the Pantheon DE, AppCenter, and curated apps.

User-friendly interface suitable for beginners.

Lightweight and efficient performance.

Strong community support and regular updates.

Disadvantages

Performance Overhead: Will always run slower than a bare-metal installation due to the virtualization layer. Noticeable in graphics-intensive tasks or heavy compilation.

Hardware Access: Limited direct access to specialized hardware (e.g., high-end GPUs for gaming, specific peripherals might need USB passthrough configuration).

Resource Consumption: Consumes host system RAM, CPU, and

disk space.

Complexity: Adds a layer of software (the hypervisor) that needs managing.

Graphics Limitations: Even with 3D acceleration and Guest Additions, complex graphical effects or games may not perform well.

i. Conclusion

Installing Elementary OS in a virtual machine like VirtualBox is a straightforward process that provides a safe and flexible environment for exploring this user-friendly Linux distribution. By following the steps outlined, allocating sufficient resources, and installing Guest Additions, users can achieve a functional and responsive Elementary OS experience without impacting their primary operating system. It serves as an excellent platform for testing, learning, and development purposes.

j. Future Outlook / Recommendation

Explore Elementary OS: Dive into the AppCenter, customize the Pantheon desktop, learn basic terminal commands, and explore the pre-installed applications.

Use Snapshots: Leverage the snapshot feature of VirtualBox before making significant changes or installing potentially unstable software.

Shared Folders: Set up shared folders (via Devices -> Shared Folders in VirtualBox, after Guest Additions are installed) for easy

file exchange between the host and guest OS.

Increase Resources: If performance is sluggish, try allocating more RAM or CPU cores to the VM (within the limits of your host hardware).

Consider Bare-Metal: If you enjoy Elementary OS and need better performance, especially for graphics or demanding tasks, consider installing it directly on hardware (dual-booting or replacing your current OS).

Keep Software Updated: Regularly update VirtualBox, Elementary OS, and the Guest Additions for the best performance, security, and compatibility.

2. Virtualization in Modern Operating Systems

What is Virtualization?

Virtualization is the process of creating a software-based, or virtual, representation of something, such as virtual applications, servers, storage, and networks. In the context of operating systems and hardware, it refers to the ability to run multiple operating systems simultaneously on a single physical machine. This is achieved through a software layer called a hypervisor (also known as a Virtual Machine Monitor or VMM). The hypervisor creates and manages Virtual Machines (VMs), allocating physical hardware resources (CPU, RAM, storage, network) to each VM.

Why is Virtualization Used?

Server Consolidation: Reduces the number of physical servers

needed, saving costs on hardware, power, cooling, and space.

Resource Efficiency: Improves utilization of hardware resources, as VMs can share the underlying physical capacity.

Testing and Development: Provides isolated environments for developing, testing, and debugging software without impacting production systems or the developer's main OS.

Disaster Recovery & Business Continuity: VMs can be easily backed up, replicated, and migrated to different hardware, simplifying disaster recovery plans.

Legacy Application Support: Allows running older operating systems and applications on modern hardware.

Security Isolation: VMs provide strong boundaries, isolating applications and operating systems from each other and the host. Malware in one VM generally doesn't affect others or the host.

Desktop Virtualization (VDI): Host desktop environments centrally and stream them to users on various devices.

Cloud Computing: Virtualization is a foundational technology for Infrastructure as a Service (IaaS) cloud offerings.

How Does Virtualization Work?

The hypervisor abstracts the physical hardware from the virtual machines. When a guest OS running inside a VM tries to perform a privileged operation (like accessing hardware), the hypervisor intercepts this request.

CPU Virtualization: Modern CPUs include hardware extensions

(Intel VT-x, AMD-V) that allow the hypervisor to run guest instructions directly on the CPU much more efficiently and securely, minimizing software emulation overhead. The hypervisor schedules time slices for each VM's virtual CPUs on the physical CPU cores.

Memory Virtualization: The hypervisor manages the physical RAM, mapping it to the virtual RAM allocated to each VM. Techniques like memory ballooning or sharing can optimize usage.

I/O Virtualization: The hypervisor intercepts requests for storage and network access, managing access to physical disks and network interfaces, often through emulated or paravirtualized drivers for better performance. Paravirtualized drivers (like those in Guest Additions/VMware Tools) allow the guest OS to communicate more directly and efficiently with the hypervisor.

3. System Calls:

Brief Explanation

What are System Calls?

System calls are the fundamental interface between a process (a running program in user space) and the operating system kernel (which runs in a protected, privileged kernel space). They are how user programs request services that they cannot perform themselves because they require higher privileges, such as interacting with hardware, managing files, creating or managing other processes, and network communication.

Below is simple C programming that demonstrating basic system calls.

```
#include <stdio.h>

#include <unistd.h>


int main() {
    // System call to create a new process
    pid_t pid = fork();

    if (pid == -1) {
        perror("Fork failed");
        exit(EXIT_FAILURE);
    } else if (pid == 0) {
        // Child process
        printf("Hello from child process!\n");
    } else {
        // Parent process
        printf("Hello from parent process! Child PID: %d\n", pid);
    }
```

```
    return 0;
}
```

Explanation:

• fork() is a system call that creates a new process by duplicating the calling process.

• The return value of fork() helps identify whether we are in the parent or child process.

This example illustrates how system calls enable interaction with the operating system's process management functionalities.

Implementing system calls in an operating system like Elementary OS, which is based on Ubuntu, requires a good understanding of the Linux kernel and its architecture.

implementing system calls in a Linux-based operating system:

1. Set Up Your Environment:

• Install the necessary packages for kernel development:

sudo apt-get install build-essential linux-source

2. Download the Kernel Source:

• Download the Linux kernel source code that matches your Elementary OS version. You can usually find it in /usr/src or download it from the kernel.org website.

3. Navigate to Kernel Source Directory:

```
cd /usr/src/linux-<version>
```

4. Choose a System Call Number:

• Each system call has a unique number. Choose an unused number from include/asm/unistd.h.

5. Implement Your System Call:

• Create a new C file for your system call in the kernel/ directory, e.g., my_syscall.c.

• Implement your system call function. For example:

```
#include <linux/kernel.h>
#include <linux/syscalls.h>

SYSCALL_DEFINE0(my_syscall) {
    printk(KERN_INFO "My syscall was called\n");
    return 0;
}
```

6. Register Your System Call:

• Add your system call to the syscall table. This is usually located in arch/x86/entry/syscalls/syscall_64.tbl for x86_64 architectures.

• Add a line like this:

<syscall_number> common my_syscall sys_my_syscall

7. Compile the Kernel:

• Configure the kernel if needed:

make menuconfig

• Compile the kernel and modules:

make

sudo make modules_install

sudo make install

8. Update Bootloader:

• Update your bootloader (GRUB) if necessary and reboot the system.

9. Test Your System Call:

• Write a simple user-space program to test your new system call:

#include <stdio.h>

#include <unistd.h>

```
#include <sys/syscall.h>
#define __NR_my_syscall <syscall_number>
int main() {
    long result = syscall(__NR_my_syscall);
    printf("Result: %ld\n", result);
    return 0;
}
```

• Compile and run your test program.

10. Debugging:

• Check the kernel logs using dmesg to see if your syscall is being invoked correctly.

Important Notes

• Modifying the kernel can lead to system instability. Always back up important data.

• Consider using a virtual machine or a separate partition for kernel development.

• Follow best practices for coding and testing to avoid introducing bugs into the kernel.

Conclusion

Implementing system calls in Elementary OS or any Linux-based OS requires careful manipulation of the kernel source code and understanding of how system calls work at a low level.

Thank you for all!