Contents

1. Minix Operating System

1.1. Introduction and Motivation

An operating system is the backbone of any computing device, serving as the interface between the hardware and the user. It manages hardware resources, provides essential services, and enables the execution of various applications, making computing accessible and efficient. It is responsible for critical tasks, such as: Resource Management, Process Management, File Management and User Interface. It can be of various types to serve different purposes. For example:

- General-Purpose OS: Like Windows, macOS, and Linux that are designed for personal or professional use.
- Mobile OS: Like Android and iOS are used for mobile devices.
- Embedded OS: Specialized OS for embedded systems found in appliances, cars, and IoT devices.
- Real-Time OS (RTOS): Designed for systems that require immediate processing, such as industrial robots or medical devices.

Here we will focus on Minix. Minix is a small, **UNIX-like** operating system originally developed as a teaching tool by Andrew S. Tanenbaum in 1987. It was created to demonstrate the principles of operating systems and is known for its **microkernel** architecture, which emphasizes modularity and simplicity. Minix served as an inspiration for the development of Linux, and its design is widely regarded for its educational value. The motivation behind Minix was to provide an open, accessible system for students and developers to learn and experiment with operating system concepts.

1.1. Objectives

Minix was created with clear objectives in mind. It was primarily designed as an educational tool to teach operating system concepts and provide practical experience in understanding OS structures. By making the source code accessible, Tanenbaum ensured that students and developers could explore, modify, and learn from the OS's inner workings.

Our main objectives for installing Minix OS in a virtual environment are:

- ✓ Understanding Minix OS Features:
  - Gaining experience with the Minix Operating System
  - Explore its educational uses in teaching operating systems concepts.

- ✓ Experimentation in Virtual Environments:
  - We will use the virtual tool VMware Workstation to safely set up and experiment with Minix OS, without affecting the physical hardware.
  - Learn about resource allocation and compatibility within virtual setups.

- ✓ Filesystem Exploration:
  - Study the Minix Operating System's support for different file systems and understand its default file management system, known as the **MINIX file system**.

- ✓ Skill Development:
  - Enhance technical expertise in installing and configuring operating systems in virtual environments.
  - Troubleshoot issues and document solutions effectively.

- ✓ Practical Application:
  - ✓ It will help us prepare for real-world scenarios where microkernel-based operating systems may be useful (e.g., embedded systems, research).

1.3. Requirements

1.3.1 Hardware

To run Minix OS in a virtual environment, the following hardware requirements are recommended:

- ✓ Processor: A modern x86-64 processor with virtualization support (Intel VT-x or AMD-V).
- ✓ RAM: Minimum 512 MB (1 GB or more is preferable for smoother performance).
- ✓ Storage: At least 1 GB of available disk space.
- ✓ Graphics: Basic VGA-compatible graphics hardware is sufficient.
- ✓ Network: Internet connectivity is optional but useful for updates or additional resources and packages.

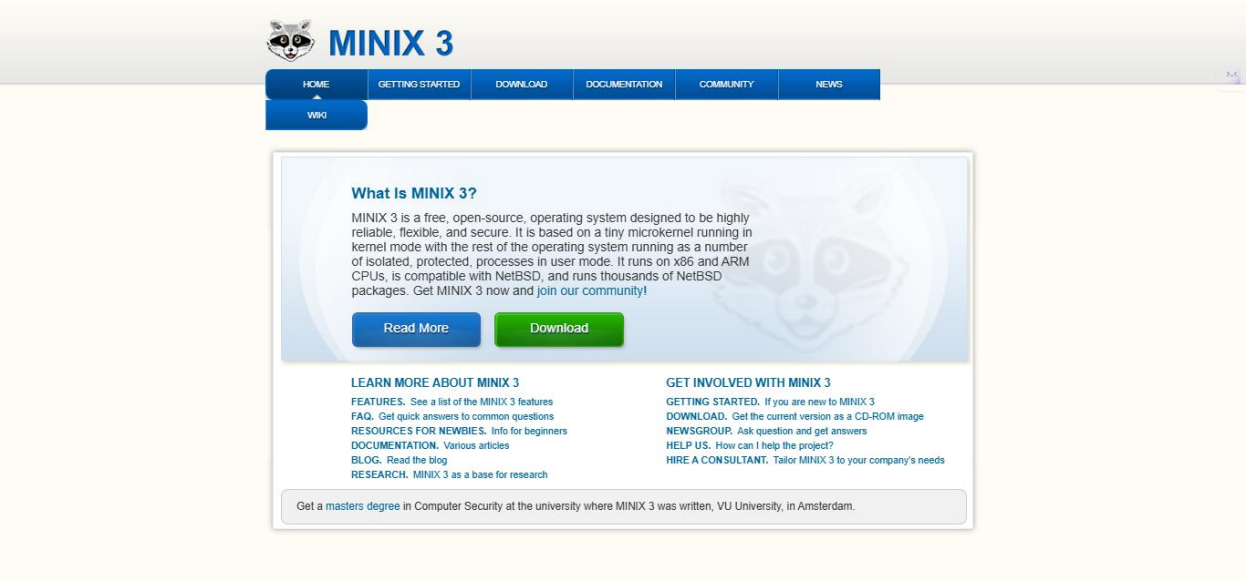
1.3.2 Software

- ➢ Virtualization Software:
  - ✓ VMware Workstation or VMware Player
- ➢ Operating System (ISO):
  - ✓ Minix OS ISO file, which can be downloaded from the official websit.
- ➢ Host OS:
  - ✓ A supported operating system to run the virtualization software (e.g., Windows, macOS, Linux).
- ➢ Filesystem:
  - ✓ Support for the filesystem used by Minix OS (e.g., ext2/ext3, as Minix OS uses its own MINIX file system).

1.4. Installation steps

- Go to the official Minix website (https://www.minix3.org/) and download the latest ISO image file
- Download and Install VMware
- Launch the application and create a new virtual machine
- Assign the hardware resources
- Start the Virtual Machine and load the Minix OS installer
- Follow Installation Prompts
- Use the installer to set up Minix OS
- Partition the virtual disk (follow the Minix setup instructions for filesystem selection)
- Create a user account with your full name
- Verify Installation
- Once the setup is complete, restart the virtual machine
- Ensure that Minix OS is installs correctly boots correctly then an account can be created through the admin.

❖ In the next part we will see screenshots of most of the steps involved:



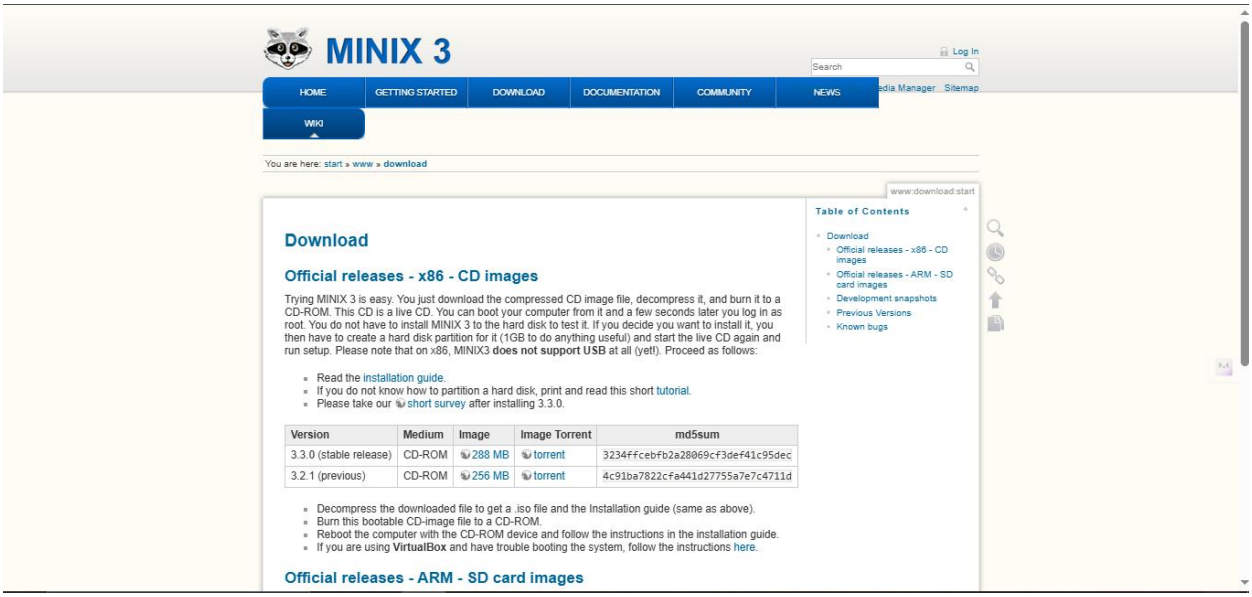


Image-set 1. Downloading Minix's ISO file from their official website

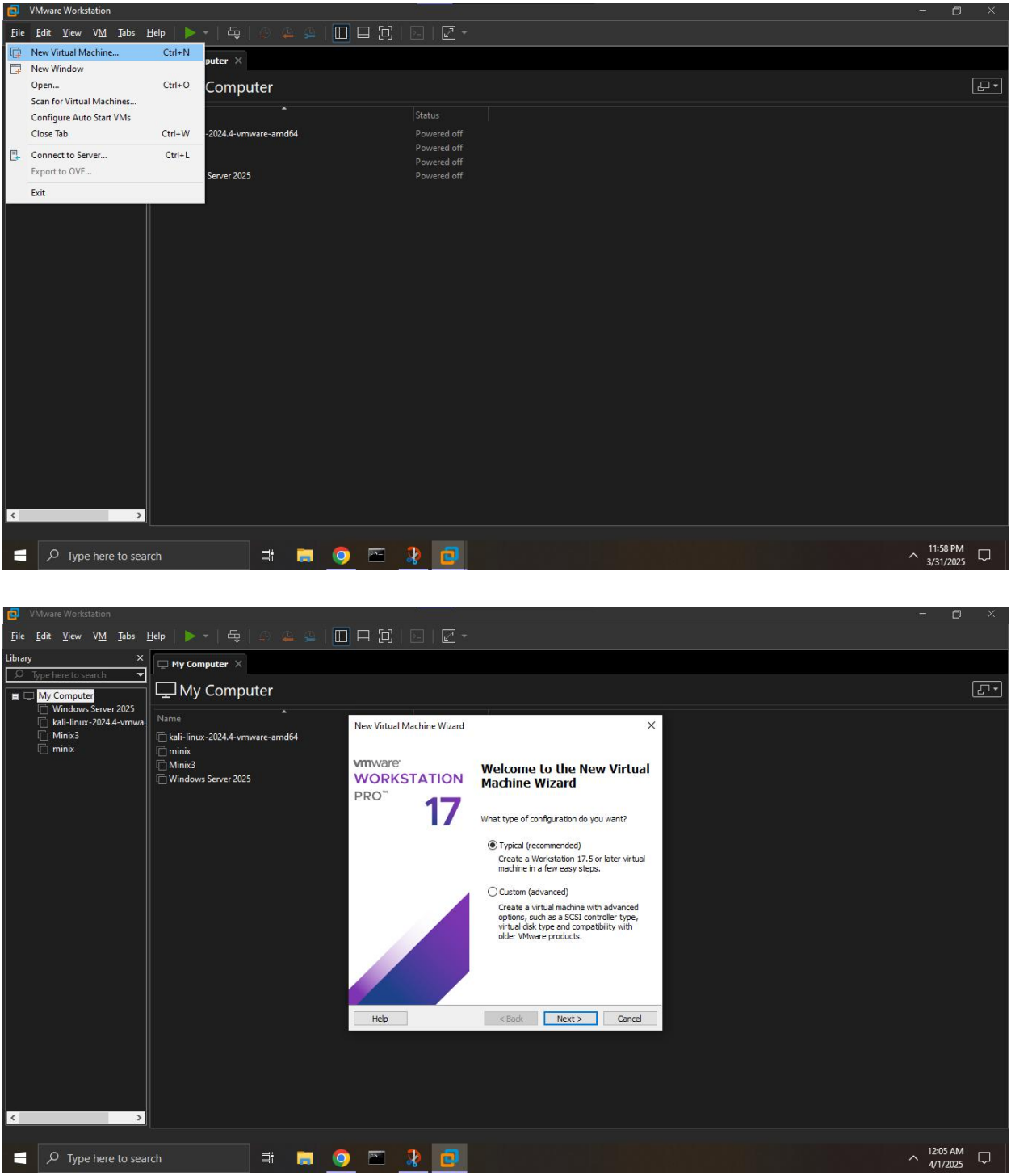
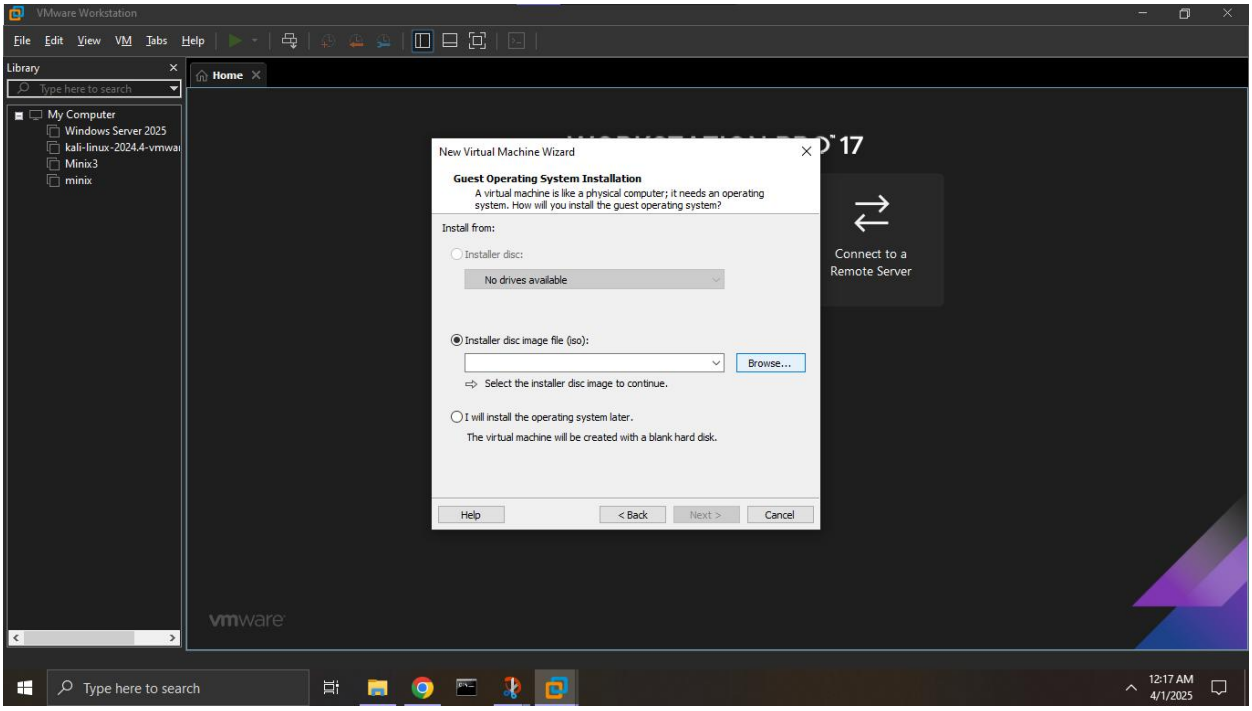


Image-set 2. Setting up the virtual machine



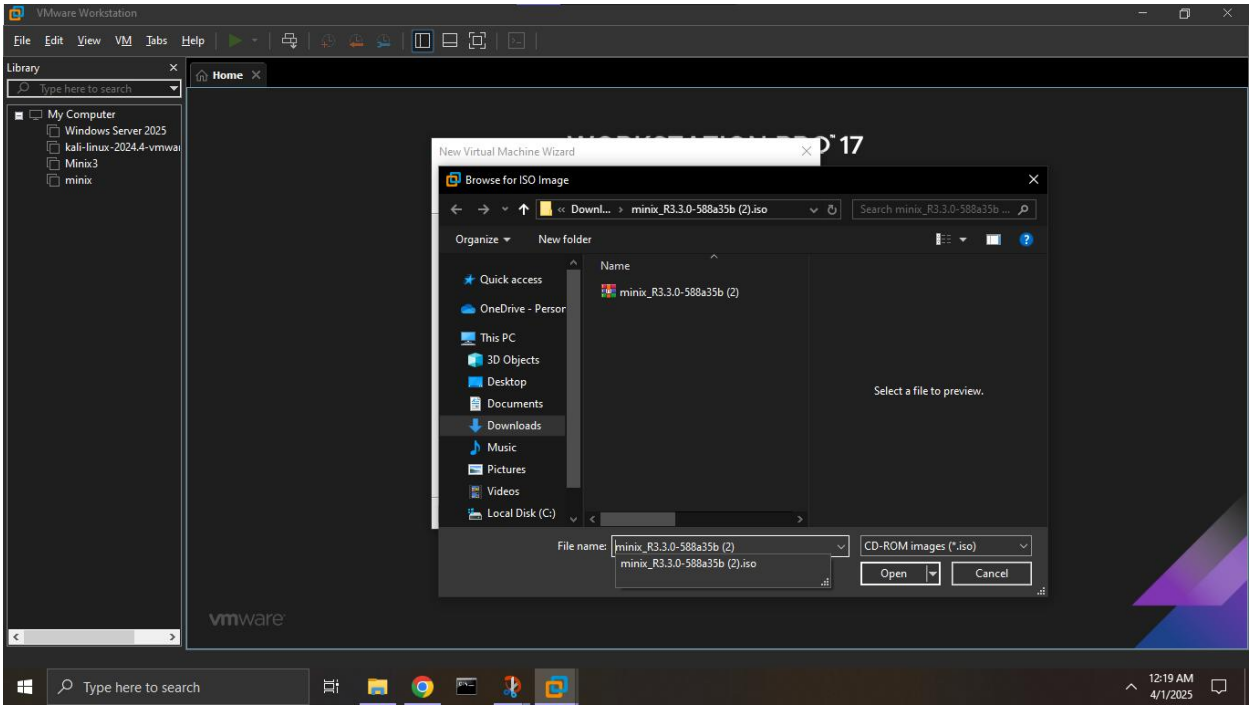
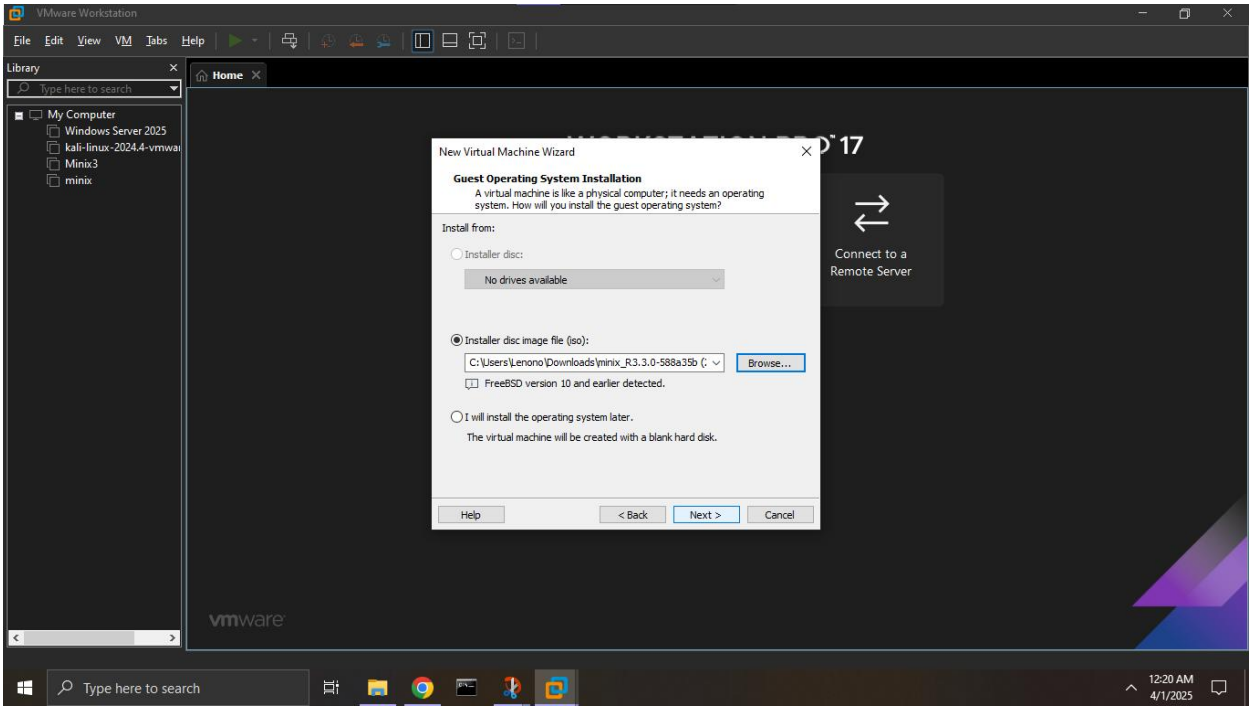


Image-set 3. Loading the Minix ISO file into the new virtual machine



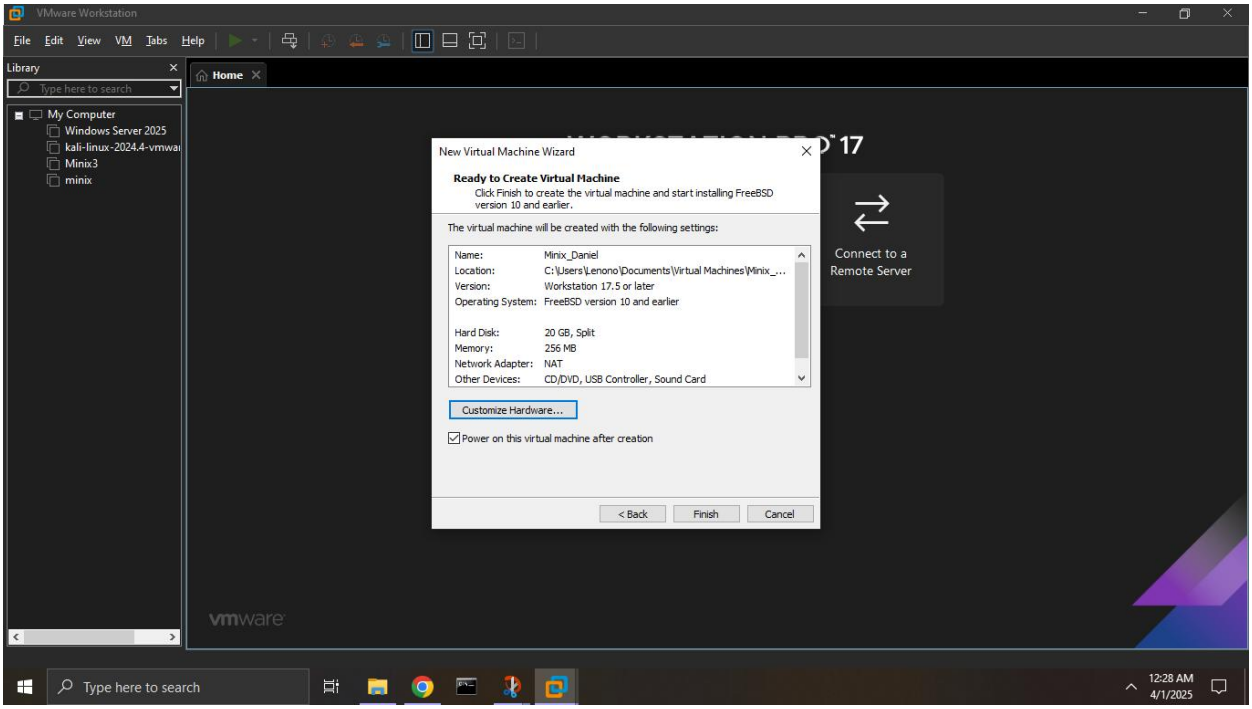
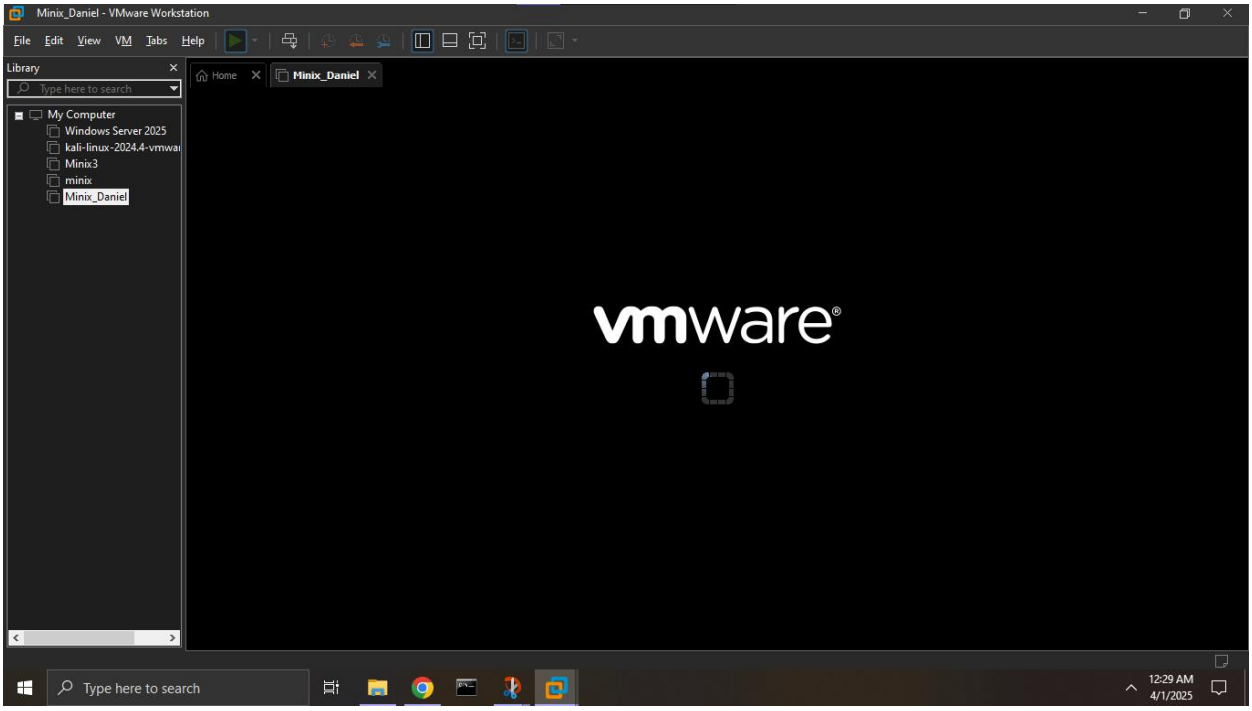


Image-set 4. Assigning hardware



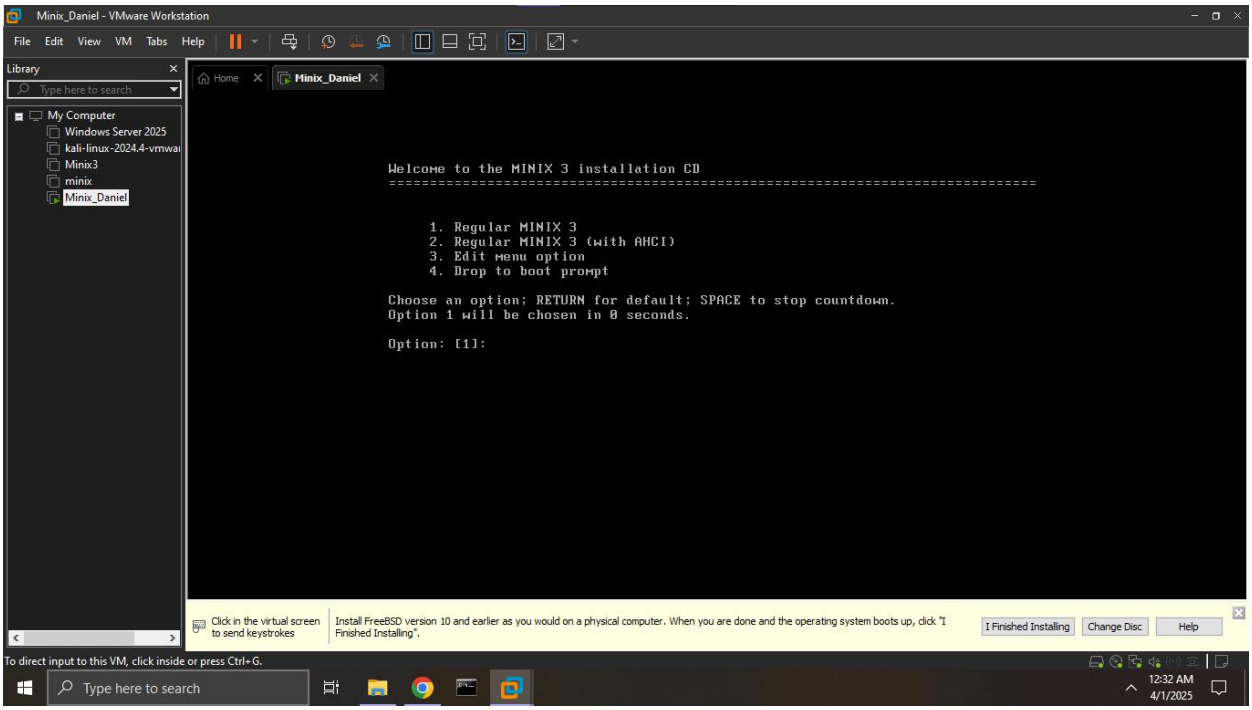
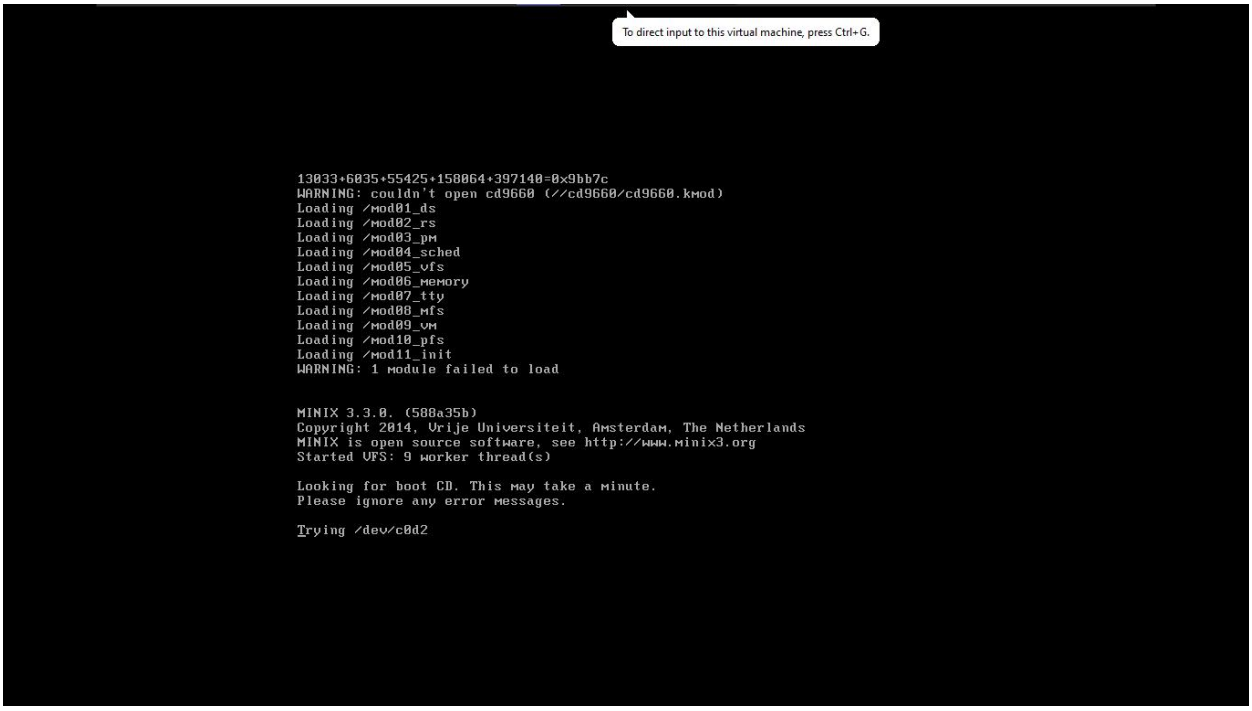


Image-set 5. Turning on the Guest Operating System



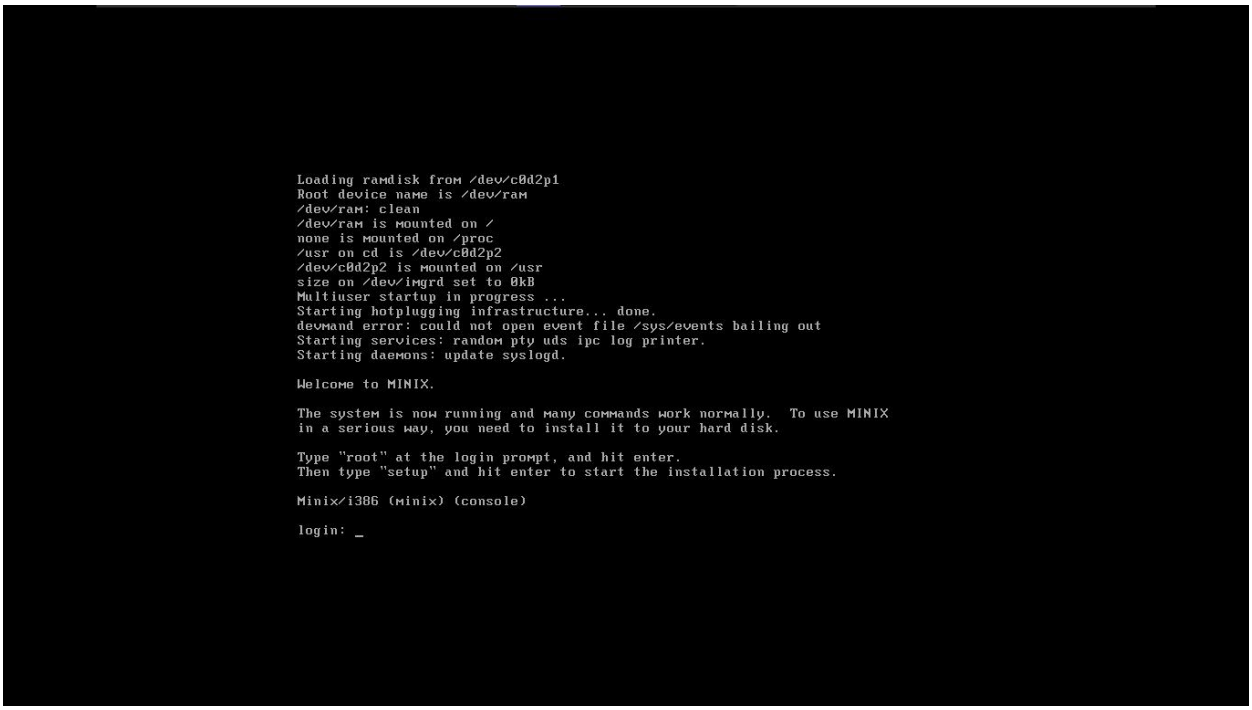
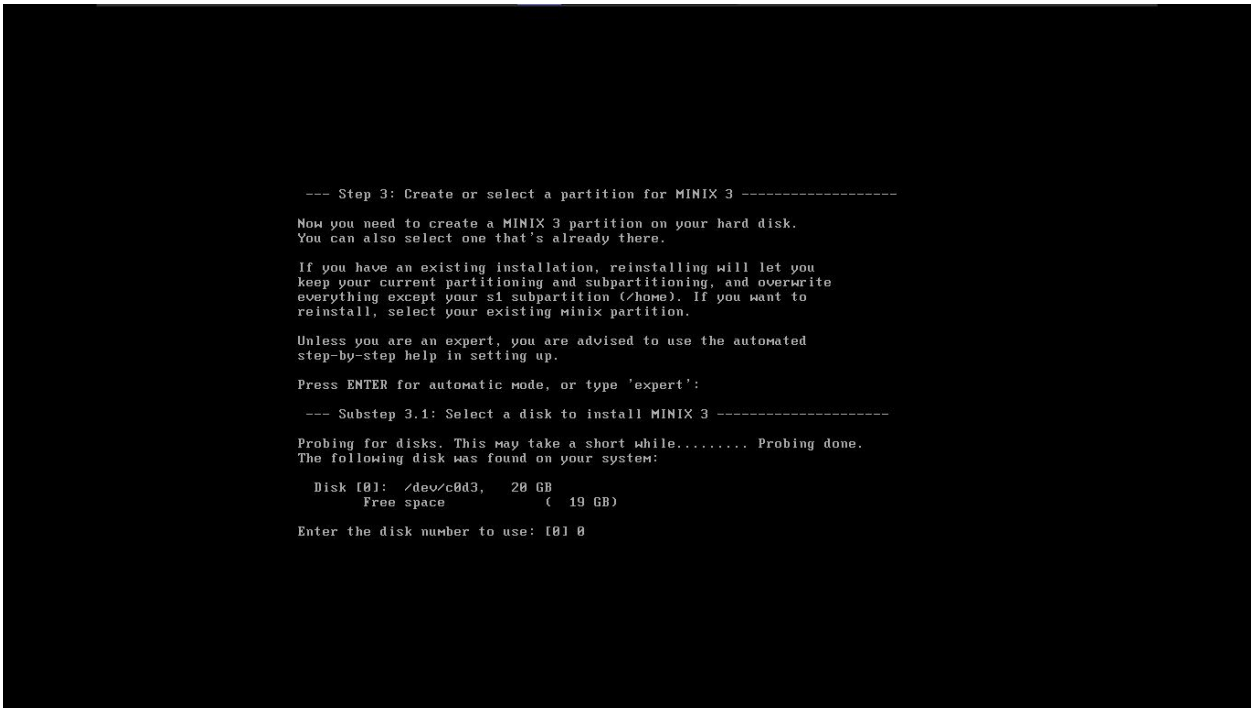


Image-set 6. Root login

Image-set 7. Initial setting up of the virtual machine

Image-set 8. Choosing keyboard language and setup type



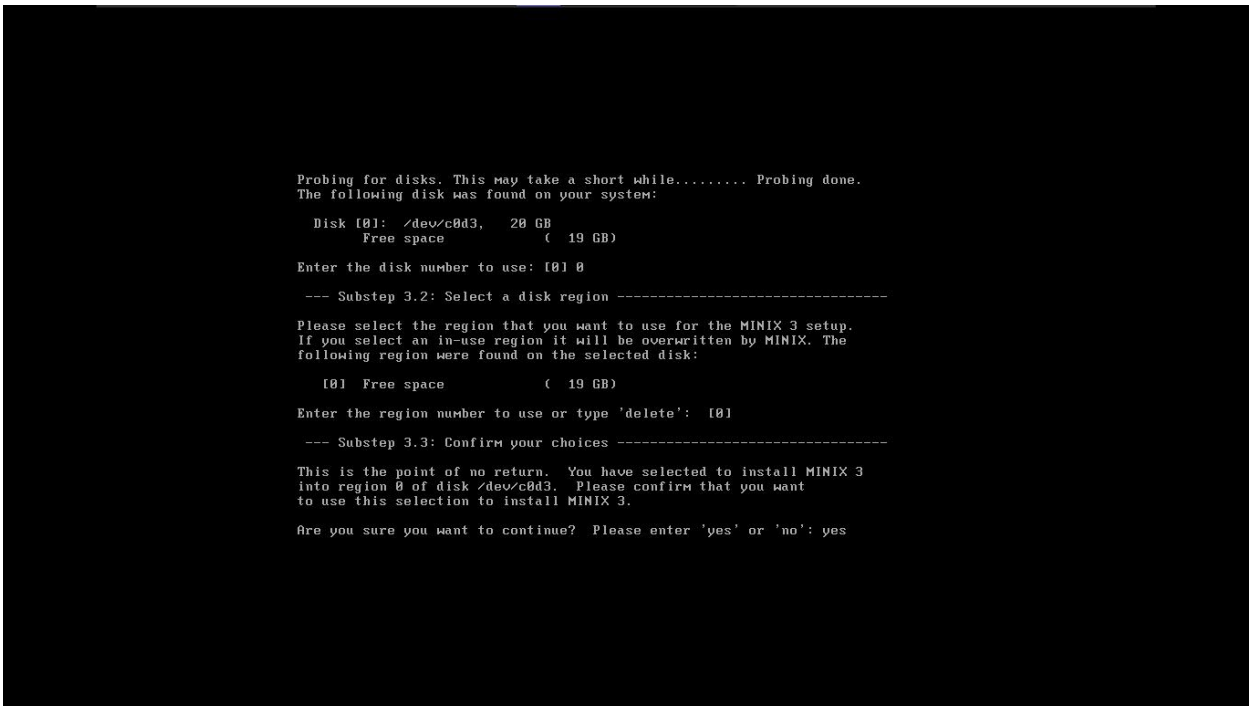
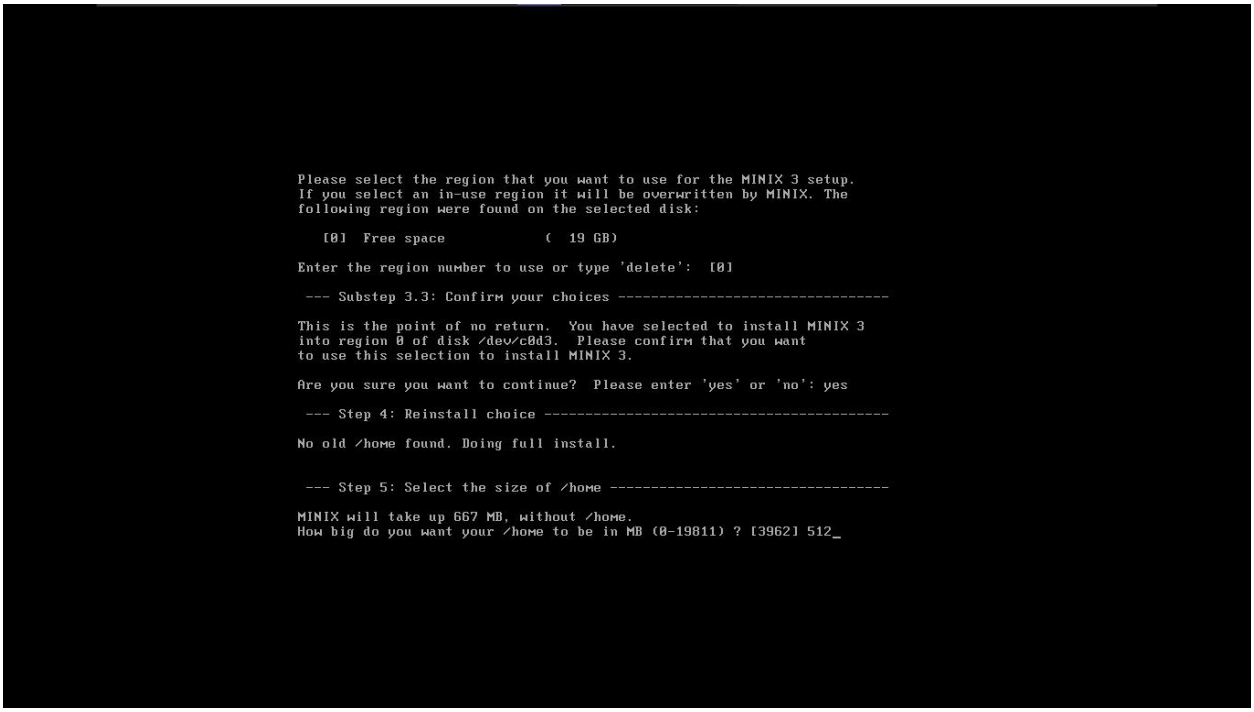


Image-set 9. Partition and disk region setup



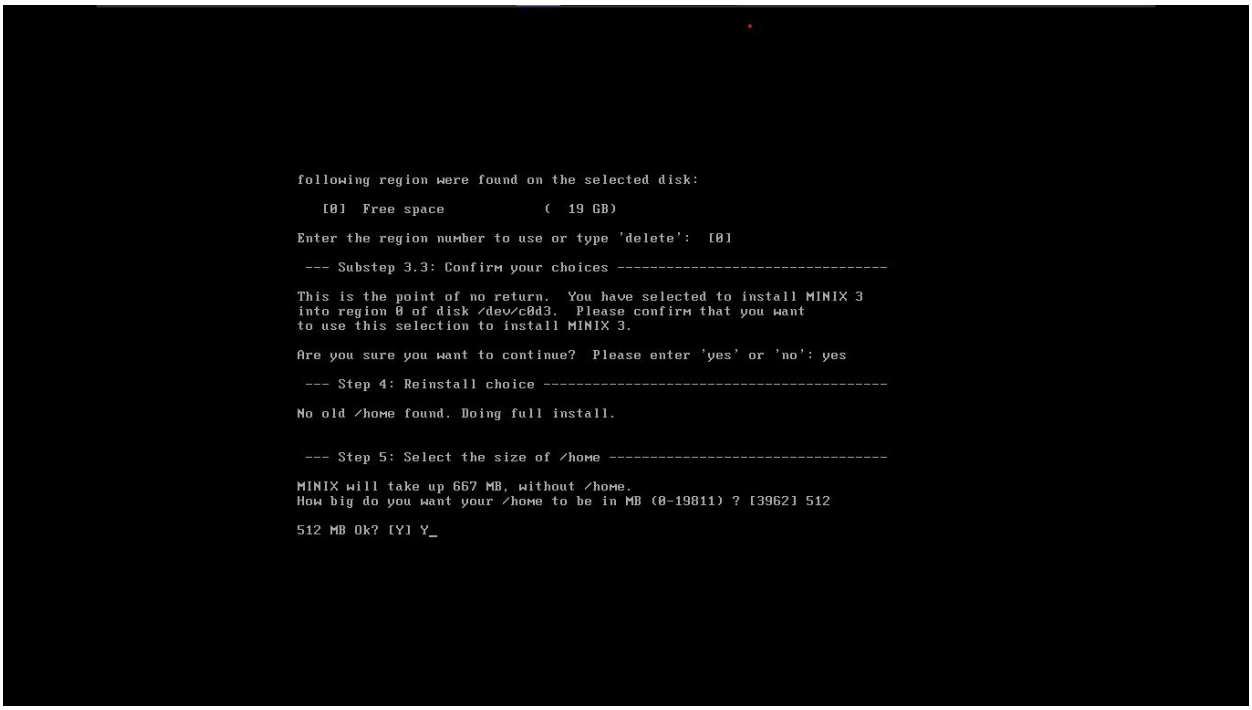
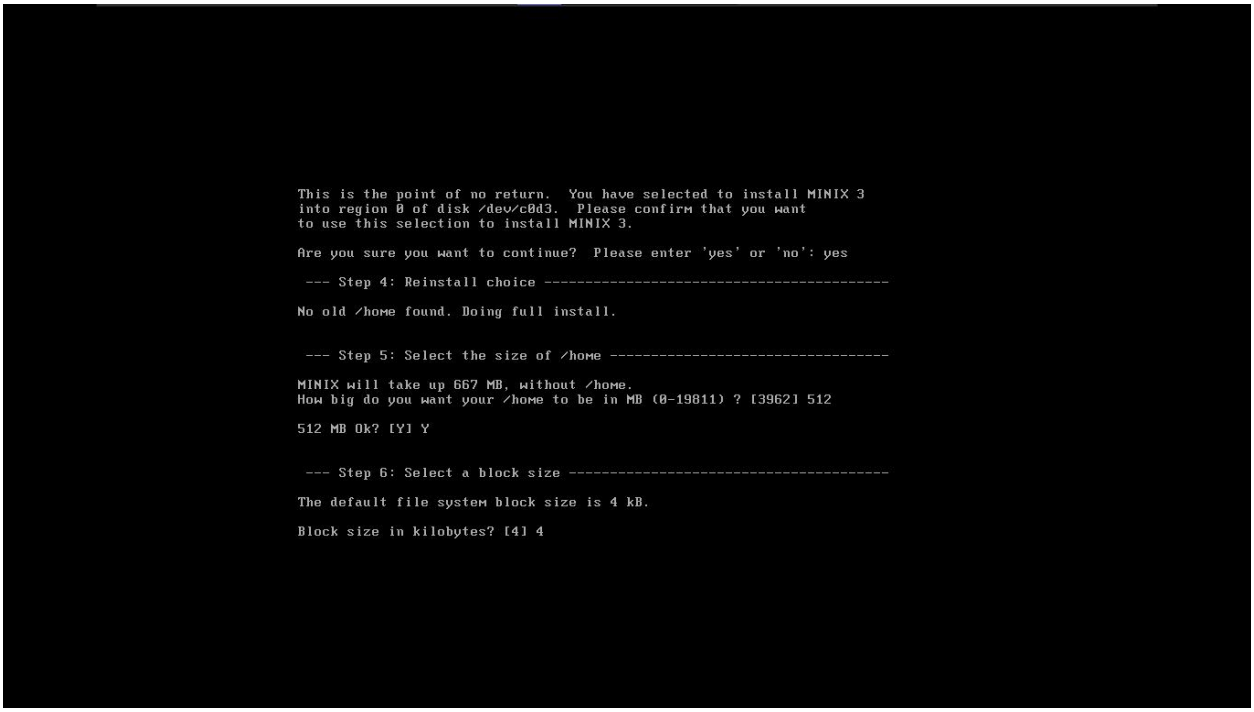


Image-set 10. Home setup



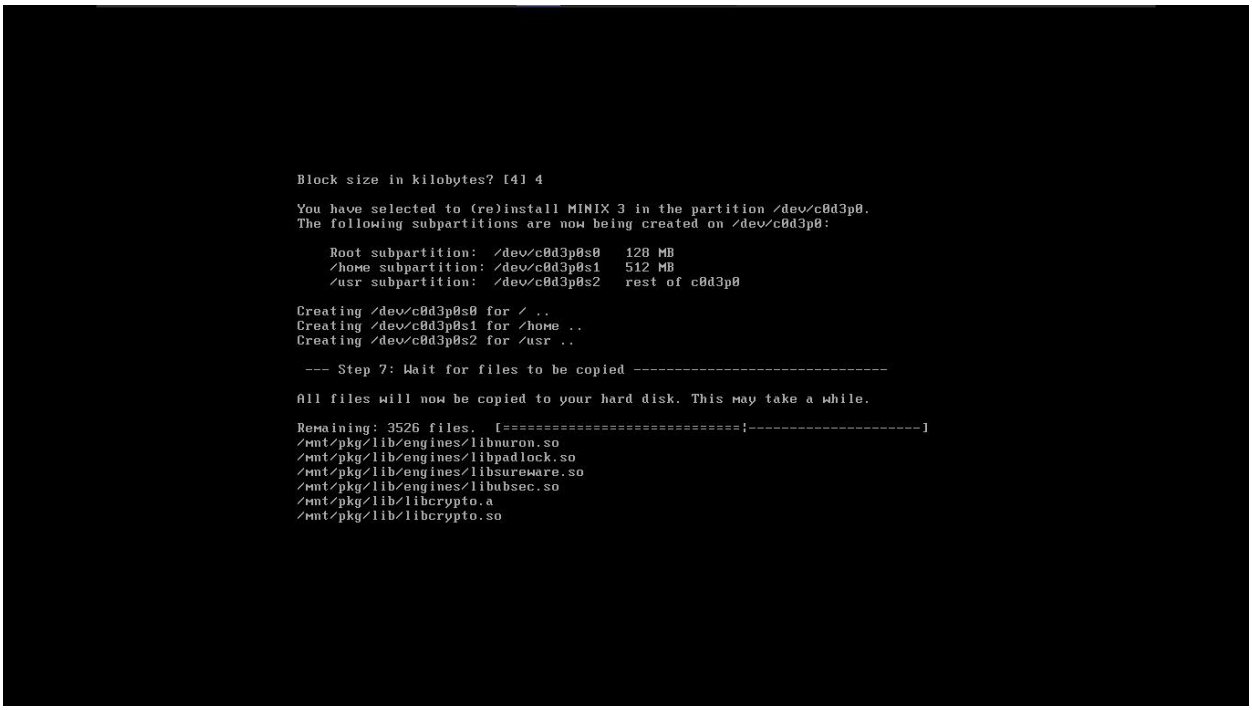


Image-set 11. Selecting block size and copying files to hard disk

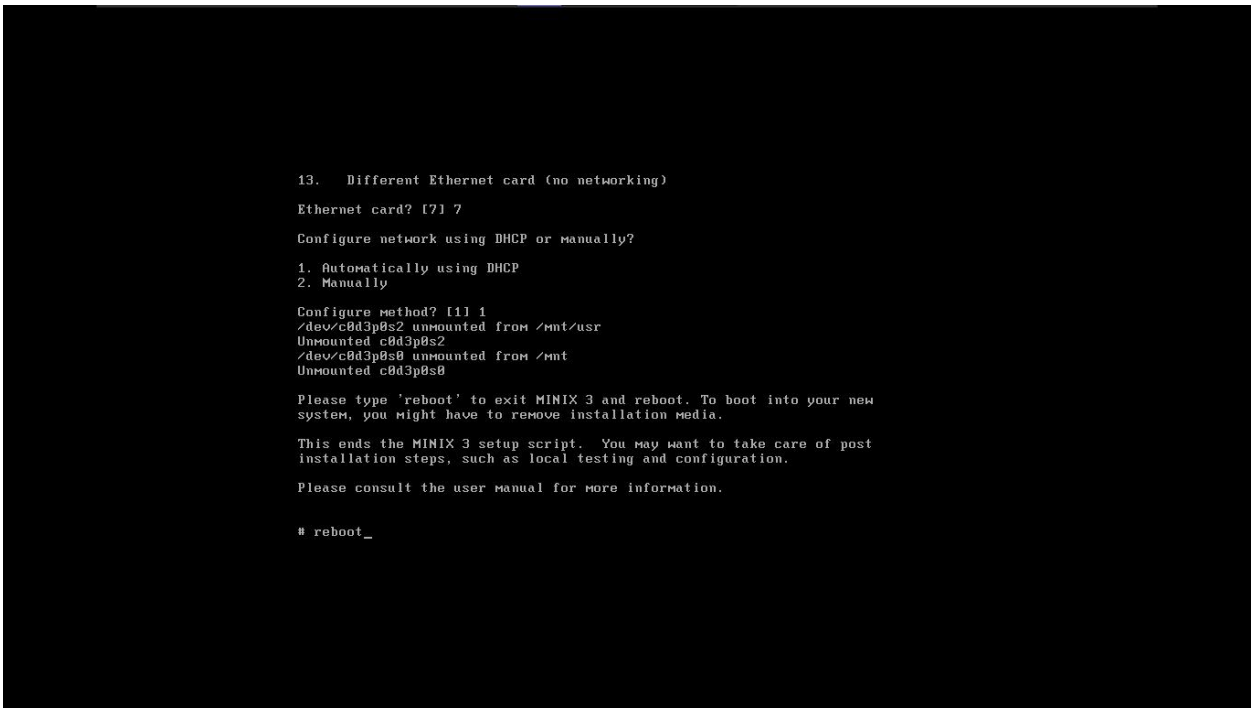
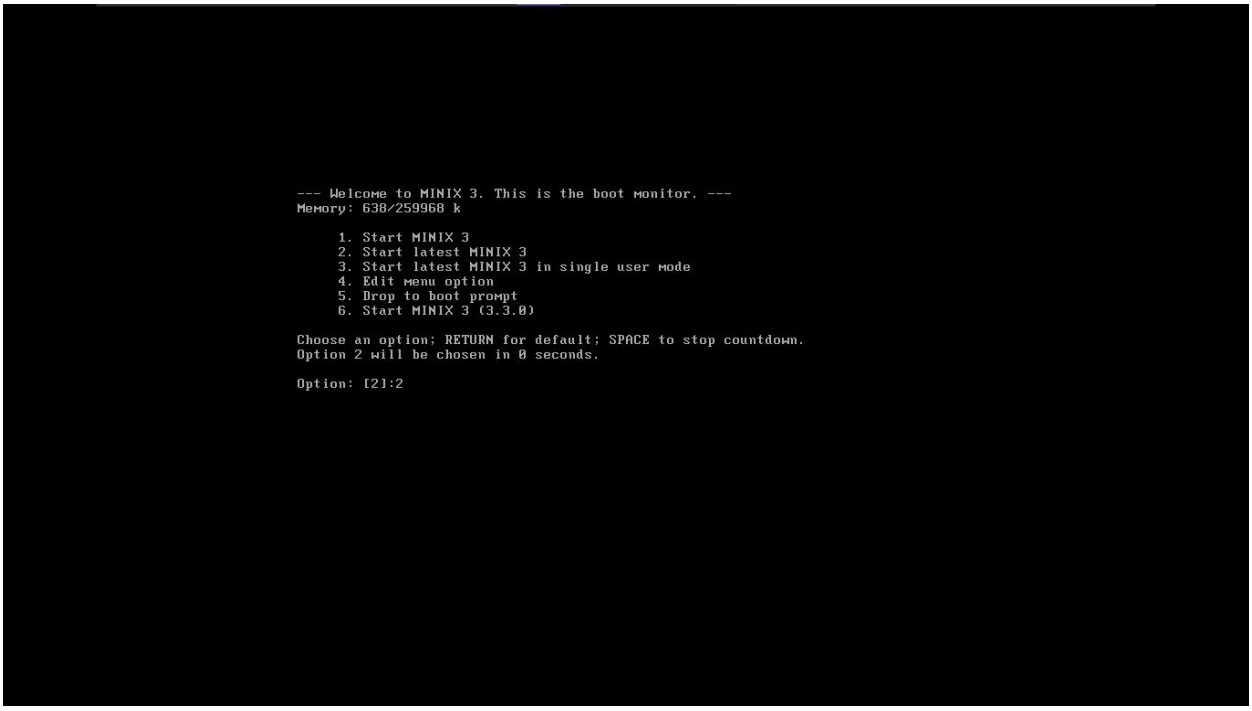


Image-set 12. Network configuration and reboot



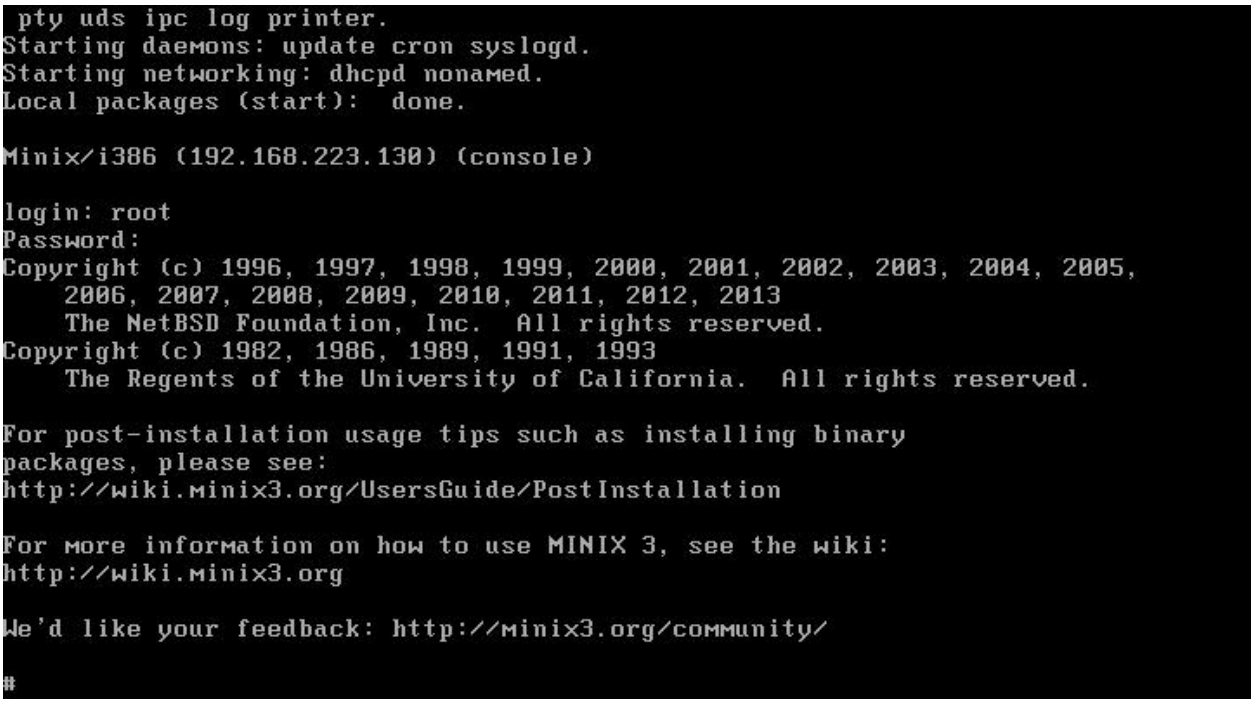


Image-set 13. Minix was successfully installed and correctly booted

Image-set 14. Creating a group called software and a personal user account

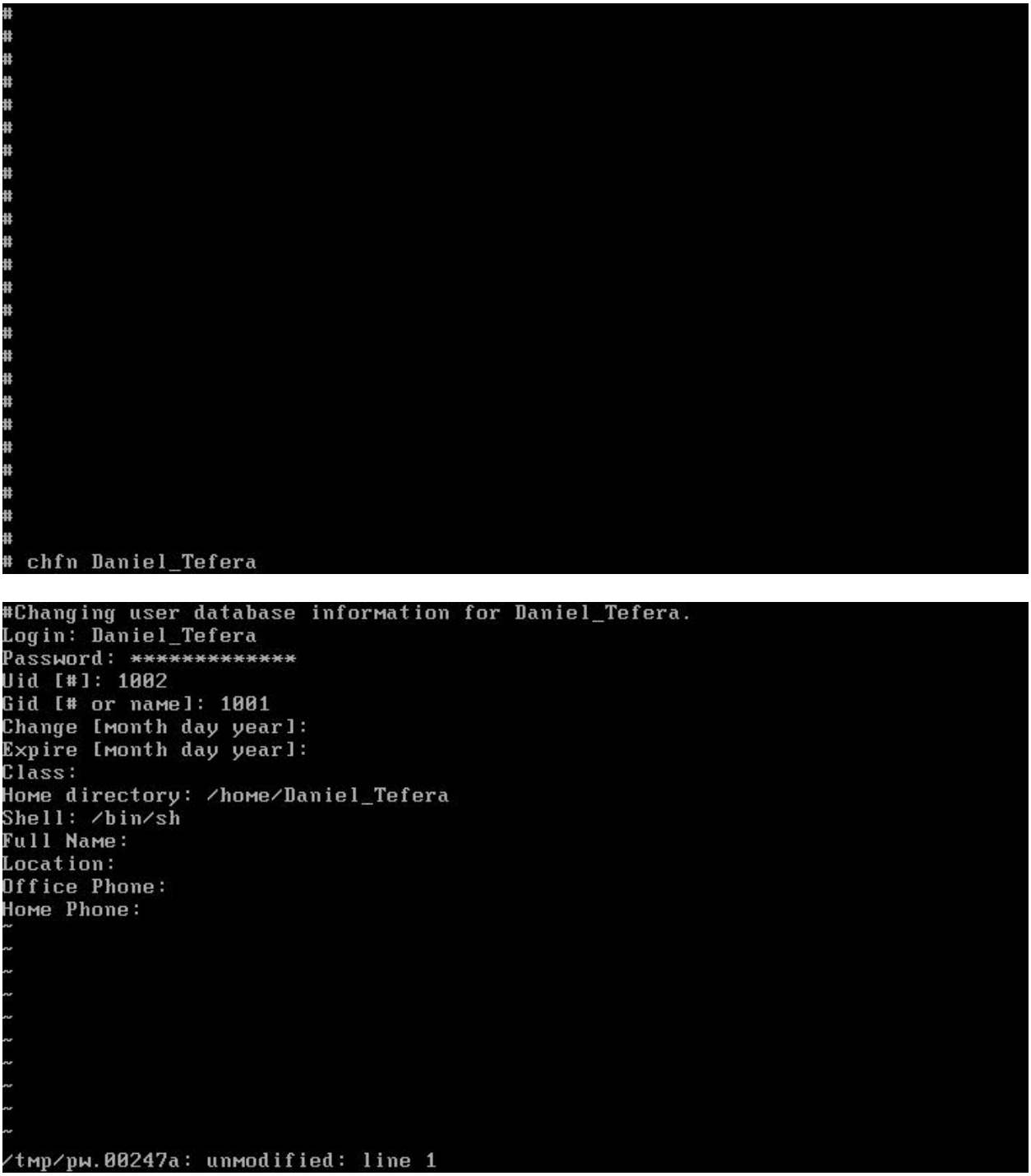


Image-set 15. using chfn command to view and edit the user information

1.5. Issues

- The first issue was that VM Ware needs the user to turn on VT-x options when they are available but it was turned off in my computer.

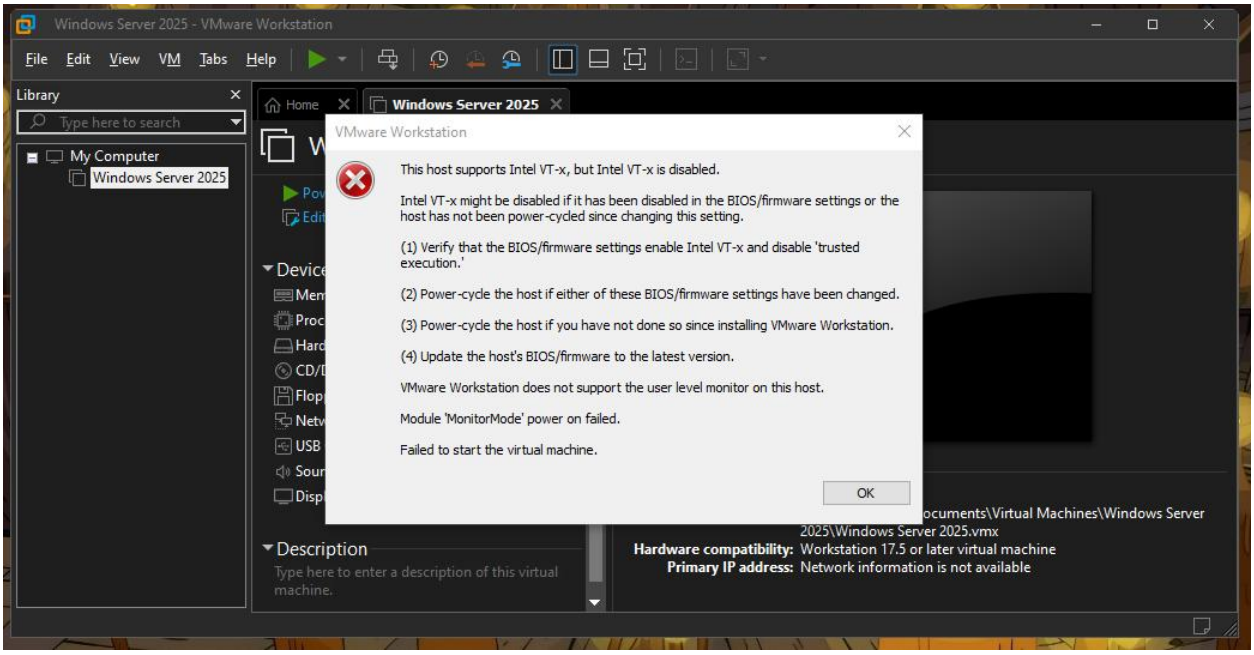


Image 1. VT-x error

- The second problem was that the GUI interface for Minix was unavailable for both updating and installation so I couldn't use the command **Startx** to start the GUI mode.
- The other problem was with installing the necessary packages and the unfamiliar command line interface environment and commands.
- There also was a limit to the characters used for the username, so I couldn't use my full name.

1.6. Solutions

- For the first problem I had to go to the BIOS mode in my computer to enable VT-x.
- For the second problem I researched and it happens that the problem is a bug in the latest available and stable version of Minix, Minix 3.0. There is Minix 4.0 but it isn't available on their official website. So we should wait until it gets fixed but that may not happen as the operating system could be in a "coma" like state. Its last update was on may 2017 G.C. 8 years ago!
- For the third problem I had to update all packages (**pkgin update**) then specifically install what I needed like the editing environment nano(**pkgin install nano**) and the compiler clang(**pkgin intall clang**). The gcc compiler isn't available in Minix.
- I had to use Daniel_Tefera instead of Daniel_Tefera_Alamirew.

1.7. Filesystem support

Minix OS is primarily designed to use the Minix filesystem, but it can also support other filesystems.

1.7.1. Supported Filesystems

- ✓ **MINIX Filesystem**: This is the default filesystem for Minix OS, designed for simplicity and efficiency. It is ideal for educational purposes and is lightweight, making it suitable for the OS's microkernel architecture. It has limited maximum file size and partition size compared to more modern filesystems.
- ✓ **Ext2**: This is an older Linux filesystem that can be used with Minix OS for broader compatibility. It is robust and has no journaling, which aligns with the simplicity of Minix. It is fast and straightforward, suitable for lightweight systems.
- ✓ **FAT32**: Its uinversally supported.
- ✓ **Ext3/Ext4**, **NTFS** and **Btrfs/ZFS** are not supported.

1.8. Advantages and disadvantages

1.8.1. Advatages

- ✓ Educational Focus: Minix OS is specifically designed for teaching and research purposes in operating systems. Its simplicity makes it easier to understand the internal workings of OS design.
- ✓ Microkernel Architecture: The OS uses a microkernel structure, which enhances modularity, reliability, and maintainability.
- ✓ Lightweight: Minix OS has minimal hardware requirements, making it ideal for resource constrained environments.
- ✓ Filesystem Simplicity: The Minix filesystem is straightforward and great for beginners learning about file management systems.
- ✓ Portability: It can run on diverse hardware platforms and in virtual environments, making it versatile for experimentation.

1.8.2. Disadvantages

- ✓ Limited Features: Minix OS lacks the advanced features found in modern operating systems, such as sophisticated GUI interfaces and multitasking capabilities.
- ✓ Filesystem Restrictions: The Minix filesystem has size limitations, making it less suitable for handling large files or partitions.
- ✓ Driver Support: Limited support for modern hardware drivers, which can lead to compatibility issues.
- ✓ Not Suitable for Production: While excellent for educational purposes, Minix OS is not designed for commercial or production environments.

- ✓ Older Technology: Being designed with educational goals in mind, it does not stay updated with modern computing needs and trends. And bugs like the GUI installation problem we saw earlier may not get fixed.

1.9. Conclusion

Minix OS provides an excellent foundation for learning the core principles of operating systems due to its microkernel architecture, lightweight design, and educational focus. Setting up Minix OS in a virtual environment allows users to safely experiment with the OS's capabilities, understand its filesystem support, and troubleshoot issues without risking physical hardware. Despite its limitations in modern features and driver support, Minix OS remains a powerful tool for understanding modularity and reliability in system design!

1.10. Future Outlook/Recommendations

- ✓ Enhance Educational Usability: Minix OS could further develop tools and documentation to support students and researchers, making complex OS concepts more accessible.More tutorials and case studies on microkernel usage could be provided.
- ✓ Compatibility Improvements: Future updates could focus on broader hardware and software compatibility, ensuring Minix OS can function seamlessly across diverse virtual environments.
- ✓ Filesystem Development: Expanding filesystem capabilities to support larger files and partitions would make Minix OS more practical for modern use cases.
- ✓ User Interface Enhancements: Adding more interactive GUIs could make Minix OS more user-friendly while maintaining its lightweight nature.
- ✓ Integration with Modern Platforms: Minix OS could explore integration with IoT devices, embedded systems, and other cutting-edge technologies where its modularity would shine.

2. Virtualization

- Virtualization is the process of creating a virtual version of computing resources, including hardware platforms, storage devices, and network resources. This allows multiple virtual instances of operating systems to run on a single physical machine. In modern operating systems, virtualization is a very important technology that enhances resource utilization, security, and flexibility. By integrating hypervisors, supporting containerization, and providing robust management tools, these systems enable efficient operation of multiple isolated environments, making them essential for both enterprise and cloud computing.

- We Use Virtualization for:

  - ✓ Resource Optimization: Virtualization enables better utilization of hardware resources by running multiple OS instances on one physical computer. This reduces the price that we would have to pay for deploying multiple computers for tasks like application software testing.

  - ✓ Isolation: Each virtual machine (VM) operates independently. This increases security and minimizes conflicts between applications.

  - ✓ Development and Testing: Developers can create isolated environments to test applications without affecting the host Operating System.

  - ✓ Scalability: Virtualization allows for quick scaling of resources in response to varying workloads.

- How Virtualization Works:

Virtualization works by using a hypervisor to create and manage virtual machines (VMs) on a physical server. The hypervisor allocates resources like CPU, memory, and storage to each VM, allowing them to run independently as if they were separate physical computers. Users can install different operating systems on these VMs and run applications without impacting each

other. This setup enhances resource efficiency, flexibility, and security, as each VM is isolated from the others.

- ❖ Main actors and concepts in virtualization

- ✧ **Hypervisor**: A hypervisor is software that *creates and manages* virtual machines. There are two types: Type 1 (bare-metal) which runs directly on hardware, while Type 2 (hosted) runs on top of an existing operating system like VM Ware. It allocates the resources that the Guest Operating System needs.

- ✧ **Guest Operating System** : Each Virtual Machine runs *its own operating system*, known as the Guest Operating System, which can be different from the host OS.

- ✧ **Containerization**: Is the use of container technologies that provide lightweight, isolated environments for applications, enhancing deployment and scalability while sharing the *same OS kernel*..

References

https://www.minix3.org/

https://en.wikipedia.org/wiki/Minix

https://en.wikipedia.org/wiki/Virtualization