BAHIR DAR UNIVERSITY BAHIR DAR INSTITUTE OF

TECHNOLOGY

Faculty of computing

Title: Installation of an operating system (WINDOWS SERVER 2022 OS) , why ,what and why virtualization and implementation of system call(accept)

INDIVISUAL ASSIGNMENT : Operating systems and system programming

NAME: Rahel Gedefaw

ID: BDU1602308

Section:B

Sub.date:16/08/2017E.C

Sub.to: Lec.Wendimu Baye

Table of contents

1.0 Introduction

Operating System (OS)

An operating system (OS) is the fundamental software that manages computer hardware and software resources. It acts as an intermediary between applications and the hardware.

It provides essential services like:

Resource Management: Allocating CPU time, memory, and storage to different programs.

Process Management: Creating, scheduling, and terminating processes (running programs).

Memory Management: Managing the computer's memory (RAM) to ensure efficient usage.

File System Management: Organizing and storing files on storage devices.

Input /Output (I/O) Management: Handling communication between the computer and peripheral devices (e.g., keyboards, mice, printers).

User Interface (UI): Providing a way for users to interact with the computer (graphical or command-line).

Most of the time we Install operating systems on our devices. What is the motivation?

* The primary purpose of installing an operating system is to:

1. Enable Functionality: To make a computer system usable. Without an OS, a computer is just a collection of electronic components. The OS brings the hardware to life.

2. Provide a Platform: To create a foundation upon which applications can run. Applications are designed to work with a specific OS, so installing the OS is a prerequisite for running those applications.

3. Manage Resources: To efficiently manage the computer's resources (CPU, memory, storage) and ensure that applications can access those resources in a controlled and secure manner.

4. Offer a User Interface: To provide a way for users to interact with the computer, whether through a graphical user interface (GUI) or a command-line interface (CLI).

5. Enable Networking: To allow the computer to communicate with other computers on a network.

6. Provide Security: To protect the computer from malware, viruses, and unauthorized access.

Installing an OS transforms a collection of hardware into a functional, manageable, and interactive computing system. It's the foundation upon which all other software and activities are built.

From a list of operating systems now for this project I will focus on

Installation of Microsoft windows server specially the latest version like windows server 2022.

Here is a few list from its functions

• **Centralized Management:** Windows Server provides tools for centralized management of users, computers, applications, and resources, simplifying administration and improving efficiency in organizations.

- **Security**: Windows Server incorporates robust security features to protect data and prevent unauthorized access. Each version builds upon the previous ones with improved security measures. Windows Server 2022 has enhanced secured-core server capabilities, firmware protection, and hardware-based security.

- **Networking**: It enables the creation and management of complex networks, including routing, DNS, DHCP, and VPNs. It's a foundation for building network infrastructure.

- **Application Hosting**: Windows Server provides a platform for hosting a wide range of applications, from web servers (IIS) and database servers (SQL Server) to custom business applications.

- **Virtualization:** Hyper-V, Microsoft's virtualization technology, allows you to run multiple virtual machines on a single physical server, maximizing resource utilization and reducing hardware costs.

1.1 Historical Development of Microsoft Server

The Microsoft Server operating system has a long and evolving history, closely tied to the development of the Windows NT family.

Here is simplified life time

- Early Days (Windows NT Server 3.1 - 4.0): The lineage began with Windows NT Server 3.1 in 1993, designed to provide a more robust and reliable server platform compared to Windows for Workgroups. These early versions focused on file and print sharing, basic application serving, and domain-based networking. Windows NT 4.0 (1996) introduced the familiar Windows 95-like interface and improved performance. Key improvements included the introduction of the Registry and support for more hardware.

- Windows 2000 Server: A significant leap forward. Based on the NT kernel, it introduced Active Directory, a directory service that revolutionized network administration, enabling centralized management of users, computers, and resources. It was a cornerstone for enterprise networking.

- Windows Server 2003: Built upon the Windows 2000 foundation, focusing on enhanced reliability, security, and performance. It included improvements to Active Directory, IIS (Internet Information Services) web server, and the .NET Framework.

- Windows Server 2008 & 2008 R2: Marked a significant architectural change with the introduction of Server Core (a minimal installation option without a GUI) and virtualization technologies like Hyper-V, competing with VMware. 2008 R2 focused on improving virtualization, management, and power efficiency.

- Windows Server 2012 & 2012 R2: Embraced cloud computing and virtualization with significant enhancements to Hyper-V, storage, and networking capabilities. PowerShell became increasingly important for server administration.

- Windows Server 2016: Focused on security enhancements, software-defined networking, and containerization support with Docker. Nano Server (an even smaller footprint than Server Core) was introduced.

- Windows Server 2019: Improved hybrid cloud capabilities, storage migration services, and further enhanced security features.

- Windows Server 2022: The latest major release. Key features include enhanced security (secured-core server), advanced multi-layer security, hybrid capabilities with Azure integration, and platform improvements for modern applications and containers. It emphasizes security, cloud integration, and application platform enhancements.

2.0 Objectives

As software engineer the main goals of installing windows OS are:

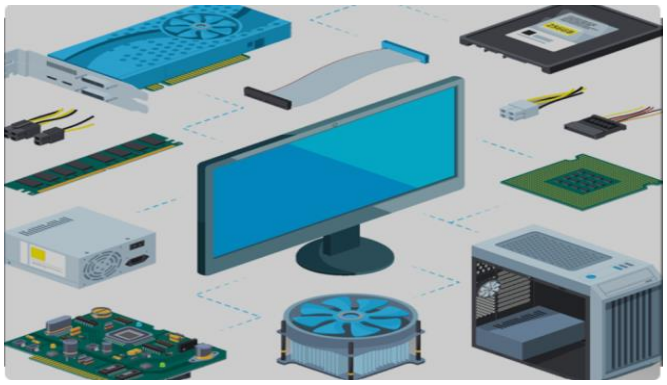
1. Environment Setup and Configuration Proficiency: Mastering the installation process, driver management, software compatibility troubleshooting, and system configuration (e.g., virtualization, dual-booting, networking) to create a stable and optimized development environment is essential. This objective develops skills in systems administration and configuration management which are applicable to various software development roles.

2. Understanding Operating System Concepts (Practical Application): The installation and use of Windows provide practical insight into core OS concepts like file systems, process management, memory management, user permissions, and the Windows Registry. Experimenting with system settings and troubleshooting issues will solidify theoretical knowledge from OS courses. This allows the student to bridge the gap between theory and practice.

3. Software Development and Debugging Experience on Windows: The primary objective is to have a working Windows environment where the student can install and use various development tools (IDEs, compilers, debuggers) specific to the Windows platform. The ability to build, test, debug, and deploy software on Windows is crucial for many software development jobs, and this hands-on experience is vital for building a portfolio and demonstrating practical skills to potential employers. This also involves understanding Windows-specific APIs and development methodologies.

3.0 Requirements

3.1 Hardware Requirements



- **Processor**

Minimum: 1.4 GHz 64-bit processor (x64 architecture).

Recommended: 2 GHz or faster 64-bit processor. Multi-core processors are recommended for better performance, especially for server workloads. Intel Xeon or AMD EPYC processors are common choices for servers.

- **RAM (Memory)**

Minimum: 2 GB.

Recommended: 4 GB for Server Core installations. 8 GB or more is strongly recommended for servers with the "Desktop Experience" (GUI) or for servers running demanding applications. More RAM is always better for server workloads.

- **Storage (Disk Space)**

Minimum: 32 GB of available disk space.

Recommended: 60 GB or more. Because sometimes user needs significantly more space for server roles, applications, data storage, and logs. 100GB or more is highly advisable.

Storage Type: SSD (Solid State Drive) is recommended for the operating system and applications for significantly improved performance. Traditional HDD (Hard Disk Drive) is acceptable for bulk data storage, but SSDs are far superior for responsiveness.

- **Network Adapter**

Requirement: Gigabit Ethernet adapter (10/100/1000baseT).

Considerations: A dedicated network interface card (NIC) is preferred for server workloads. Redundant NICs (teaming or bonding) are often used for high availability.

- **Display**

Requirement: Super VGA (1024 x 768) or higher resolution monitor.

Considerations: While a monitor is required for initial installation and troubleshooting, many servers are managed remotely after setup, so a dedicated monitor is not always necessary.

- **Optical Drive (or USB Support)**

Requirement: DVD drive (if installing from DVD media) or USB port (if installing from a bootable USB drive).

Considerations: Installing from a USB drive is generally faster than installing from DVD.

- **Firmware**

Requirement: UEFI 2.3.1c-based system and firmware that supports secure boot.

This may require some tweaking in the BIOS for it to find your drives

Considerations: Legacy BIOS systems are not fully supported. UEFI is the modern standard.

- **TPM (Trusted Platform Module)**

Requirement: TPM 2.0 is required for some security features, such as shielded VMs and some Windows Defender System Guard features.

Considerations: TPM 2.0 is increasingly important for server security.

3.2 Software Requirements

- **Installation Media**

Windows Server 2022 ISO file (downloaded from Microsoft Evaluation Center or other authorized source).

Bootable USB drive (created using Rufus or similar tool) or a physical DVD.

- **Virtualization Software (If Installing in a VM)**

VMware Workstation Player (free for personal use) or VMware vSphere ESXi (for enterprise virtualization).

Oracle VirtualBox (free and open-source).

Microsoft Hyper-V (built into Windows Server and Windows 10/11).

- **Tool for Creating Bootable USB (If Installing from USB)**

Rufus: https://rufus.ie/en/ (Recommended)

Windows USB/DVD Download Tool (older, but still functional)

- **Drivers (May be Required)**

Drivers for network adapters, storage controllers, and other hardware components may be needed if the built-in drivers are not sufficient. Download these from the hardware manufacturer's website.

For virtual machines, virtualization software typically provides virtualized drivers, so this is less of a concern.

- **Microsoft Account (Sometimes Required)**

A Microsoft account is often required for downloading the Windows Server ISO file and for activating the OS.

- **Internet Connection (Recommended)**

An internet connection is highly recommended for downloading updates, drivers, and for activating the OS. However, you can install without internet access and activate later.

- **Operating System on Host Machine (If Virtualizing)**

A host operating system (e.g., Windows 10/11, Linux) is required if you are installing Windows Server 2022 in a virtual machine.

4.0 Windows Server 2022 Installation Steps

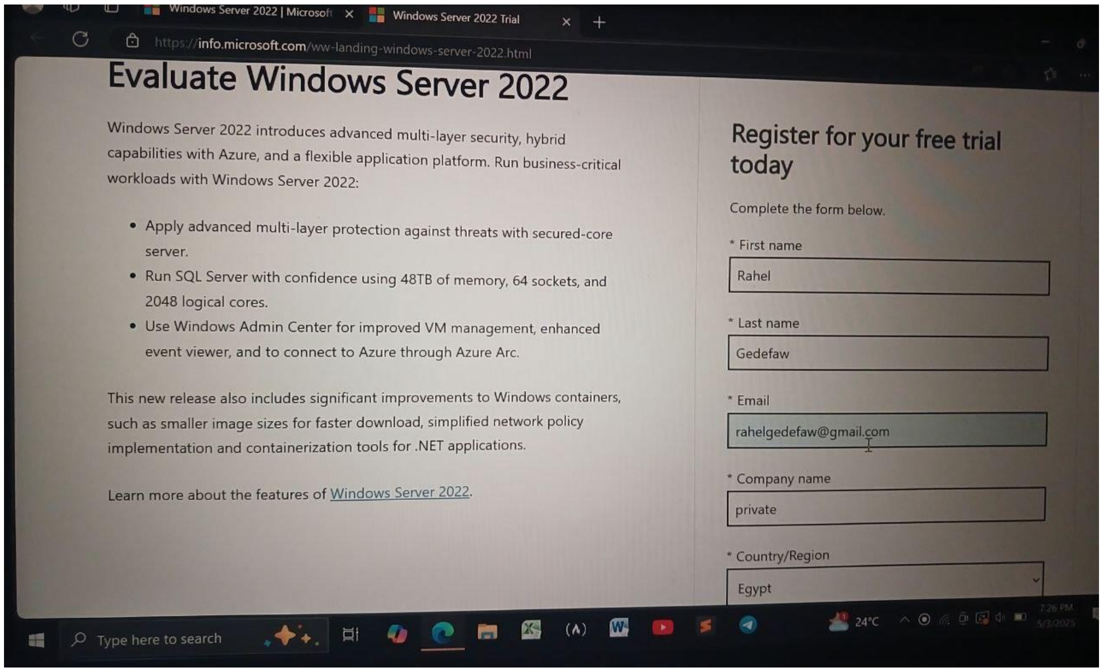
These steps are for installing **Windows Server 2022** in a virtual machine (VMware or Virtual Box) because it's the safest and most common approach for learning and experimentation. The core steps are similar for physical hardware, but driver management and hardware compatibility become more critical.

Here are steps

Step 1. Preparation:

- Verify Hardware: Ensure your host machine (if virtualizing) meets the minimum hardware requirements. Allocate sufficient resources to the VM (RAM, CPU cores, disk space).

- Download the ISO: Download the Windows Server 2022 ISO file from the Microsoft Evaluation Center.



Evaluate Windows Server 2022

Windows Server 2022 introduces advanced multi-layer security, hybrid capabilities with Azure, and a flexible application platform. Run business-critical workloads with Windows Server 2022:

- Apply advanced multi-layer protection against threats with secured-core server.
- Run SQL Server with confidence using 48TB of memory, 64 sockets, and 2048 logical cores.
- Use Windows Admin Center for improved VM management, enhanced event viewer, and to connect to Azure through Azure Arc.

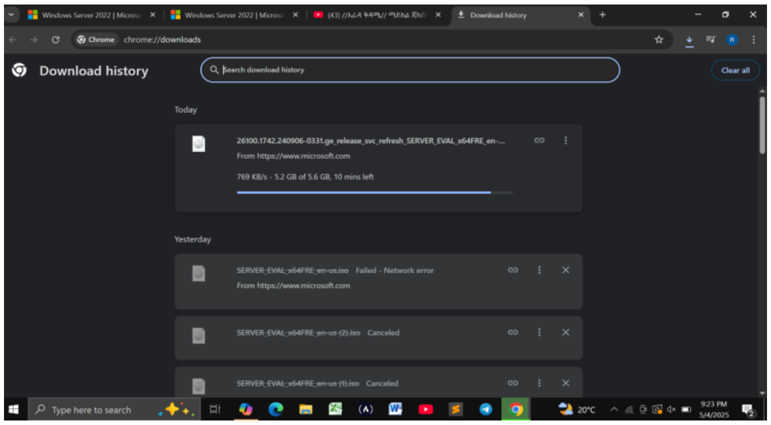
This new release also includes significant improvements to Windows containers, such as smaller image sizes for faster download, simplified network policy implementation and containerization tools for .NET applications.

Learn more about the features of Windows Server 2022.

Register for your free trial today

Complete the form below.

* First name	Rahel
* Last name	Gedefaw
* Email	rahelgedefaw@gmail.com
* Company name	private
* Country/Region	Egypt



Download history

Today

26100.1742.240906-0331.ge_release_svc_refresh_SERVER_EVAL_x64FRE_en-...
From https://www.microsoft.com
769 KB/s - 5.2 GB of 5.6 GB, 10 mins left

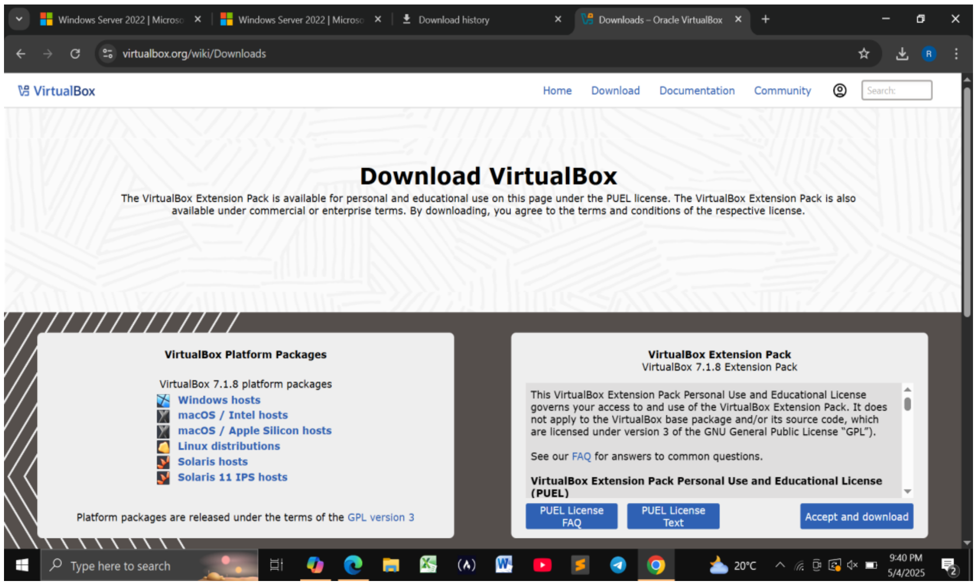
Yesterday

~~SERVER_EVAL_x64FRE_en-us.iso~~ Failed - Network error
From https://www.microsoft.com

~~SERVER_EVAL_x64FRE_en-us (2).iso~~ Canceled

~~SERVER_EVAL_x64FRE_en-us (1).iso~~ Canceled

- Virtual Machine Creation (If applicable): Create a new virtual machine in VMware or VirtualBox. Specify the ISO file as the bootable disk.


Windows Server 2022 | Microsc
Windows Server 2022 | Microsc
Download history
Downloads – Oracle VirtualBox
virtualbox.org/wiki/Downloads
VirtualBox
Home
Download
Documentation
Community
Search:
Type here to search
20°C
9:40 PM
5/4/2025

Download VirtualBox

The VirtualBox Extension Pack is available for personal and educational use on this page under the PUEL license. The VirtualBox Extension Pack is also available under commercial or enterprise terms. By downloading, you agree to the terms and conditions of the respective license.

VirtualBox Platform Packages

VirtualBox 7.1.8 platform packages
- Windows hosts
- macOS / Intel hosts
- macOS / Apple Silicon hosts
- Linux distributions
- Solaris hosts
- Solaris 11 IPS hosts

Platform packages are released under the terms of the GPL version 3

VirtualBox Extension Pack
VirtualBox 7.1.8 Extension Pack

This VirtualBox Extension Pack Personal Use and Educational License governs your access to and use of the VirtualBox Extension Pack. It does not apply to the VirtualBox base package and/or its source code, which are licensed under version 3 of the GNU General Public License "GPL").

See our FAQ for answers to common questions.

VirtualBox Extension Pack Personal Use and Educational License (PUEL)

Step 2. Start the Virtual Machine:

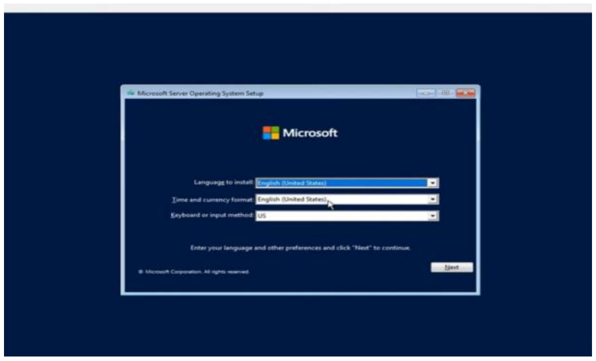
- Select your newly created VM in VMware or Virtual Box and click **"Start**" or "**Power On**."

Step 3. Boot from the Installation Media:

- The VM should automatically boot from the ISO file you specified. If it doesn't, you may need to adjust the boot order in the VM's BIOS settings (usually accessible by pressing a key like F2, Delete, or Esc during startup – the specific key varies by VM software and firmware).

Step 4. Windows Setup – Language, Time, and Keyboard

- Select your preferred language, time zone, and keyboard layout. Click "**Next."**



Language to install: English (United States)
Time and currency format: English (United States)
Keyboard or input method: US

Enter your language and other preferences and click "Next" to continue.

© Microsoft Corporation. All rights reserved.

Step 5. Install Now

- Click the "**Install Now**" button.

Step 6. Enter Product Key (or Skip)

- If you have a product key, enter it. If you're using the evaluation version, click **"I don't have a product key."**

Step 7. Select Operating System Edition (Crucial Step)

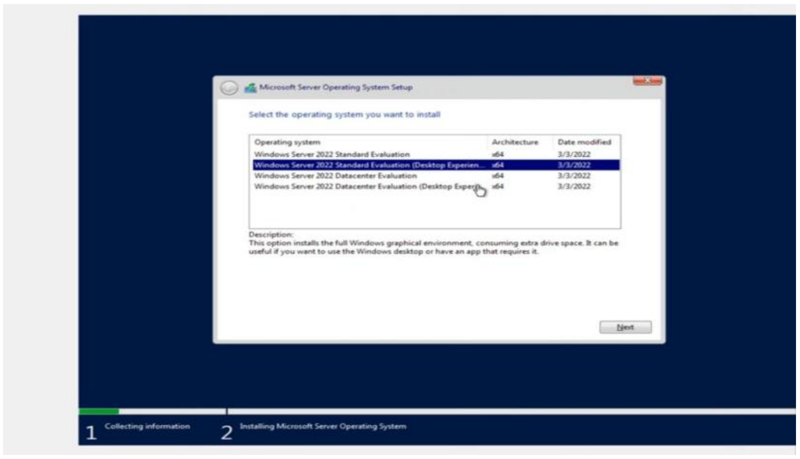
- Carefully select the correct edition of Windows Server 2022:

   * Windows Server 2022 Standard (Desktop Experience): This is generally the best choice for beginners. It includes the graphical user interface (GUI).

   * Windows Server 2022 Datacenter (Desktop Experience): This is the more advanced edition with features for large-scale deployments and virtualization. It also includes the GUI.

   * Windows Server 2022 Standard: (Without "Desktop Experience") This is the Server Core installation, which is command-line only. It has a smaller footprint and can be more secure but requires more expertise.

   * Windows Server 2022 Datacenter: (Without "Desktop Experience") This is the Server Core version of the Datacenter edition.


Microsoft Server Operating System Setup
Next
1 Collecting information
2 Installing Microsoft Server Operating System

Select the operating system you want to install

Operating system	Architecture	Date modified
Windows Server 2022 Standard Evaluation	x64	3/3/2022
Windows Server 2022 Standard Evaluation (Desktop Experien...	x64	3/3/2022
Windows Server 2022 Datacenter Evaluation	x64	3/3/2022
Windows Server 2022 Datacenter Evaluation (Desktop Exper...	x64	3/3/2022

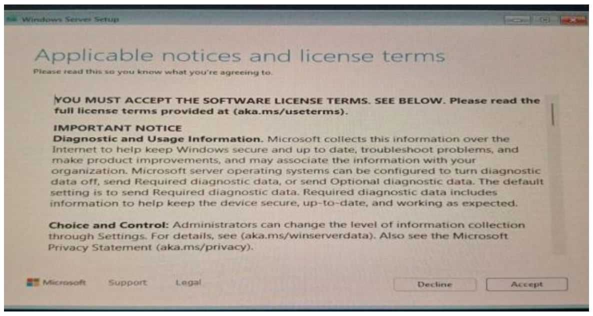
Description:
This option installs the full Windows graphical environment, consuming extra drive space. It can be useful if you want to use the Windows desktop or have an app that requires it.

For this project I prefer to select "Standard (Desktop Experience).

- Click **"Next**."

Step 8. Accept the License Terms

- Check the "**I accept the license terms**" box and click "**Next."**



Applicable notices and license terms

Please read this so you know what you're agreeing to.

YOU MUST ACCEPT THE SOFTWARE LICENSE TERMS. SEE BELOW. Please read the full license terms provided at (aka.ms/useterms).

IMPORTANT NOTICE
Diagnostic and Usage Information. Microsoft collects this information over the Internet to help keep Windows secure and up to date, troubleshoot problems, and make product improvements, and may associate the information with your organization. Microsoft server operating systems can be configured to turn diagnostic data off, send Required diagnostic data, or send Optional diagnostic data. The default setting is to send Required diagnostic data. Required diagnostic data includes information to help keep the device secure, up-to-date, and working as expected.

Choice and Control: Administrators can change the level of information collection through Settings. For details, see (aka.ms/winserverdata). Also see the Microsoft Privacy Statement (aka.ms/privacy).

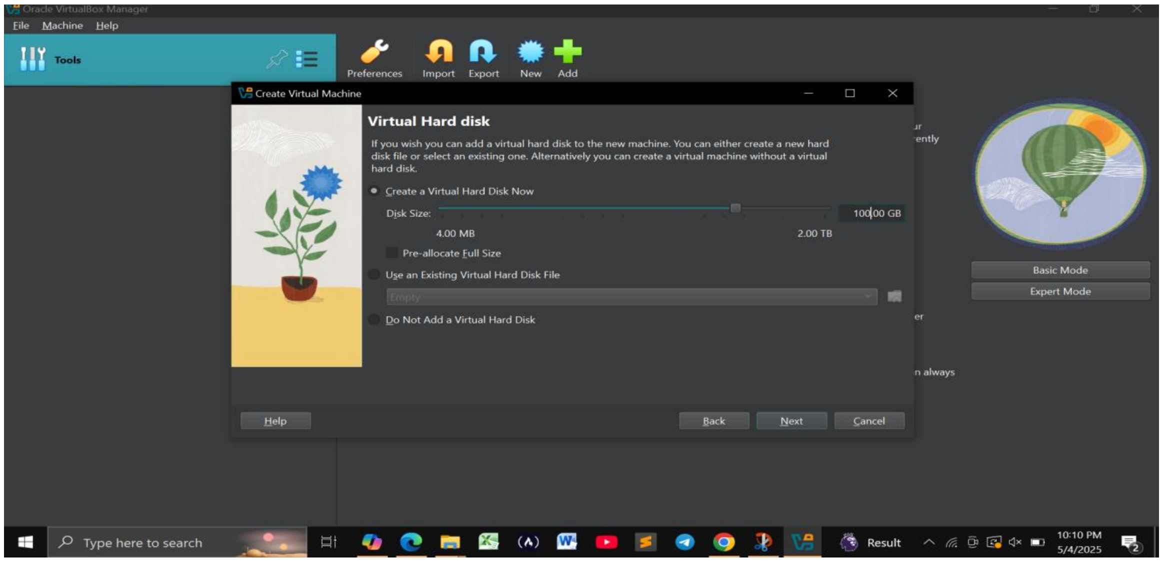
Step 9. Choose Installation Type

- Select "**Custom: Install Windows only (advanced)."** This option allows you to choose the partition where you want to install the OS.

Step 10. Where do you want to install Windows? – Disk Selection

- Select the virtual disk you created when setting up the VM. If the disk is unallocated, select it and click "**New**" to create a partition. Accept the default size (or adjust as needed) and click **"Apply."** Windows may create additional system partitions. Select the primary partition you created and click "Next."

- If no drives are listed, it could indicate a driver issue. For VMs, this is rare. For physical hardware, you might need to load a storage controller driver from a USB drive.



Virtual Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select an existing one. Alternatively you can create a virtual machine without a virtual hard disk.

● Create a Virtual Hard Disk Now

Disk Size: 100.00 GB

4.00 MB   2.00 TB

Pre-allocate Full Size

Use an Existing Virtual Hard Disk File

Empty

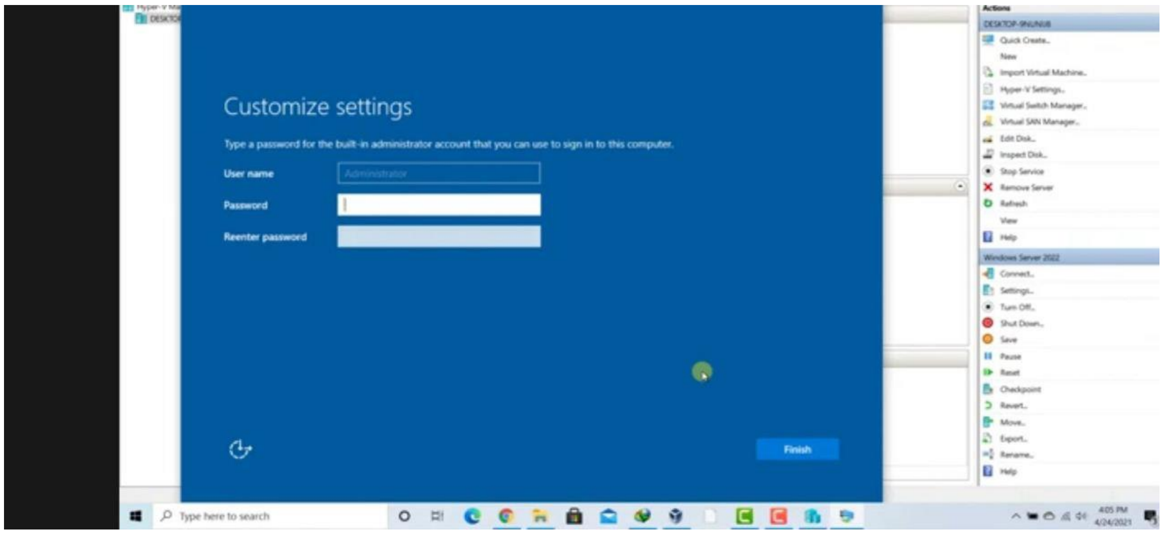
Do Not Add a Virtual Hard Disk

Step 11. Installing Windows

- The installation process will begin, copying files, installing features, and installing updates. This may take 20-40 minutes or longer, depending on your hardware. The VM will restart automatically several times during this process.

Step 12. Setting the Administrator Password

- After the final restart, the system will boot into the setup screen. You'll be prompted to set a password for the built-in "Administrator" account. Choose a strong password and remember it.


Customize settings

Type a password for the built-in administrator account that you can use to sign in to this computer.

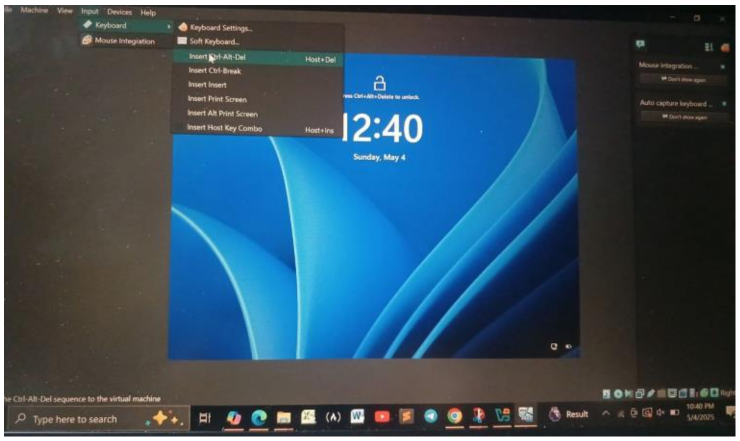
User name: Administrator

Password:

Reenter password:

Step 13. Login:

- Press **Ctrl+Alt+Delete (or Ctrl+Alt+Insert in some VMs)** to bring up the login screen. Log in as "Administrator" with the password you just set.



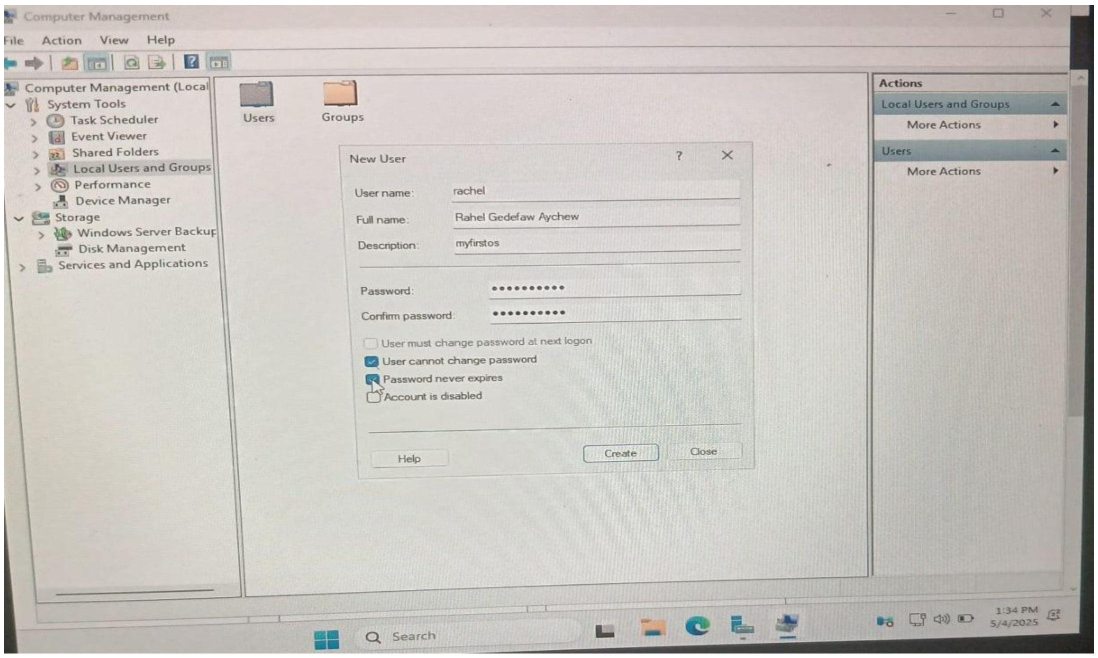
Then you can create a Local User Account:

Step 14.

- After logging in as Administrator, you should create a new local user account on the server, using your full legal name as the username . This is in addition to the built-in Administrator account.

4.1 Here is how to Create a Local User Account (After Installation):

- Open Computer Management: Right-click on the Start button and select "**Computer Management**."
- Expand "Local Users and Groups": In the left pane, expand "**Local Users and Groups**."
- Select "Users": Click on "**Users**."
- Create a New User: Right-click in the right pane and select "**New User**..."
- Fill in the Details:
  - **User name**: Enter your full name example (“Rahel Gedefaw”).
  - **Full name**: You can also enter your full name here.
  - **Description**: (Optional) but you can enter like ("Student Account").
  - **Password**: Enter a **strong** password.
  - Confirm password: Re-enter the password.
  - **User must change password at next logon**: Uncheck this box. For a test environment, it's usually easier to manage if you know the password.
  - User cannot change password: (Optional) Check this box if you don't want the user to be able to change their password.
- Password never expires: Check this box
- Account is disabled: Make sure this box is unchecked.



New User

User name: rachel

Full name: Rahel Gedefaw Aychew

Description: myfirstos

Password: ••••••••••

Confirm password: ••••••••••

- [ ] User must change password at next logon
- [✓] User cannot change password
- [✓] Password never expires
- [ ] Account is disabled

Help | Create | Close

• Click **"Create**" and then "**Close."**

• Add to Administrators Group (If Required): If you need the new account to have administrative privileges, double-click the new user account, go to the "Member Of" tab, click "Add...", type "Administrators" and click "OK."

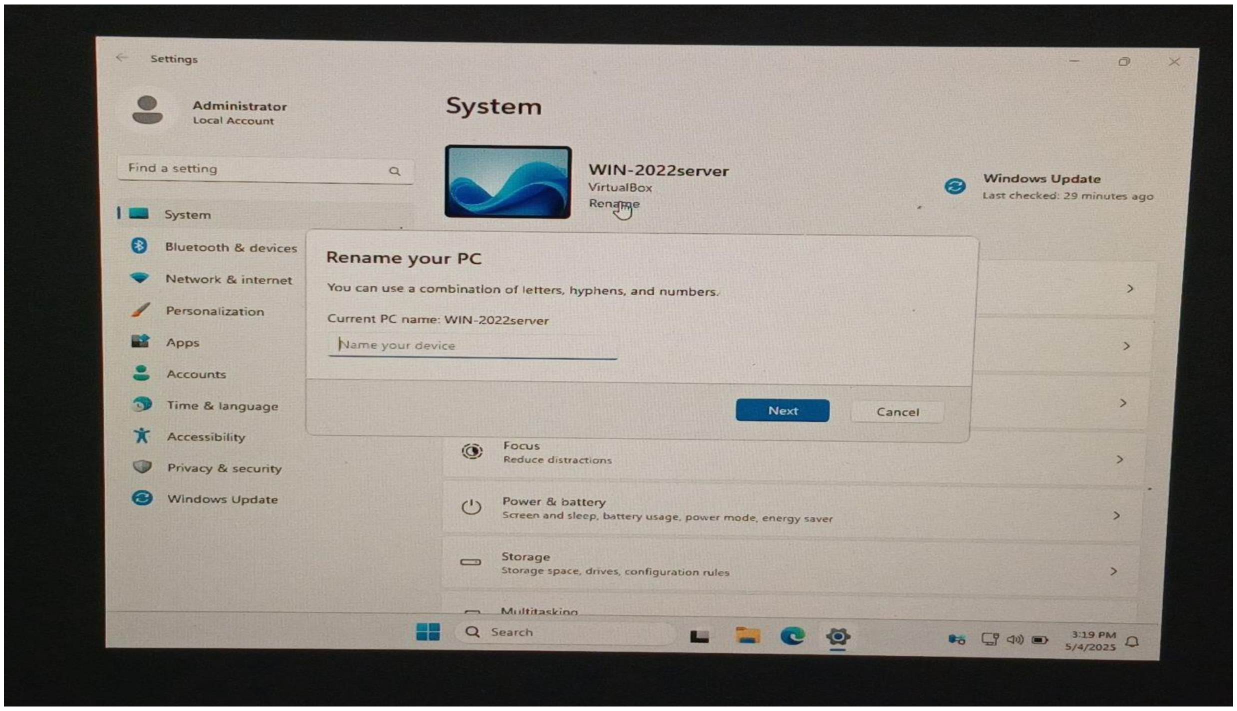
4.2 Post-Installation Configuration (Key Tasks):

1**. Set Static IP Address**: Assign a static IP address to the server. This is essential for servers to have a consistent address. In Server Manager, go to "Local Server" and click on "Ethernet" (or the name of your network adapter).

2. Install Windows Updates: In Server Manager, go to "Local Server" and check for updates. Install all available updates. This is crucial for security and stability.

3**. Rename the Computer**: In Server Manager, go to "Local Server" and change the computer name to something descriptive. Restart the server after renaming.

4. Configure Windows Firewall: Configure the Windows Firewall to allow necessary traffic for the services you plan to run.



Rename your PC

You can use a combination of letters, hyphens, and numbers.

Current PC name: WIN-2022server

Name your device

5. Install Server Roles and Features: Use Server Manager to add the roles and features required for your assignment (e.g., Active Directory Domain Services, DNS Server, Web Server (IIS)).

6. Create Local User Account (with Your Full Name): As described above.

By following these steps carefully and paying attention to the details, you should be able to successfully install Windows Server 2022 and create the required user account.

5.0 Issues or problem faced

No issues or problems are faced b/c I have used virtual box to download this OS and It is safest method to protect my operating system

6.0 Key File Systems for Windows Server 2022

1. NTFS (New Technology File System):

- Support Level: Native and Primary File System. We should use it because:

Security: NTFS provides robust security features, including access control lists (ACLs) that allow you to control which users and groups have access to files and folders. This is essential* for a server environment.

Reliability: NTFS is a journaling file system, which means that it keeps a log of changes to the file system. If the system crashes, NTFS can use the journal to recover the file system to a consistent state.

Large File and Partition Sizes: NTFS supports very large files and partitions (theoretically up to 16 exabytes).

Compression: NTFS supports file and folder compression, which can save disk space.

Encryption: NTFS supports encryption using Encrypting File System (EFS), which protects sensitive data from unauthorized access.

Quotas: NTFS supports disk quotas, which allow you to limit the amount of disk space that users can use.

Active Directory Integration: NTFS is tightly integrated with Active Directory, making it easy to manage file and folder permissions in a domain environment.

- Usage: NTFS is the default file system for the system partition (where Windows Server is installed) and is generally recommended for all internal hard drives in a Windows Server system.

2. ReFS (Resilient File System):

- Support Level: Supported, with specific use cases.

The following are its qualities:

Data Integrity: ReFS is designed to maximize data availability, despite errors that would historically cause data loss. ReFS detects when data becomes corrupt and then proactively repairs that corruption.

Resilience: ReFS is more resilient to data corruption than NTFS. It uses checksums to detect data corruption and can automatically repair corrupted data from redundant copies.

Availability: If corruption occurs, ReFS stays online. The file system proactively detects when data becomes corrupt and then proactively repairs that corruption, so there is no need to take down the volume.

Scalability: ReFS is designed to handle very large volumes and files.

- Usage: ReFS is primarily intended for:

Storage Spaces Direct (S2D): ReFS is the recommended file system for Storage Spaces Direct deployments, which are used for creating highly available and scalable storage solutions.

Backup Storage: ReFS is a good choice for backup storage due to its data integrity features.

Virtualization: ReFS is optimized for virtualization workloads, such as storing virtual machine files.

3. FAT32:

- Support Level: Read-only support for compatibility.

Its limitations are:

Limited File Size: FAT32 has a maximum file size limit of 4GB, which is too small for many modern applications and data files.

No Security: FAT32 does not support ACLs or other security features.

No Reliability Features: FAT32 does not have journaling or other reliability features.

- Usage: FAT32 is typically used for USB flash drives and other removable media for compatibility with older systems. Windows Server 2022 can read FAT32 volumes, but you should not use FAT32 for internal drives or volumes containing important data.

4. **exFAT:**

- Support Level: Supported for removable media.

Some qualities are:

Larger File Sizes: exFAT supports larger file sizes than FAT32 (no 4GB limit).

Compatibility: exFAT is compatible with Windows, macOS, and Linux.

Some limitations to use it Internally:

No Security: exFAT does not support ACLs or other security features.

No Reliability Features: exFAT does not have journaling or other reliability features.

- Usage: exFAT is commonly used for USB flash drives, SD cards, and other removable media, especially when you need to store files larger than 4GB.

4. **ext4 (Linux File System)**

- Support Level: Limited support (read-only through third-party tools).

These are some advantages:

Not Native to Windows: ext4 is the primary file system for many Linux distributions, but it's not natively supported by Windows Server.

No Native Write Support: You can use third-party tools to read* ext4 partitions from Windows, but writing to ext4 partitions is generally not recommended due to potential compatibility issues.

- Usage: If you need to access files on an ext4 partition from Windows Server, you can use a third-party tool like Ext2Fsd. However, it's generally better to transfer the files to an NTFS or ReFS partition.

5. Btrfs (Linux File System)

- Support Level: Very limited (read-only through third-party tools).

Same reasons as ext4. It is not native to windows.

6. ZFS (Advanced File System - Originally from Sun/Oracle)

- Support Level: Very limited (read-only through third-party tools).
- Why Not Use ZFS? ZFS is a powerful file system with advanced features like data integrity, snapshots, and volume management, but it's not natively supported by Windows Server. There are limited third-party tools to access ZFS partitions from Windows, but they are not widely used or recommended for production environments.

7. HFS+ and APFS (macOS File Systems):

- Support Level: Very limited (read-only through third-party tools).

• Do not Use HFS+/APFS? These are the file systems used by macOS, and they are not natively supported by Windows Server. You can use third-party tools to read HFS+ or APFS partitions from Windows, but writing to them is generally not recommended.

6.1 Summary and Recommendations

• For the System Partition (where Windows Server is installed): Always use NTFS.

• For Internal Hard Drives (Data Storage): Use NTFS unless you have a specific reason to use ReFS (e.g., Storage Spaces Direct or backup storage).

• For Removable Media (USB Drives, SD Cards): exFAT is a good choice for compatibility and larger file sizes. FAT32 can be used for maximum compatibility with older systems, but it has file size limitations.

• Avoid Using Non-Native File Systems (ext4, Btrfs, ZFS, HFS+, APFS) for Internal Drives. If you need to access data on these file systems, use third-party tools to read the data, but transfer it to an NTFS or ReFS partition for long-term storage and use on Windows Server.

7.0 Advantages and Disadvantages of Microsoft Windows Server OS

7.1 Advantages

• Ease of Use (with Desktop Experience): The familiar Windows GUI makes it relatively easy for administrators with Windows desktop experience to manage the server.

• Wide Software Compatibility: Supports a vast ecosystem of applications, including both Microsoft products and third-party software.

• Active Directory Integration: Streamlines user and resource management, security policies, and network administration in a domain environment.

• Hyper-V Virtualization: Built-in virtualization technology allows efficient resource utilization and consolidation of workloads.

• Strong Hardware Support: Compatible with a wide range of server hardware from various vendors.

• Security Features: Robust security features, including Windows Defender, firewall, and access control lists (ACLs), help protect against threats.

• Azure Integration: Seamless integration with Microsoft Azure cloud services for hybrid cloud scenarios.

• PowerShell Automation: PowerShell provides powerful scripting capabilities for automating administrative tasks.

• Large Community Support: A large and active community of users and developers provides ample resources for troubleshooting and support.

7.2 Disadvantages

• Cost: Windows Server licenses can be expensive, especially for larger deployments or Datacenter editions.

• Resource Intensive (with Desktop Experience): The GUI-based "Desktop Experience" consumes significant system resources, which can impact performance. Server Core installation addresses this but requires more command-line expertise.

• Security Vulnerabilities: Like any complex OS, Windows Server is subject to security vulnerabilities that require regular patching and updates.

• Reboots Required: Many configuration changes and updates require system reboots, which can cause downtime.

• Vendor Lock-in: Reliance on Microsoft technologies can lead to vendor lock-in.

• Command-Line Complexity (Server Core): Managing Server Core installations requires strong command-line skills, which can be a barrier for some administrators.

• Licensing Complexity: Windows Server licensing can be complex and difficult to understand.

7.3 Conclusion

Microsoft Windows Server OS remains a powerful and versatile platform for managing networks, hosting applications, and delivering IT services. Its strengths lie in its ease of use (with the Desktop Experience), wide software compatibility, and Active Directory integration.

However, the cost, resource intensiveness, and security vulnerabilities are important considerations. The choice of Windows Server depends on the specific needs and budget of the organization.

7.4 Future Recommendations

Embrace Server Core: For experienced administrators, transitioning to Server Core installations is highly recommended to reduce resource consumption and improve security.

• Explore Azure Stack HCI: For hyperconverged infrastructure, consider Azure Stack HCI as a modern alternative to traditional Windows Server deployments.

• Automate Everything: Invest in PowerShell scripting and automation tools to streamline administrative tasks and reduce manual errors.

• Prioritize Security: Implement a comprehensive security strategy, including regular patching, vulnerability scanning, and intrusion detection.

• Consider Containerization: Explore containerization technologies like Docker and Kubernetes to improve application portability and scalability.

• Cloud Skills: Develop expertise in cloud technologies and Azure services to leverage the benefits of hybrid cloud deployments.

• Keep Learning: Stay up-to-date with the latest Windows Server features, best practices, and security threats. The IT landscape is constantly evolving.

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

What is Virtualization

Virtualization is the technology that allows you to create virtual instances of computer resources, such as operating systems, servers, storage devices, or network components. It abstracts the physical hardware, allowing multiple virtual machines (VMs) to run concurrently on a single physical machine. Each VM operates as if it has its own dedicated hardware, even though it's sharing the underlying physical resources.

Why Virtualization is Important (The Benefits)

• Resource Optimization: Maximizes the utilization of physical hardware by running multiple VMs on a single server. This reduces hardware costs, power consumption, and datacenter footprint.

• Cost Savings: Reduces capital expenditures (CAPEX) on hardware and operating expenses (OPEX) on power, cooling, and maintenance.

• Isolation and Security: VMs are isolated from each other, which enhances security. If one VM is compromised, it doesn't affect other VMs on the same physical machine.

• Flexibility and Agility: Makes it easy to provision and deploy new VMs quickly. You can easily create snapshots and backups of VMs for disaster recovery.

• Testing and Development: Provides a safe and isolated environment for testing new software or configurations without affecting production systems.

• Legacy Application Support: Allows you to run older operating systems and applications that may not be compatible with modern hardware.

• Simplified Management: Centralized management tools make it easier to manage and monitor VMs.

• Disaster Recovery: VMs can be easily replicated to a secondary site for disaster recovery purposes.

How Virtualization Works (The Key Components)

1. Hypervisor (Virtual Machine Monitor - VMM)

The core component of virtualization. It's a software layer that sits between the physical hardware and the VMs. The hypervisor's job is to:

- Abstract the Hardware: Presents a virtualized view of the hardware to each VM.
- Allocate Resources: Allocates CPU, memory, storage, and network resources to each VM.
- Manage VM Execution: Schedules the execution of VMs on the physical CPU.
- Isolate VMs: Enforces isolation between VMs to prevent interference.

Lets go further to its types:

Types of Hypervisors:

- **Type 1 (Bare-Metal Hypervisors)**: Run directly on the hardware, without a host operating system. Examples: VMware ESXi, Microsoft Hyper-V Server (free version).
- **Type 2 (Hosted Hypervisors)**: Run on top of a host operating system. Examples: VMware Workstation, Oracle VirtualBox, Parallels Desktop.

2. Virtual Machines (VMs)

Each VM consists of:

- Virtual Hardware: A virtual representation of the hardware resources (CPU, memory, disk, network adapter) that the VM uses.
- Guest Operating System: The operating system that runs inside the VM (e.g., Windows Server, Linux).
- Applications: The applications that run on the guest OS.

3. Hardware-Assisted Virtualization (Intel VT-x/AMD-V)

Modern CPUs include hardware features that improve the performance of virtualization. These features allow the hypervisor to directly access and control the CPU, memory, and other hardware resources.

The Basic Process

1. The hypervisor is installed on the physical hardware (or on top of a host OS).

2. VMs are created and configured, specifying the virtual hardware resources they need.

3. The hypervisor allocates resources to the VMs.

4. When a VM needs to access the hardware, the hypervisor intercepts the request and translates it into a physical hardware operation.

5. The hypervisor schedules the execution of VMs on the physical CPU, ensuring that each VM gets its fair share of resources.

In summary, virtualization is a powerful technology that enables efficient resource utilization, isolation, and flexibility. It's a fundamental building block for modern cloud computing and datacenter environments

***Implementation of system call accept()

The accept() system call is a critical function in server-side networking, enabling a server to handle incoming client connections. When a server creates a listening socket and calls listen(), it's essentially advertising its willingness to accept connections. The accept() system call is then used to actually accept those connections and establish communication with individual clients. Its primary purpose is to take a connection request from the listening socket's queue (if one exists) and create a new, dedicated socket for communicating with that client. The original listening socket remains open to accept further connections. The new socket provides an isolated channel for the server to exchange data with a particular client.

The core steps can be summarized as follows:

1. **System Call Entry Point**: The process begins with a system call handler function within the kernel. This function acts as the entry point when a user-space program invokes the accept() system call (e.g., through a libc wrapper function).

2. **Socket Validation**: A crucial first step is to validate the file descriptor (sockfd) passed as an argument. The kernel must verify that this descriptor is valid, that it refers to a socket, and, crucially, that the socket is in the LISTENING state. This ensures that the system call is being invoked on a socket that is actually prepared to accept connections. The kernel uses internal data structures to track file descriptors and their associated resources.

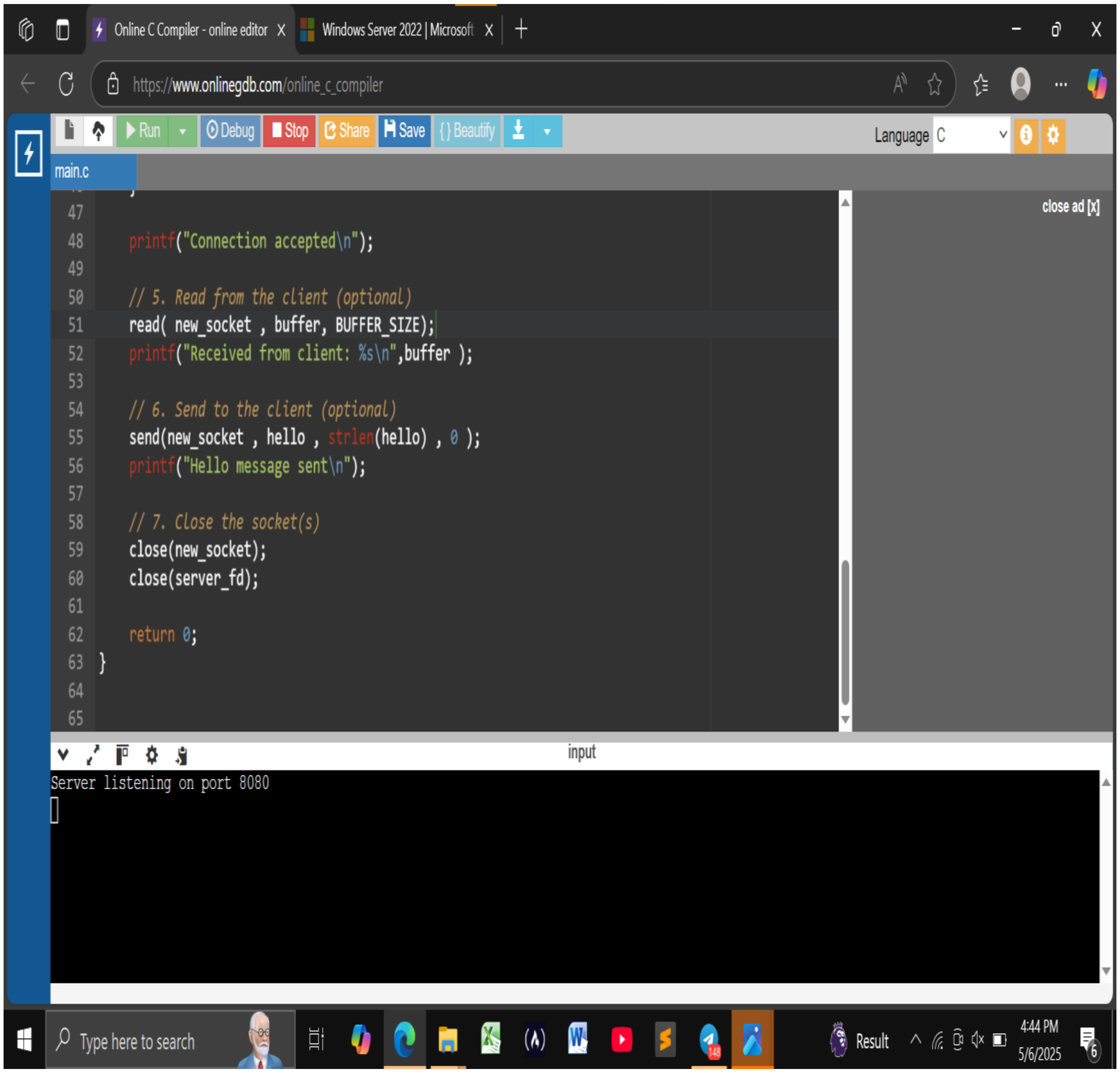
3. **Connection Handling (Blocking and Non-Blocking)**: This is where the core complexity arises. Ideally, accept() should block (wait) if there are no pending connections on the listening socket. This involves putting the calling process into a waiting state, adding it to a wait queue associated with the listening socket, and then invoking the kernel's scheduler to switch to another runnable process. When a connection request arrives (signaled by a network interrupt), the waiting process needs to be woken up and removed from the wait queue. In a non-blocking scenario (e.g., if the socket is set to non-blocking

mode), accept() would return immediately with an error (e.g., EAGAIN or EWOULDBLOCK) if no connections are pending.

4. **New Socket Creation**: Upon receiving a connection (or when implementing a non-blocking version), the kernel must create a new socket to represent the connection with the specific client. This involves allocating a new socket data structure, initializing its state (source and destination IP addresses, ports), and linking it to the listening socket.

5. **File Descriptor Assignment:** The new socket needs a file descriptor so the user-space process can interact with it. The kernel must allocate an available file descriptor and associate it with the new socket.

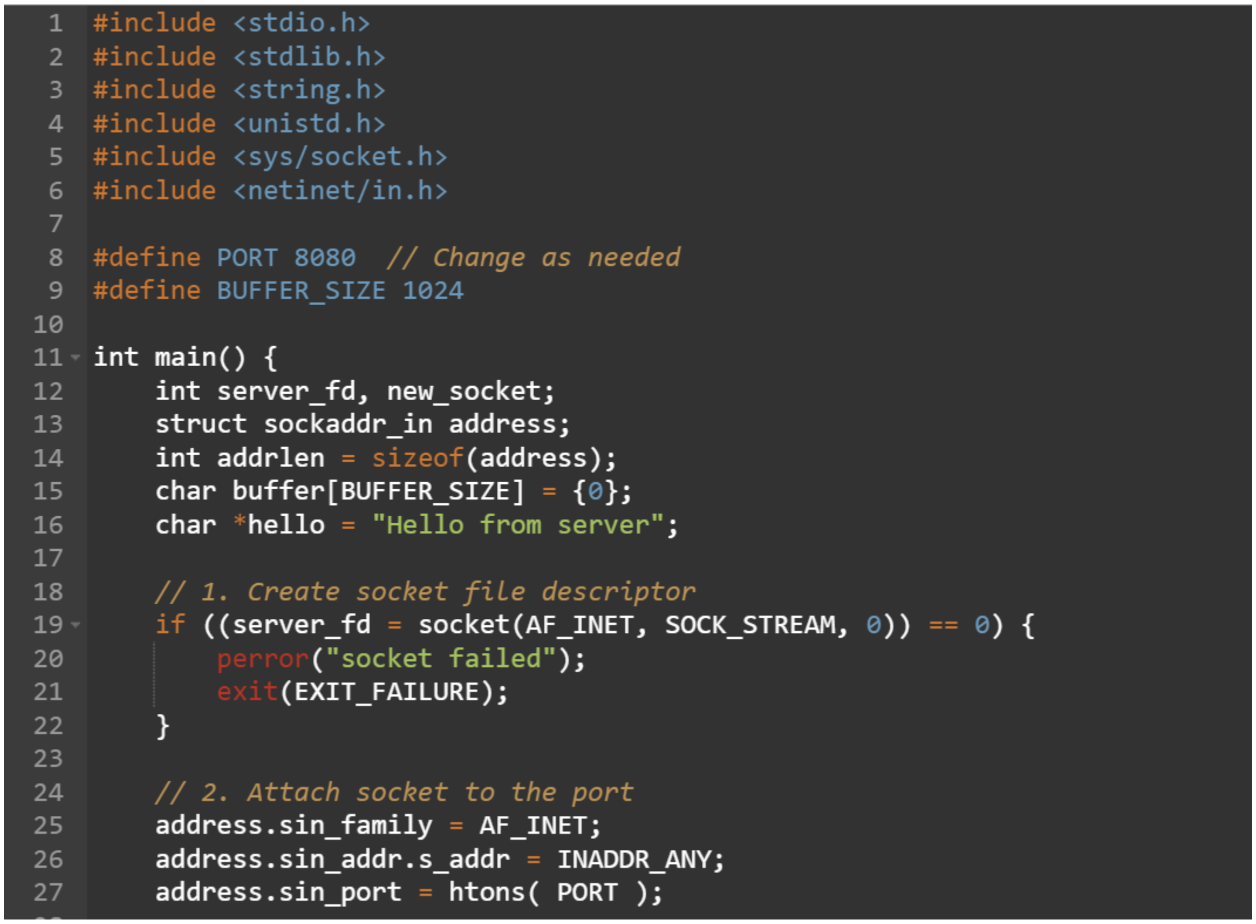
6. **Return Value: Finally**, the system call handler returns the new file descriptor to the user-space program.


Online C Compiler - online editor
Windows Server 2022 | Microsoft
https://www.onlinegdb.com/online_c_compiler
Run
Debug
Stop
Share
Save
{ } Beautify
Language C
main.c
close ad [x]
input
Type here to search
Result
4:44 PM
5/6/2025

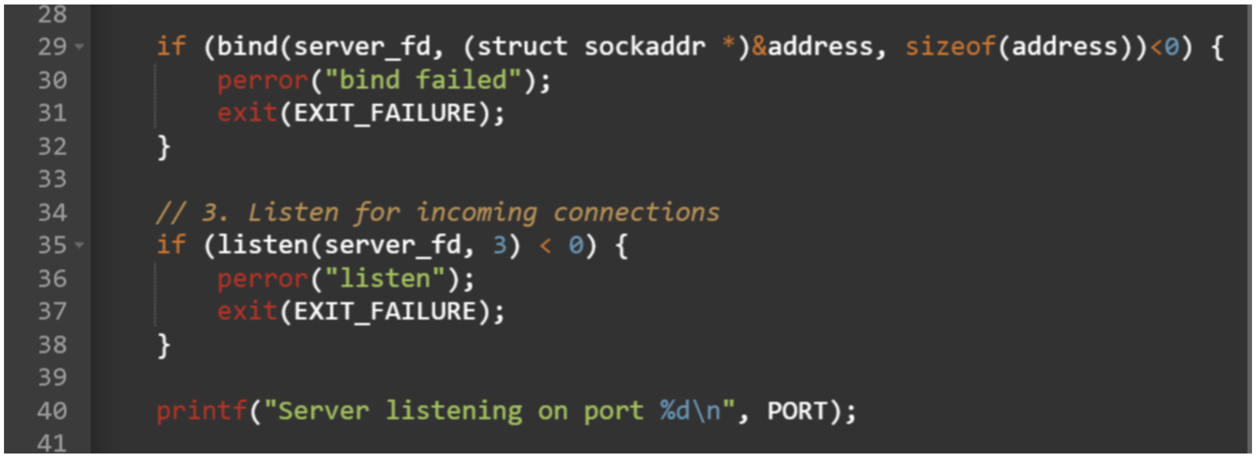
```c
    printf("Connection accepted\n");

    // 5. Read from the client (optional)
    read( new_socket , buffer, BUFFER_SIZE);
    printf("Received from client: %s\n",buffer );

    // 6. Send to the client (optional)
    send(new_socket , hello , strlen(hello) , 0 );
    printf("Hello message sent\n");

    // 7. Close the socket(s)
    close(new_socket);
    close(server_fd);

    return 0;
}
```

```
Server listening on port 8080
```

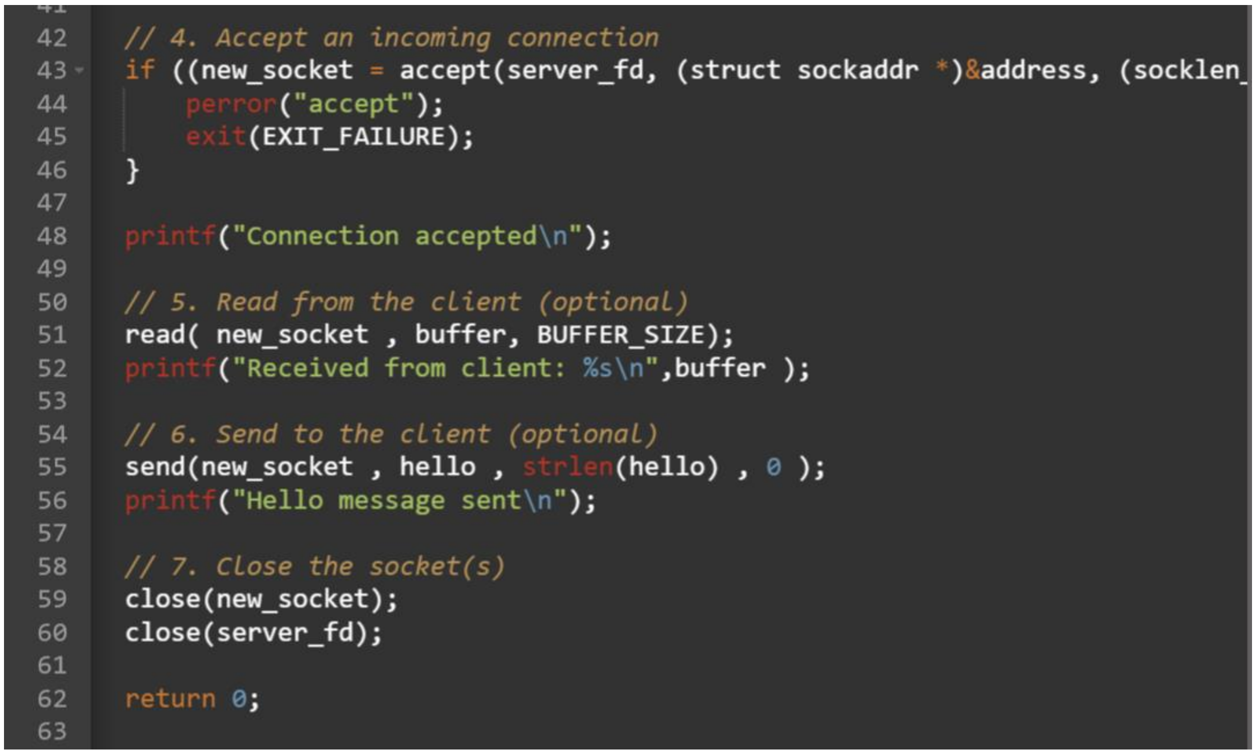
Here is full implementation



```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/socket.h>
#include <netinet/in.h>

#define PORT 8080  // Change as needed
#define BUFFER_SIZE 1024

int main() {
    int server_fd, new_socket;
    struct sockaddr_in address;
    int addrlen = sizeof(address);
    char buffer[BUFFER_SIZE] = {0};
    char *hello = "Hello from server";

    // 1. Create socket file descriptor
    if ((server_fd = socket(AF_INET, SOCK_STREAM, 0)) == 0) {
        perror("socket failed");
        exit(EXIT_FAILURE);
    }

    // 2. Attach socket to the port
    address.sin_family = AF_INET;
    address.sin_addr.s_addr = INADDR_ANY;
    address.sin_port = htons( PORT );
```



```c

    if (bind(server_fd, (struct sockaddr *)&address, sizeof(address))<0) {
        perror("bind failed");
        exit(EXIT_FAILURE);
    }

    // 3. Listen for incoming connections
    if (listen(server_fd, 3) < 0) {
        perror("listen");
        exit(EXIT_FAILURE);
    }

    printf("Server listening on port %d\n", PORT);

```



```
// 4. Accept an incoming connection
if ((new_socket = accept(server_fd, (struct sockaddr *)&address, (socklen_
    perror("accept");
    exit(EXIT_FAILURE);
}

printf("Connection accepted\n");

// 5. Read from the client (optional)
read( new_socket , buffer, BUFFER_SIZE);
printf("Received from client: %s\n",buffer );

// 6. Send to the client (optional)
send(new_socket , hello , strlen(hello) , 0 );
printf("Hello message sent\n");

// 7. Close the socket(s)
close(new_socket);
close(server_fd);

return 0;
```

THANK YOU!