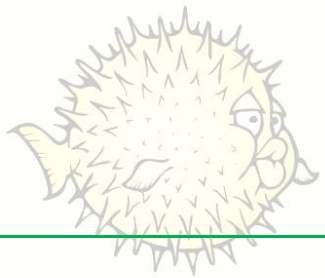
BAHIR DAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY

FACULTY OF COMPUTING

Department: Software Engineering

Year: 2nd Year, 2017 E.C



NAME **ID**

FINOTELOZA WANAW……1601526

OpenBSD

SUBMITTED TO: Mr. WONDIMU BAYE

Submission Date: 16/07/2017 E.C.

OPERATING SYSTEM ASSIGNED: OPENBSD ISO
VIRTUALIZATION TOOL: VMWARE WORKSTATION

PROJECT-WORK

OPERATING SYSTEM AND SYSTEM PROGRAMMING

1. Installation of operating system in virtual environment tools (VMware workstation, oracle vm virtual box, and etc...). Installation may not necessary if the technology tool is outdated, has no long term support (LTS), and no hardware support only. Which OS you are going to install refer on the attached document. Your name list as well as OS list is incorporated. Contents to cover when you prepare documentation:

a. Introduction (background, motivation)

b. Objectives

c. Requirements i. Hardware ii. Software

d. Installation steps: i. Include snipped images ii. When you create an account, you should name by your full-name

e. Issues (problem faced): when you face problems, list or snip the problem/s.

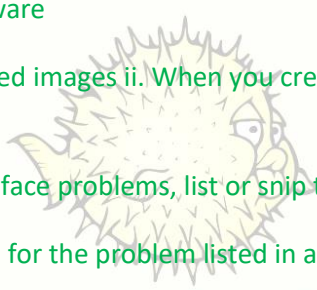
f. Solution: i. if you have the solution for the problem listed in above (e), list the solution

g. File system support.

h. Which file system support(NTFS, FAT32, exFAT, ext4, Btrfs, ZFS, HFS+, APFS) and why

i. Advantage and disadvantage.

j. Conclusion. j. Future outlook /recommendation.



OpenBSD

2. Briefly explain the what, why, and how virtualization in modern operating system.

4. Implement system calls.

a. Introduction to OPENBSD OS

OpenBSD is a free and source, Unix-like operating system based on the Berkeley Software Distribution (BSD). It is widely respected for its focus on **security**, **code correctness**, and **portability**. Developed by a global team of volunteers, OpenBSD is renowned for integrating advanced security features directly into the operating system, making it a preferred choice for firewalls, secure servers, and other critical infrastructure.

Unlike many general-purpose operating systems, OpenBSD is designed with a **"secure by default"** philosophy, minimizing vulnerabilities from the outset. Its robust security practices, such as proactive code auditing and privilege separation, have influenced many other operating systems and security tools.

Historical background of OPENBSD OS

The OpenBSD project was founded in 1995 by Theo de Raadt, a Canadian software developer who had been one of the original founders of the NetBSD project. After parting ways with the NetBSD team due to philosophical and managerial differences, Theo launched OpenBSD with the goal of creating a cleaner, more secure BSD variant.

OpenBSD started as a fork of NetBSD 1.0, but it quickly distinguished itself by placing a strong emphasis on security and code transparency. The OpenBSD team began a systematic source code audit process to identify and fix bugs, security flaws, and poorly written code. This rigorous approach resulted in a reputation for extreme reliability and security.

Over the years, OpenBSD has contributed several important tools to the open-source ecosystem, including:

- OpenSSH – A secure replacement for Telnet and other remote login programs (widely used across Unix-like systems)
- PF (Packet Filter) – A powerful firewall and traffic shaper
- LibreSSL – A more secure fork of the OpenSSL cryptographic library
- OpenSMTPD – A secure and easy-to-use mail transfer agent

Motivation behind OPENBSD OS

The primary motivations behind the creation and continued development of OpenBSD stem from a desire to address the following issues:

1. **Security**: Theo de Raadt and the OpenBSD team were frustrated with the lack of attention paid to security in many existing Unix-like operating systems at the time. They wanted to build an operating system that was inherently **secure by default**. This includes features such as:

- **Code auditing**: Every line of code is reviewed to find and fix security vulnerabilities.
- **Secure defaults**: The system is designed with security in mind from the start, minimizing the risk of misconfiguration.
- **Strong cryptography**: OpenBSD integrates cryptographic algorithms directly into the system.

2. **Code Correctness**: Another driving motivation was the desire to write **clean, correct, and simple code**. The OpenBSD developers believe that the quality of code matters greatly when it comes to security and reliability. As such, OpenBSD's development is focused on minimizing bugs and errors, ensuring the system is as stable as possible.
3. **Transparency and Openness**: Theo de Raadt emphasized **transparency** in the development process. The goal was to create an operating system that was not only open-source but also **auditable** and **trustworthy**. By making all the code publicly available, OpenBSD ensures that no malicious code or backdoors can be hidden from scrutiny.
4. **Long-Term Stability**: OpenBSD is designed to provide a reliable and stable platform for years to come. It follows a predictable release schedule and provides long-term support, making it suitable for use in mission-critical systems.

Objectives of OPENBSD OS

The core objectives of the OpenBSD project include:

1. **Security:**
   OpenBSD strives to be the most secure operating system available. It implements a proactive security model, including features such as:
   - Secure default configurations
   - Memory protection techniques like W^X (Write XOR Execute)
   - Privilege separation and reduction (e.g., pledge() and unveil() system calls)
2. **Correctness:**
   Code correctness is a top priority. The OpenBSD team conducts continuous source code audits to improve code quality and reduce bugs, which in turn leads to greater system stability and fewer security vulnerabilities.
3. **Simplicity and Clarity:**
   The project avoids unnecessary complexity and favors straightforward, maintainable code. This makes it easier to understand, audit, and maintain the system.
4. **Portability:**
   OpenBSD is highly portable and runs on many hardware architectures, both modern and legacy, making it useful for embedded systems, servers, and development environments.
5. **Documentation:**
   OpenBSD is known for its comprehensive and high-quality man pages. Almost every component of the system is well documented, providing an excellent learning resource for users and administrators.

c. Requirements

i. Hardware

- CPU: Intel Core i5 or better
- RAM: 4 GB minimum (8 GB recommended)
- Disk: 20 GB of free space
- Virtualization support enabled in BIOS

ii. Software

- Host Operating System: Windows 10/11 or Linux
- VMware Workstation (version 16 or higher)
- OpenBSD ISO image (version 7.5 or latest)

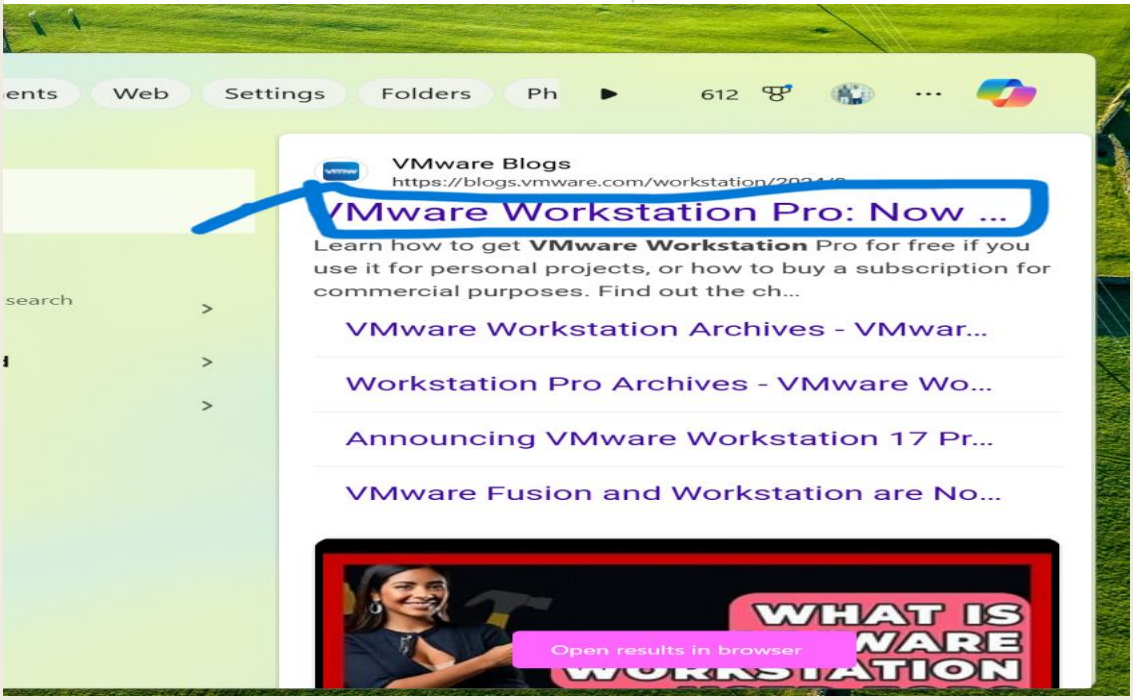
d. Installation Steps

INSTALLATION IT COULD BE DIVIDED INTO TWO:

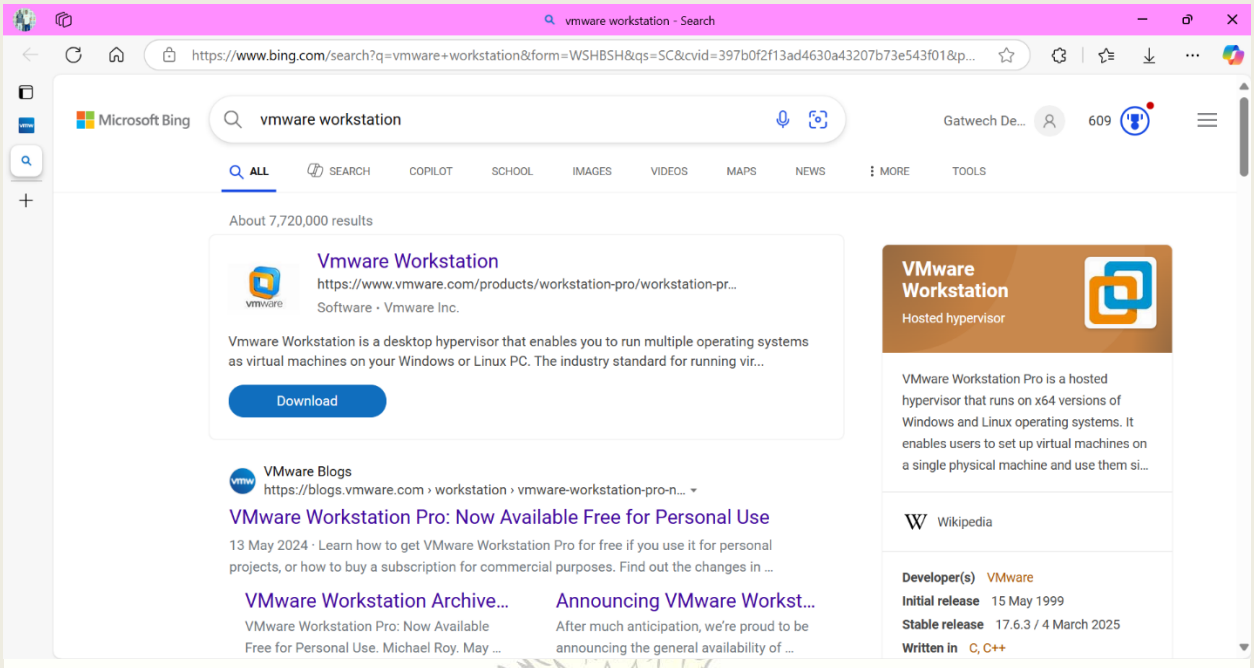
1. Virtual box downloading and installation

2. OPENBSD OS downloading and installation

Downloading Virtual box Go to any browser and search for "Virtual Box". Click the link encircled in blue color.

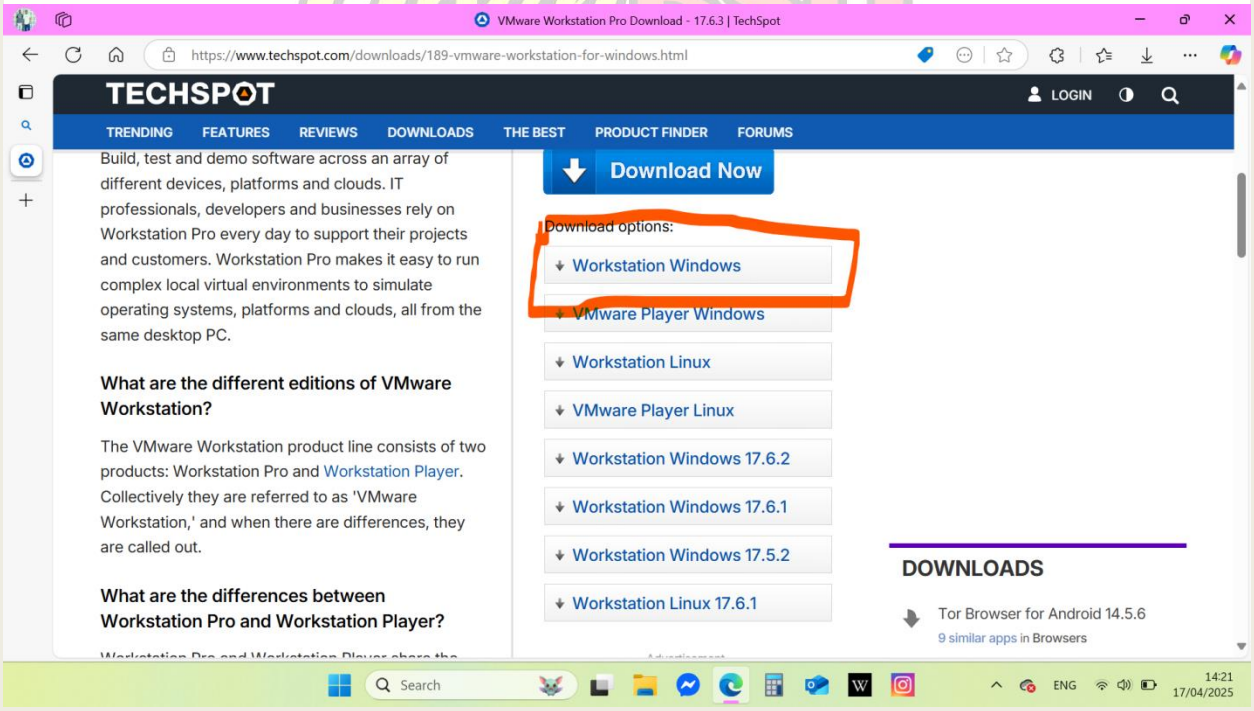


Now you will find all the websites where you can click one website

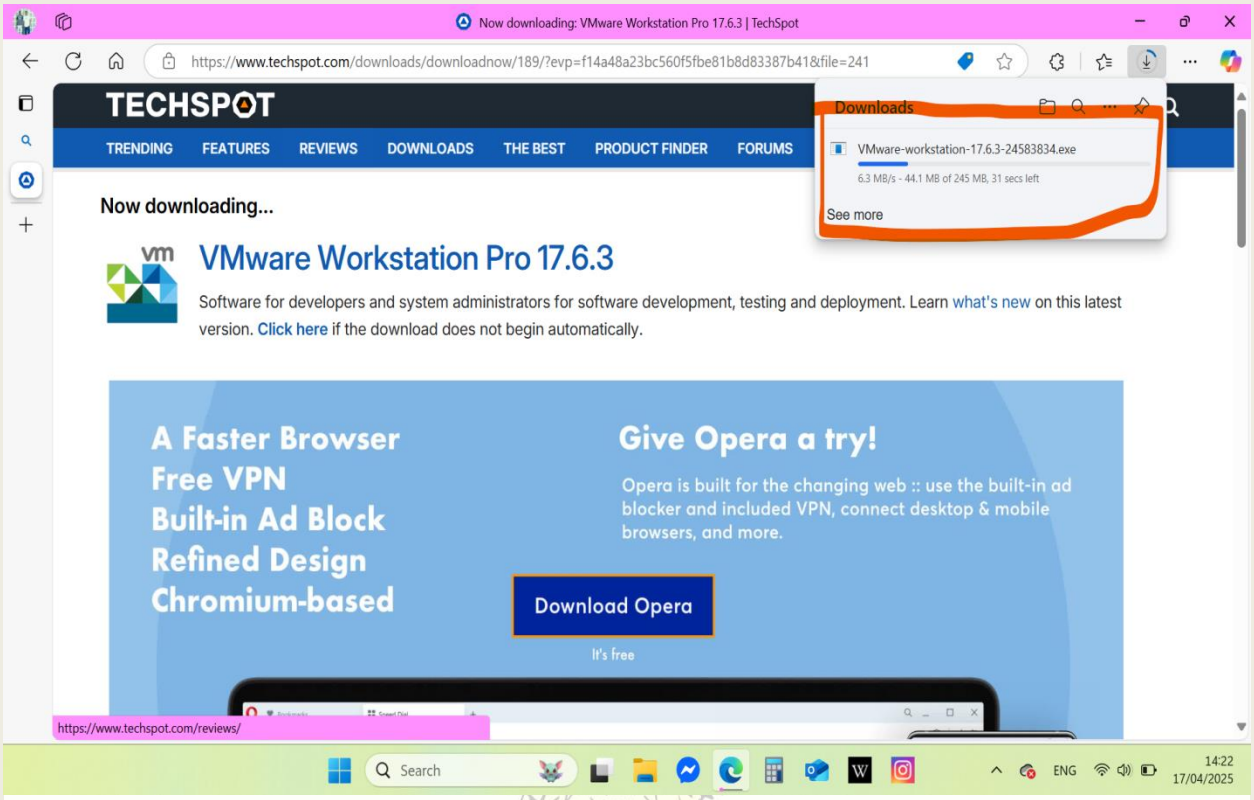


Many os will come where you will have option to decide which os is already in your computer.

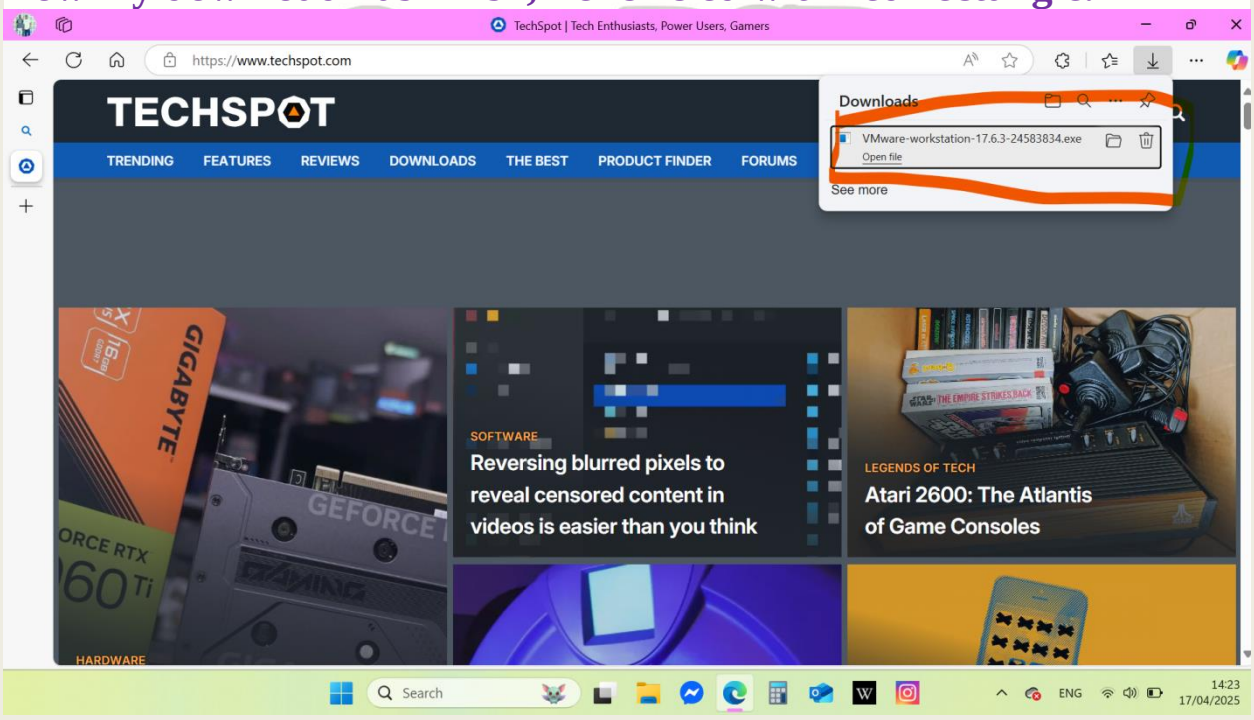
Because am running window in my pc then I have to click on window workstation with red circle.



Now my download will start; it's labeled with red rectangle.

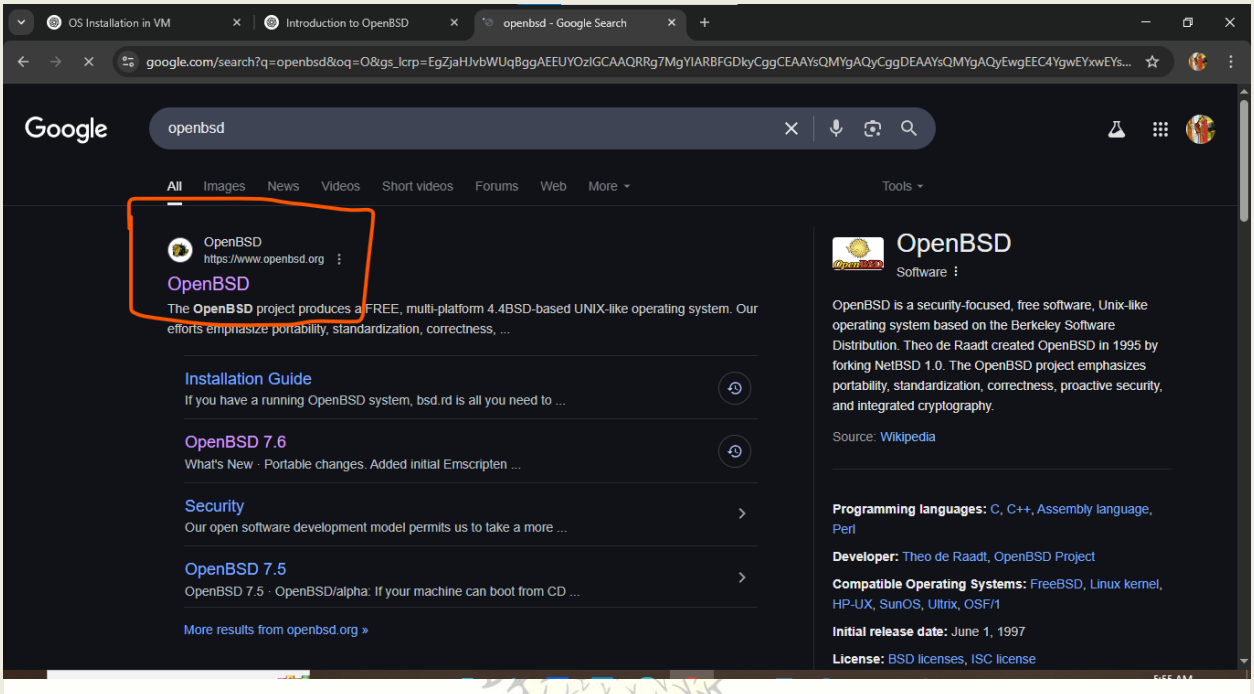


Now my download has finish, I encircled with red rectangle.

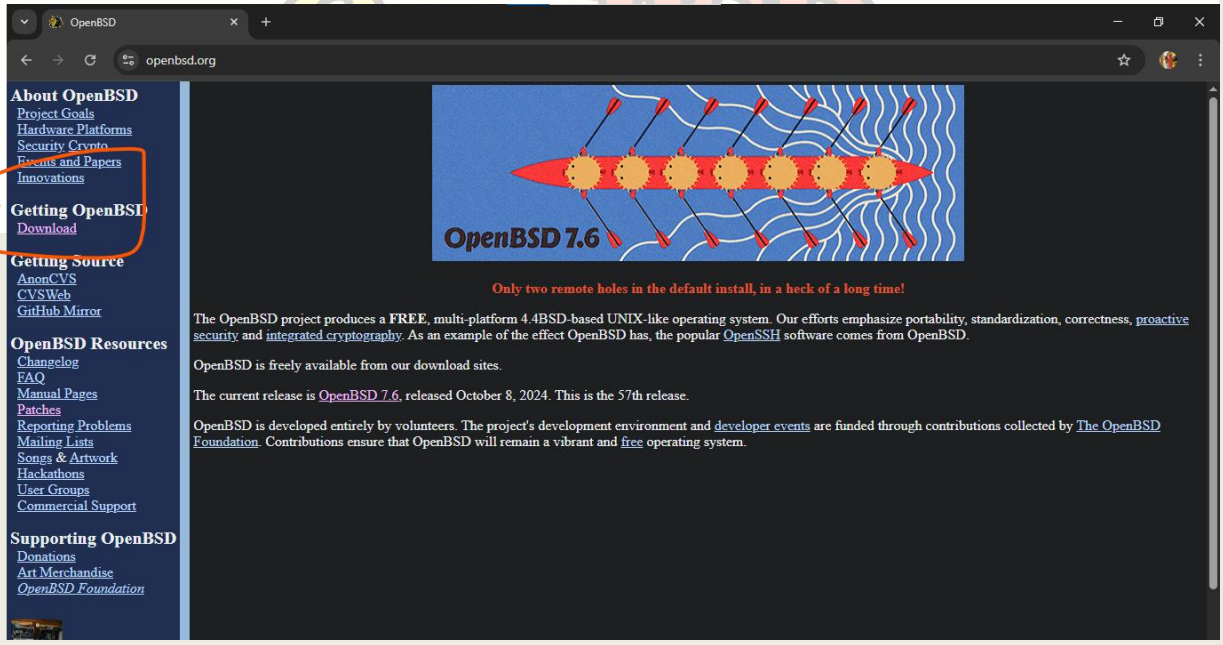


OPENBSD OS downloading steps.

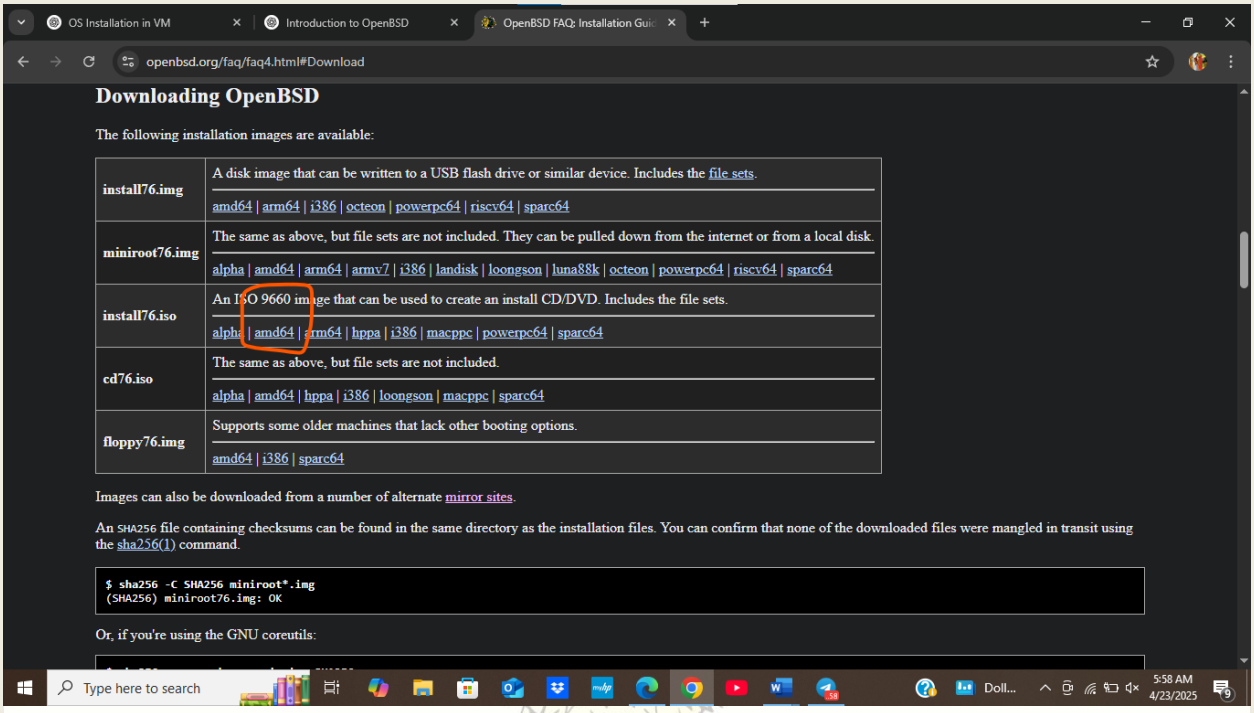
Go to any browser again and search for OPENBSD OS and click the link that encircled with red color



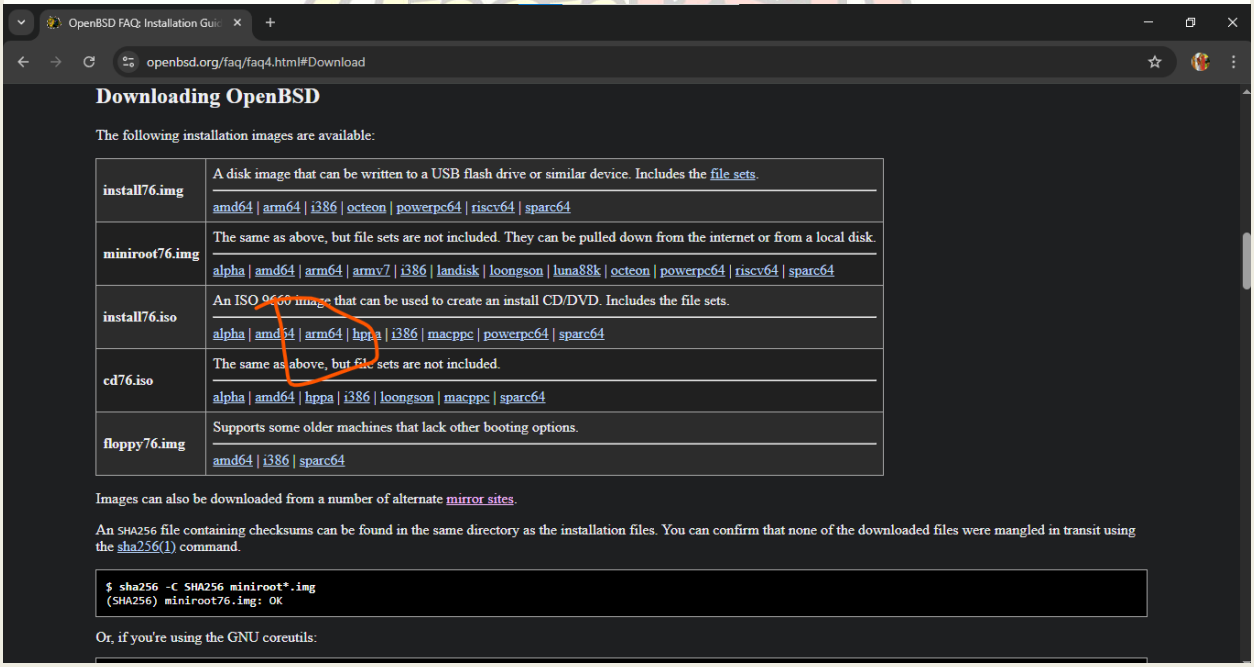
Click downloads



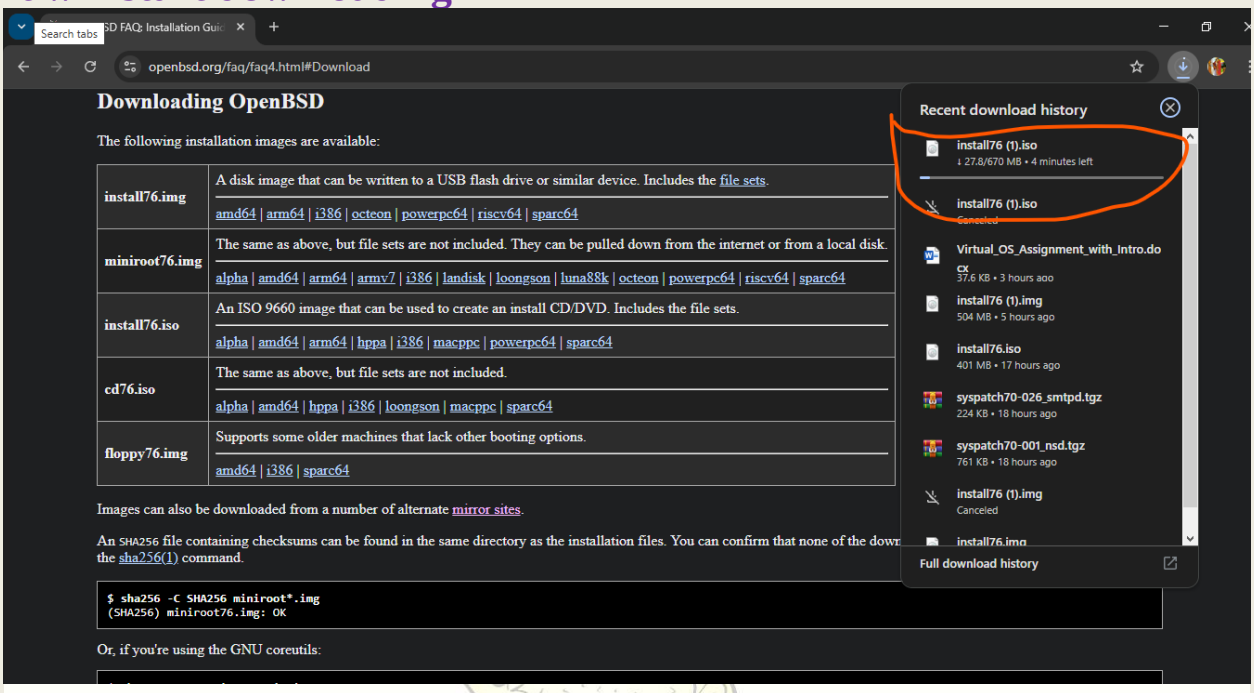
It will ask you for the option (click arms64)



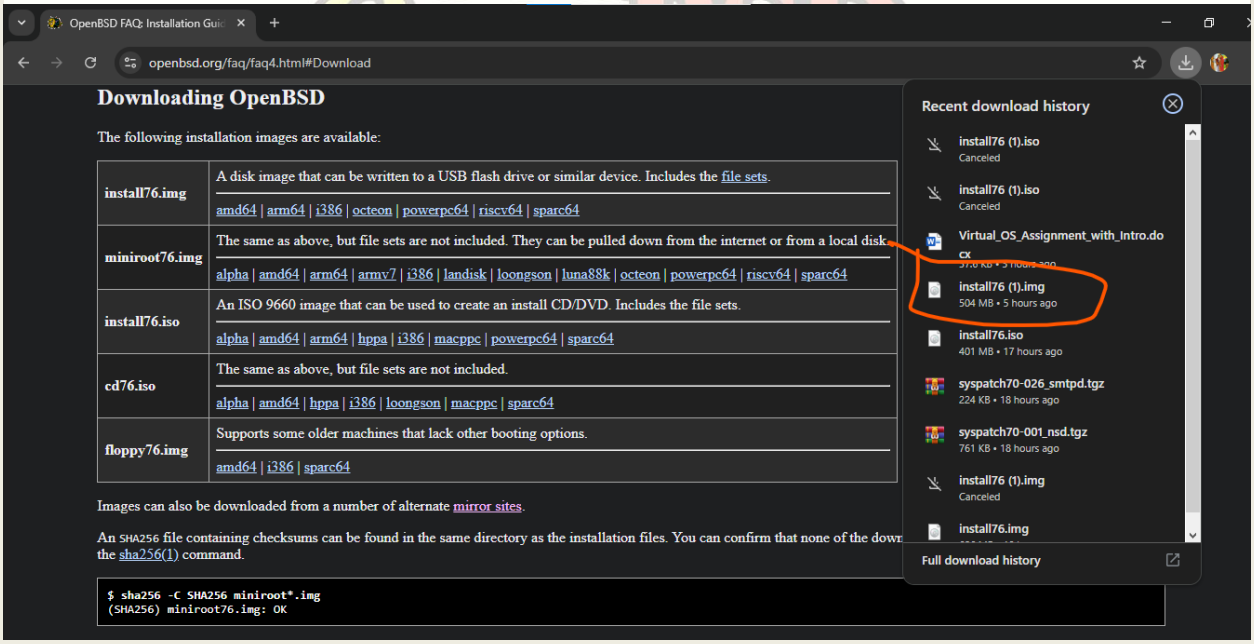
Click 64-bit Download



It will start downloading



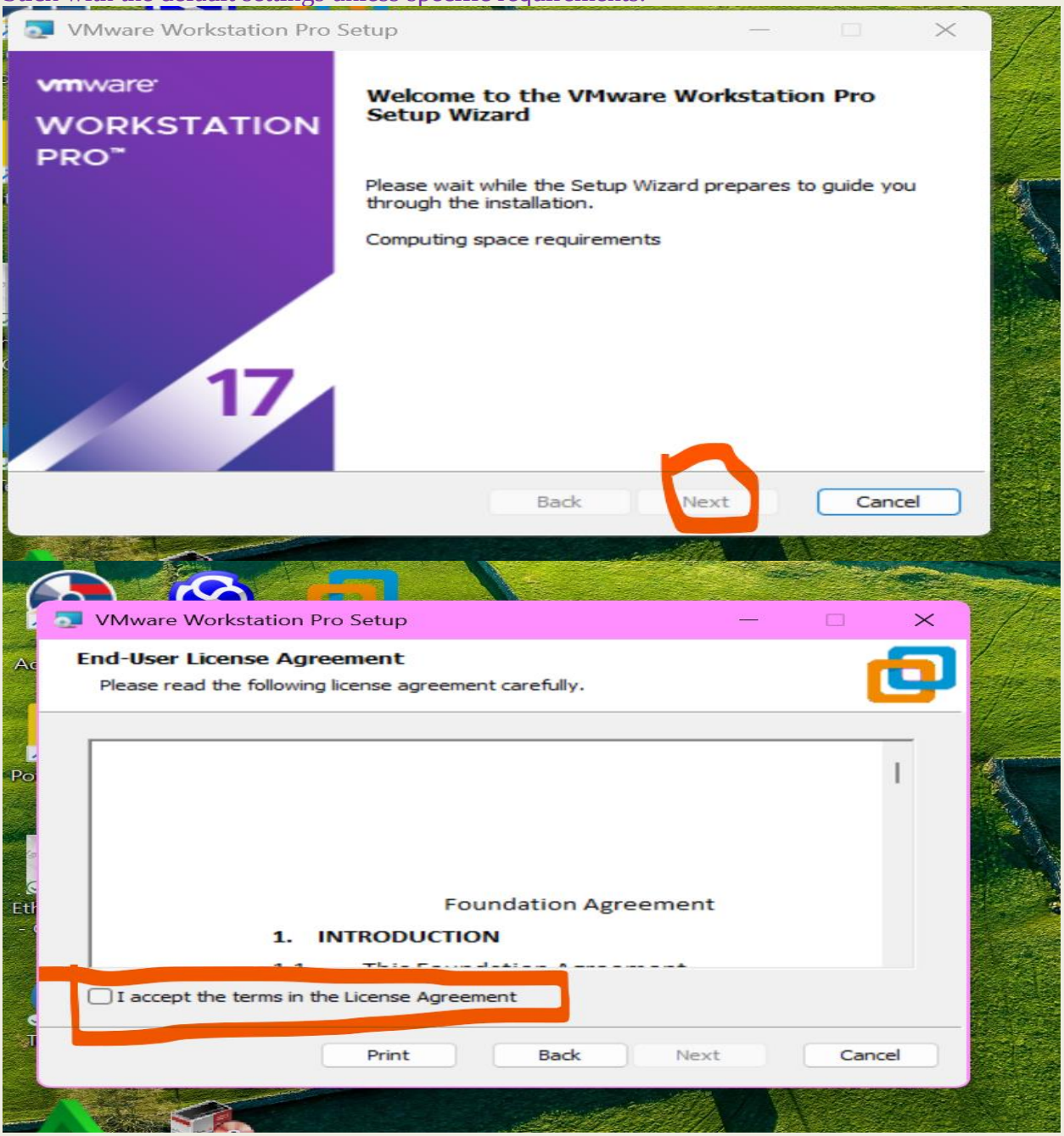
Now the downloading has finished and I can get it in the download history

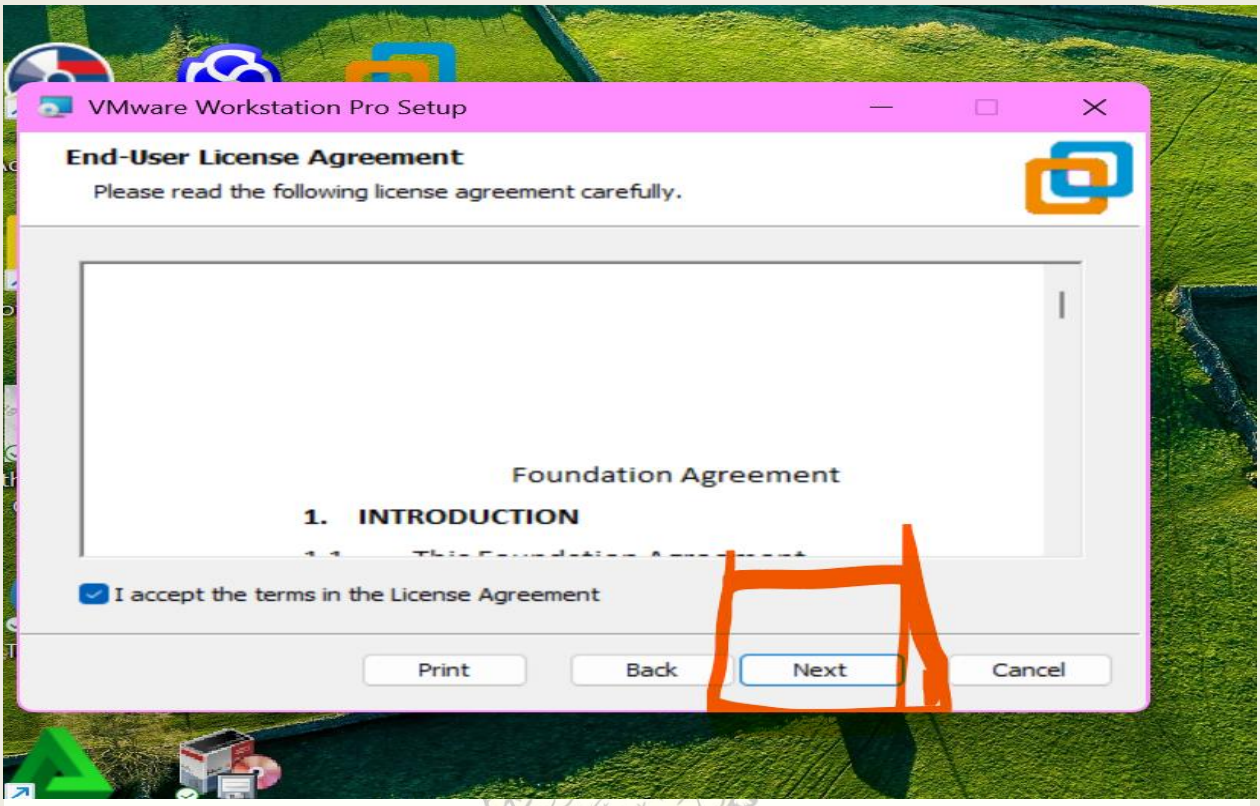


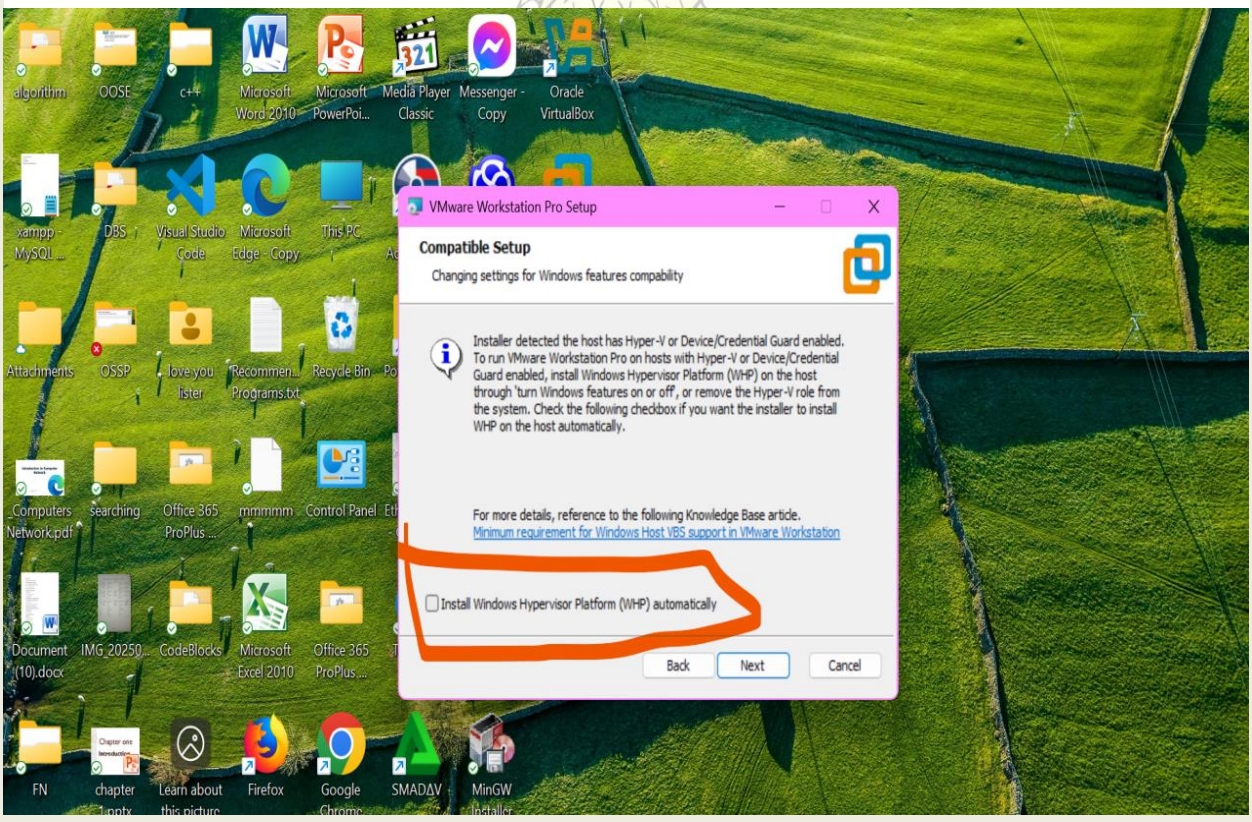
Installation of virtual Box

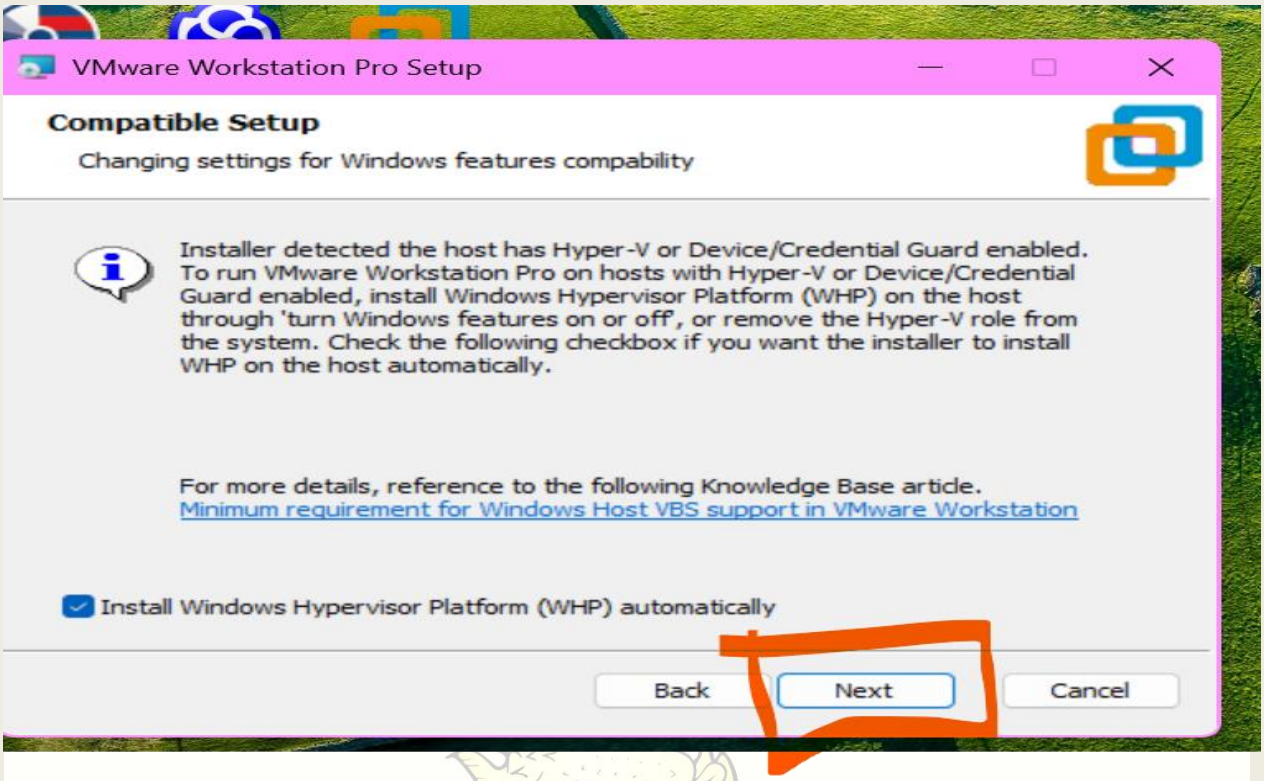
Run the installer:

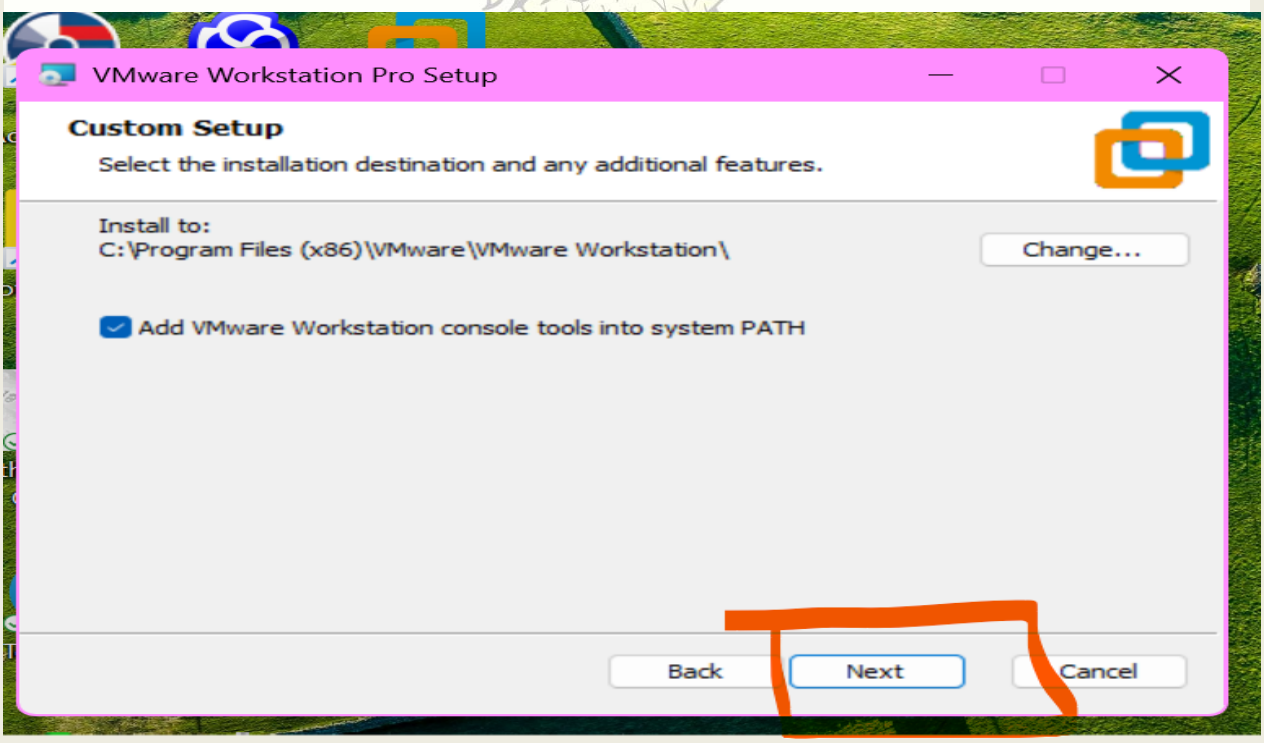
Now I downloaded it, I can run the installer. In windows, this involves clicking through a series of prompts. The installer will ask where to install virtual box and whether to create shortcuts. Stick with the default settings unless specific requirements.

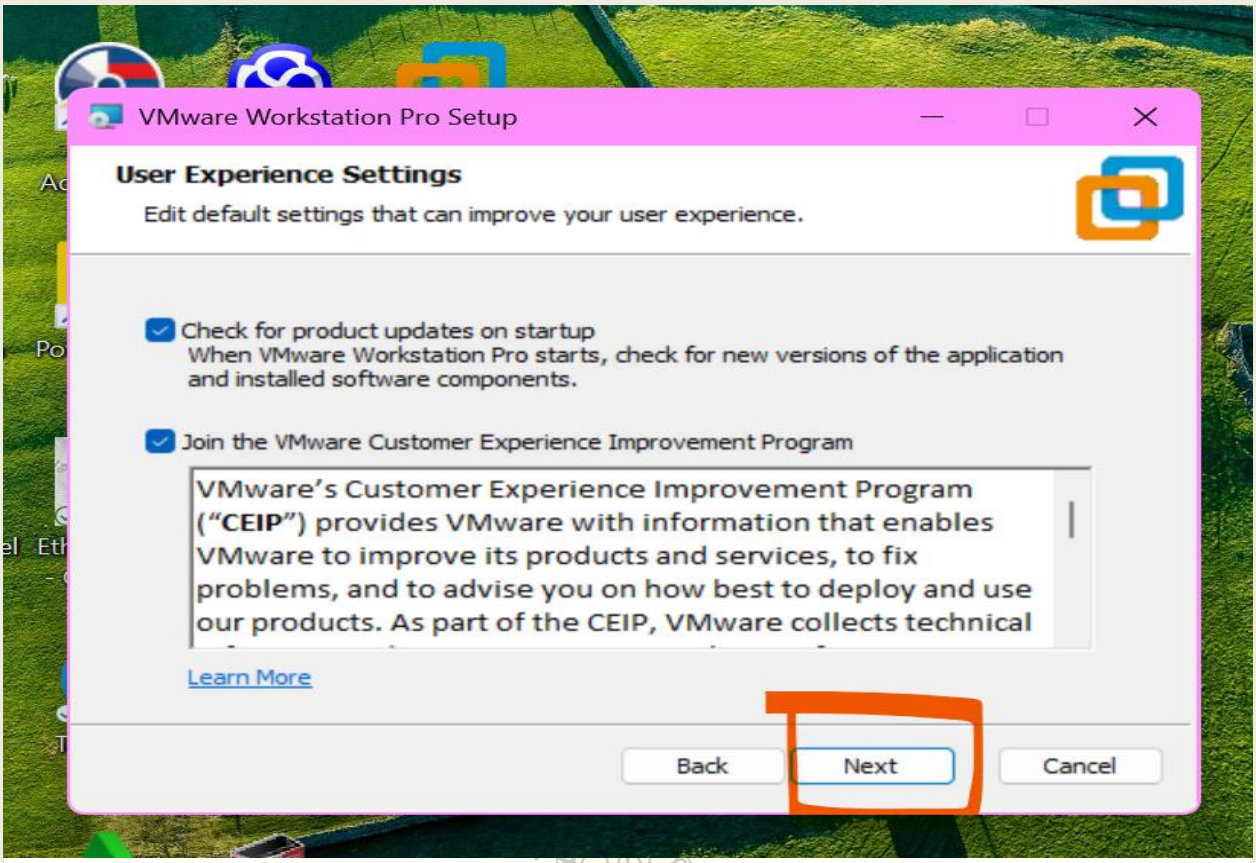


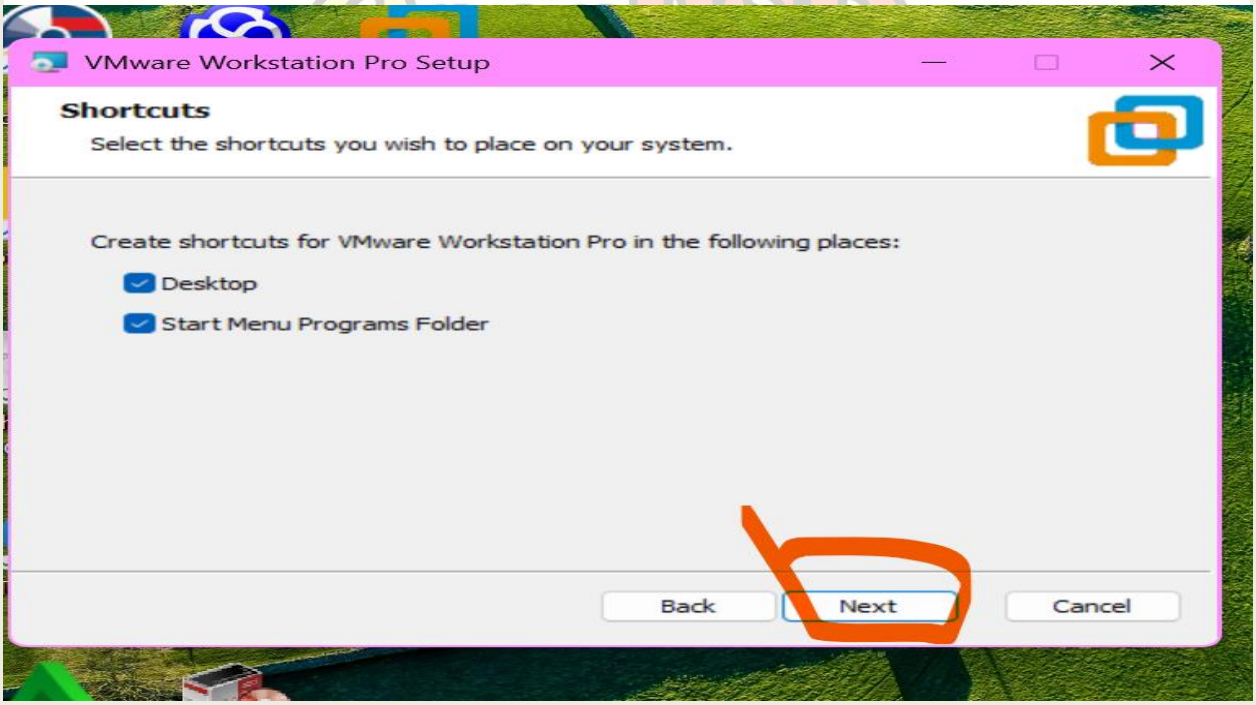

VMware Workstation Pro Setup
End-User License Agreement
Please read the following license agreement carefully.
Foundation Agreement
1. INTRODUCTION
I accept the terms in the License Agreement
Print
Back
Next
Cancel

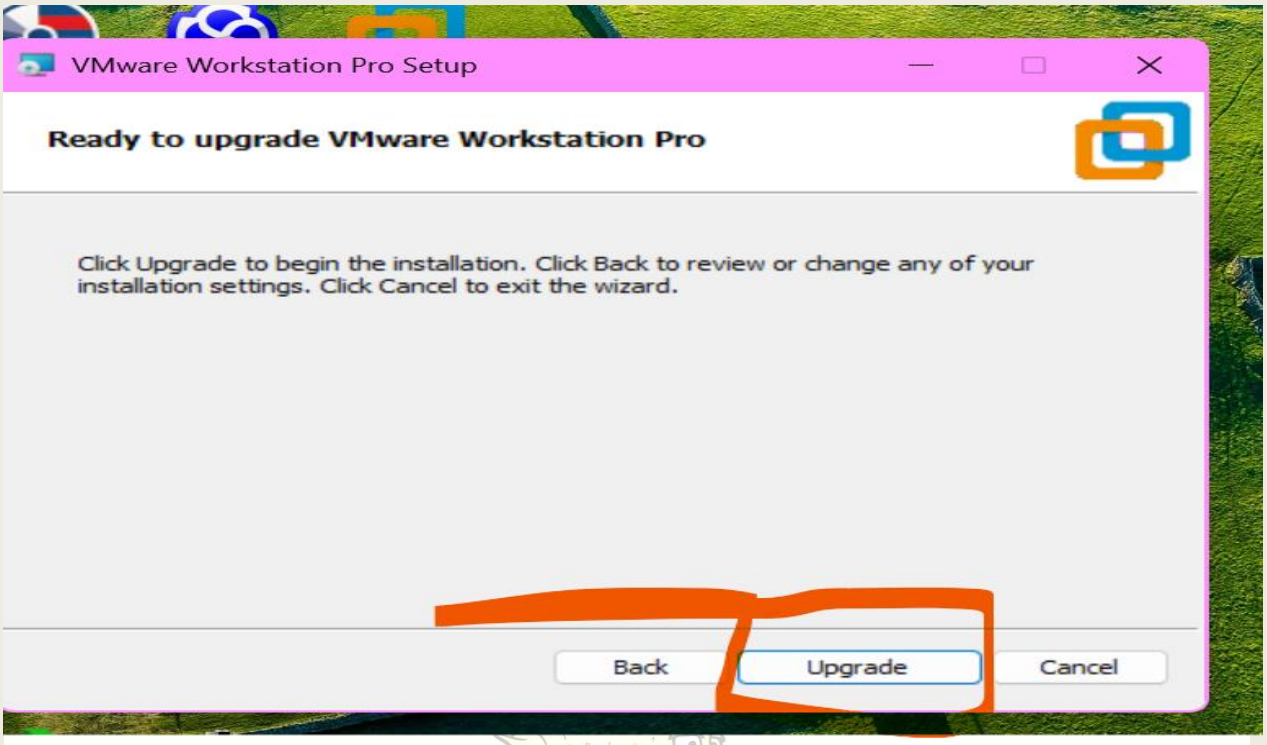

VMware Workstation Pro Setup
Compatible Setup
Changing settings for Windows features compability
Installer detected the host has Hyper-V or Device/Credential Guard enabled. To run VMware Workstation Pro on hosts with Hyper-V or Device/Credential Guard enabled, install Windows Hypervisor Platform (WHP) on the host through 'turn Windows features on or off', or remove the Hyper-V role from the system. Check the following checkbox if you want the installer to install WHP on the host automatically.
For more details, reference to the following Knowledge Base article.
Minimum requirement for Windows Host VBS support in VMware Workstation
Install Windows Hypervisor Platform (WHP) automatically
Back
Next
Cancel

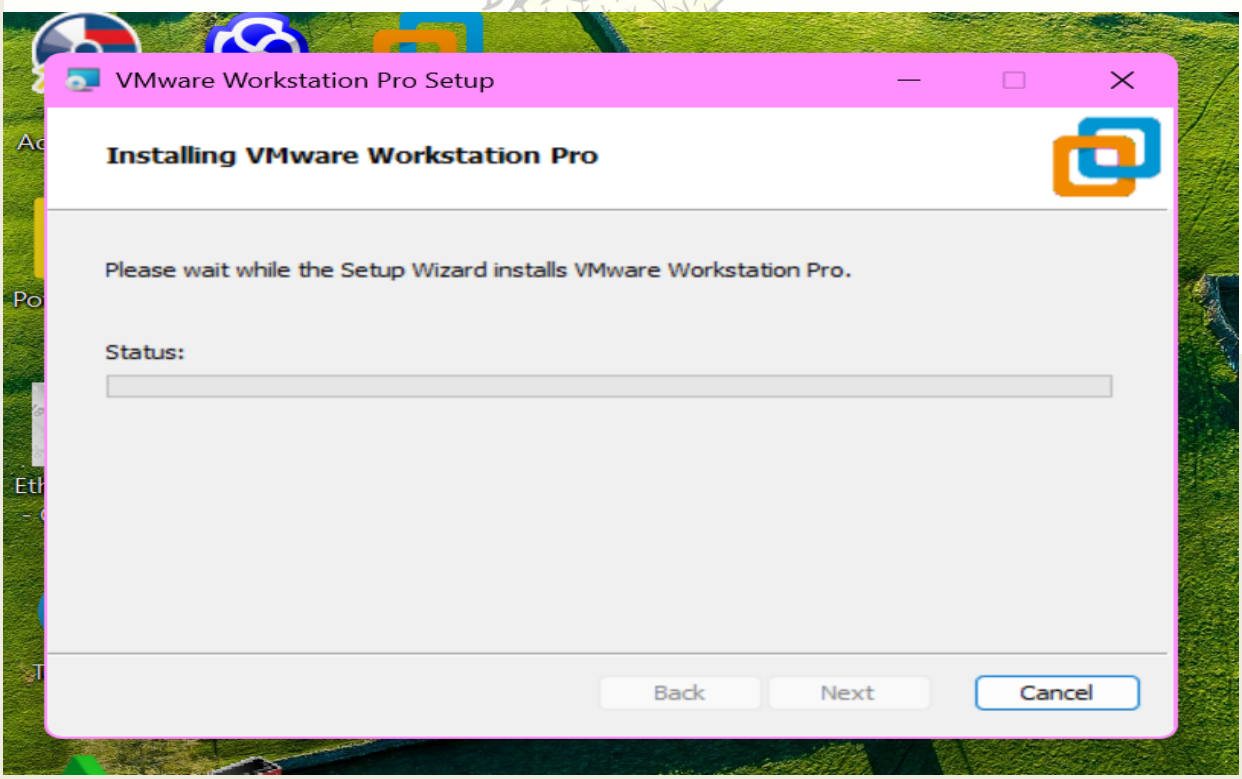

VMware Workstation Pro Setup
Compatible Setup
Changing settings for Windows features compability
Installer detected the host has Hyper-V or Device/Credential Guard enabled. To run VMware Workstation Pro on hosts with Hyper-V or Device/Credential Guard enabled, install Windows Hypervisor Platform (WHP) on the host through 'turn Windows features on or off', or remove the Hyper-V role from the system. Check the following checkbox if you want the installer to install WHP on the host automatically.
For more details, reference to the following Knowledge Base article.
Minimum requirement for Windows Host VBS support in VMware Workstation
Install Windows Hypervisor Platform (WHP) automatically
Back
Next
Cancel

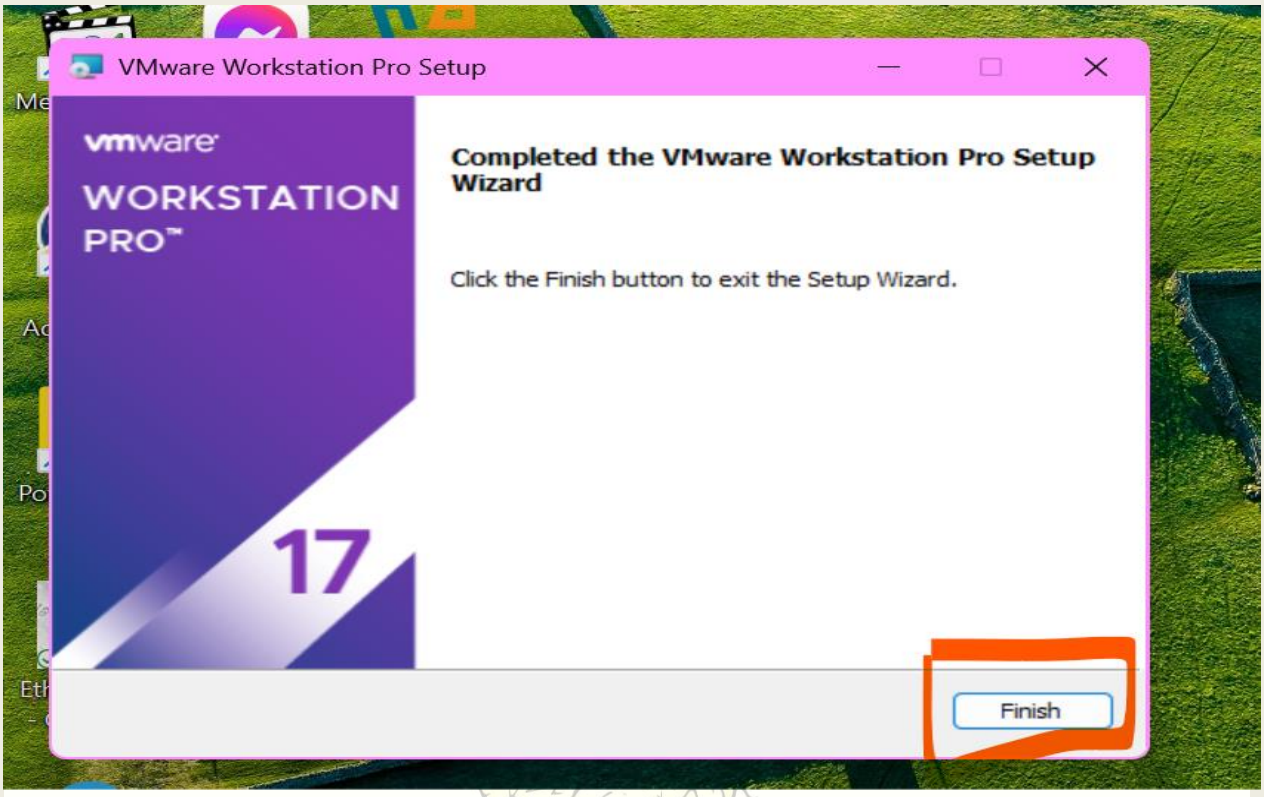

VMware Workstation Pro Setup
Custom Setup
Select the installation destination and any additional features.
Install to:
C:\Program Files (x86)\VMware\VMware Workstation\
Change...
Add VMware Workstation console tools into system PATH
Back
Next
Cancel


VMware Workstation Pro Setup
User Experience Settings
Edit default settings that can improve your user experience.
Check for product updates on startup
When VMware Workstation Pro starts, check for new versions of the application and installed software components.
Join the VMware Customer Experience Improvement Program
VMware's Customer Experience Improvement Program ("CEIP") provides VMware with information that enables VMware to improve its products and services, to fix problems, and to advise you on how best to deploy and use our products. As part of the CEIP, VMware collects technical
Learn More
Back
Next
Cancel


VMware Workstation Pro Setup
Shortcuts
Select the shortcuts you wish to place on your system.
Create shortcuts for VMware Workstation Pro in the following places:
Desktop
Start Menu Programs Folder
Back
Next
Cancel

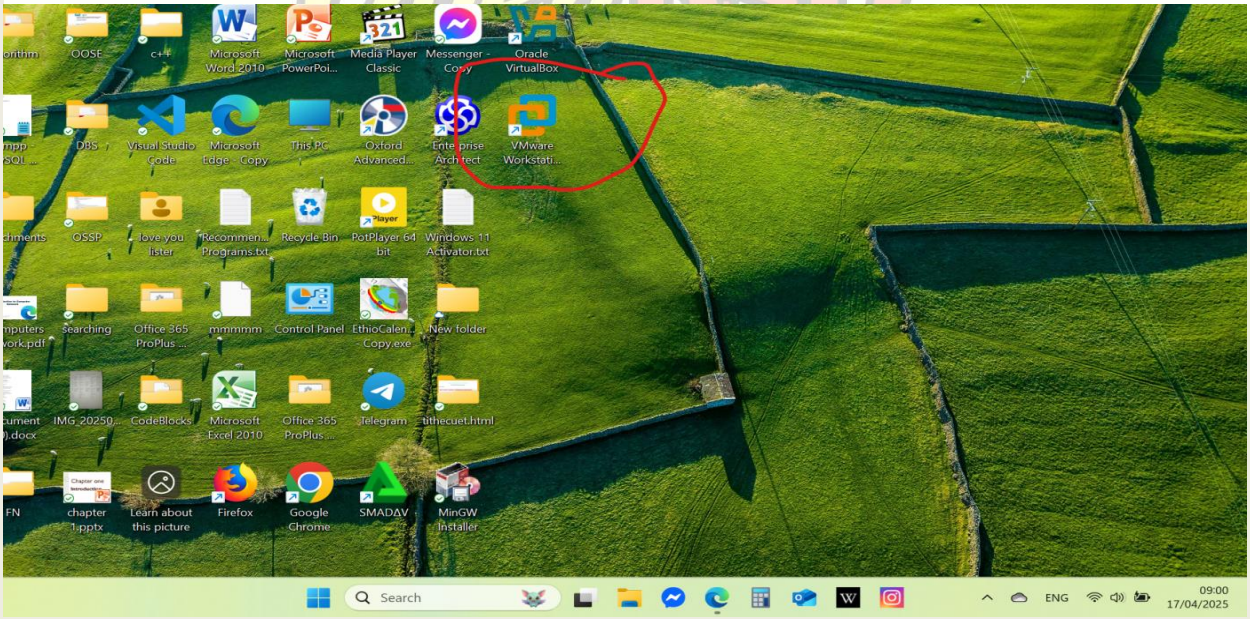

VMware Workstation Pro Setup
Ready to upgrade VMware Workstation Pro
Click Upgrade to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.
Back
Upgrade
Cancel

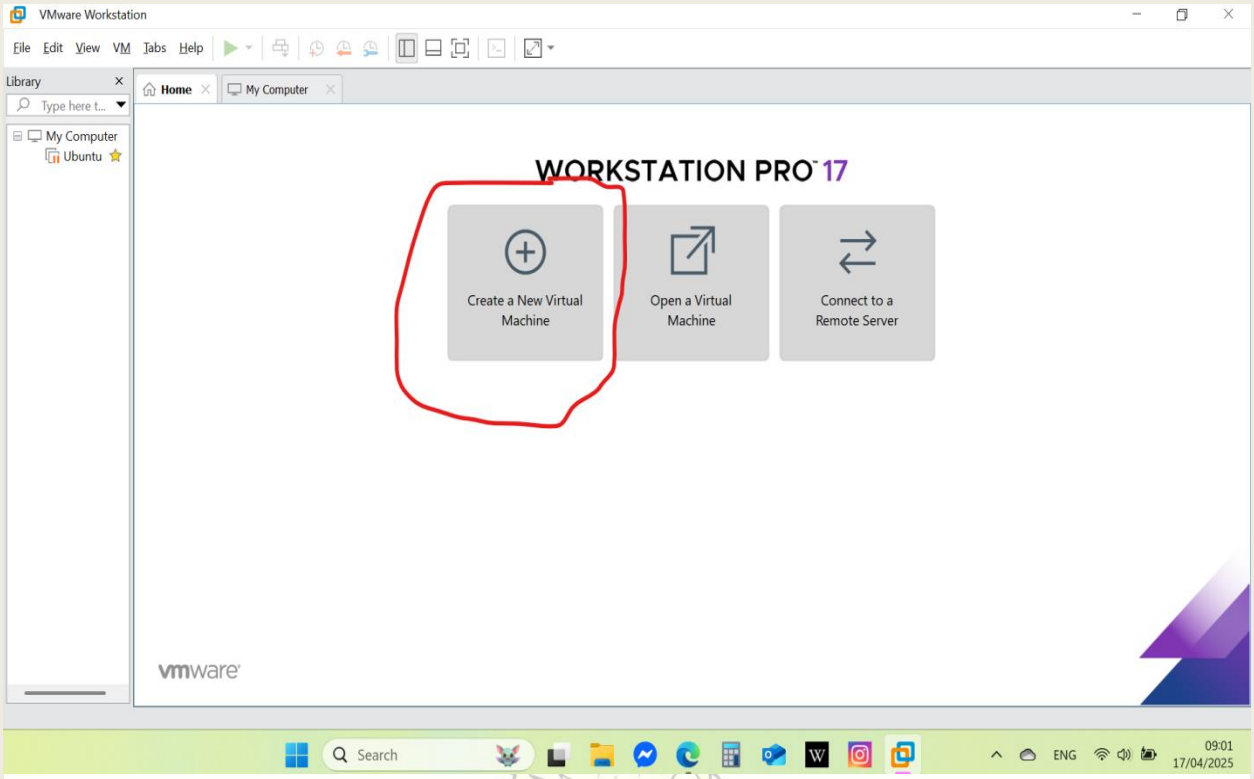

VMware Workstation Pro Setup
Installing VMware Workstation Pro
Please wait while the Setup Wizard installs VMware Workstation Pro.
Status:
Back
Next
Cancel



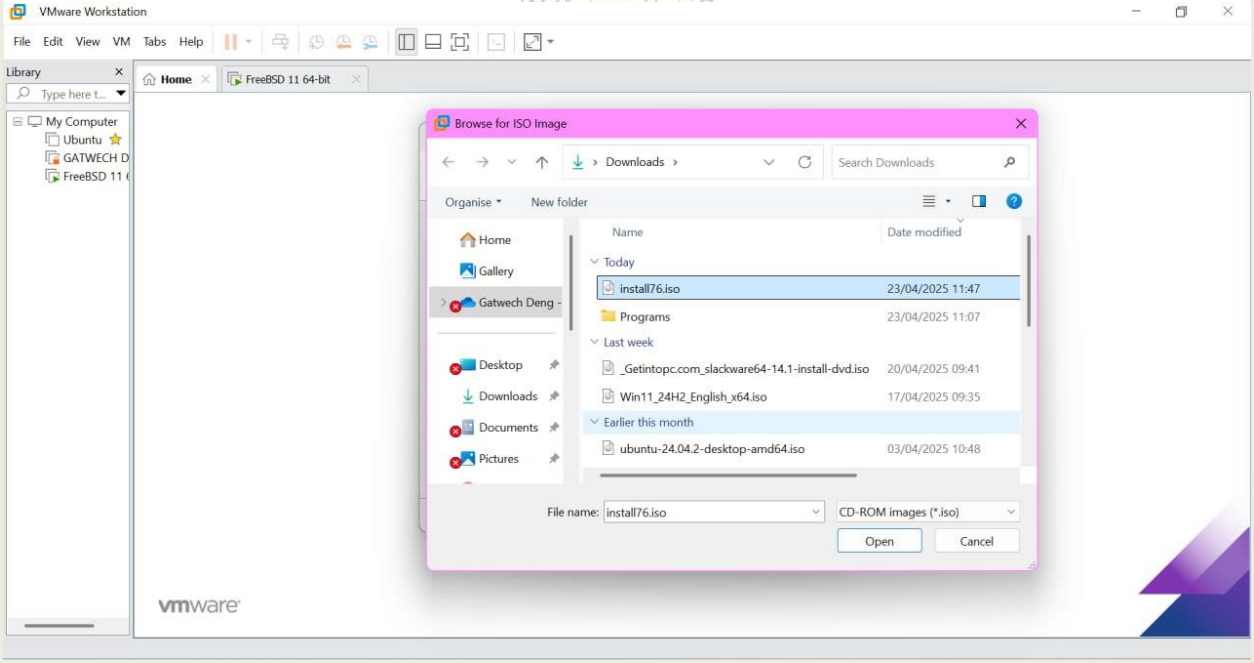
i. Snipped Images Placeholder

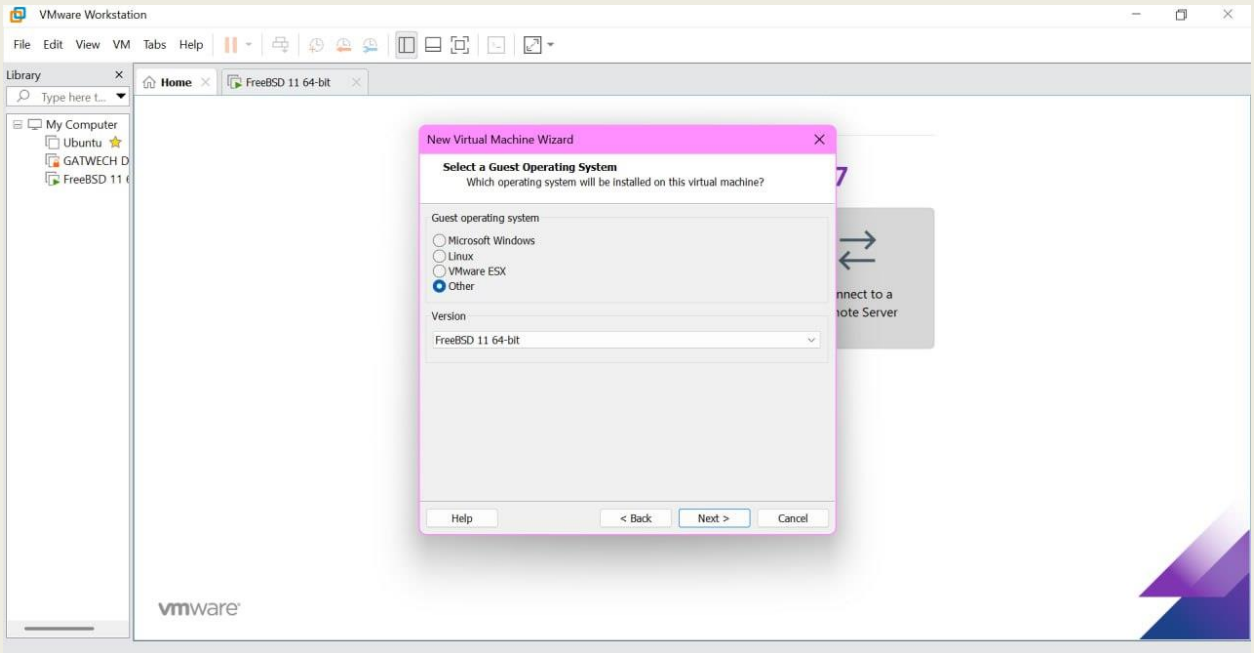
1. **Open VMware Workstation** → New Virtual Machine



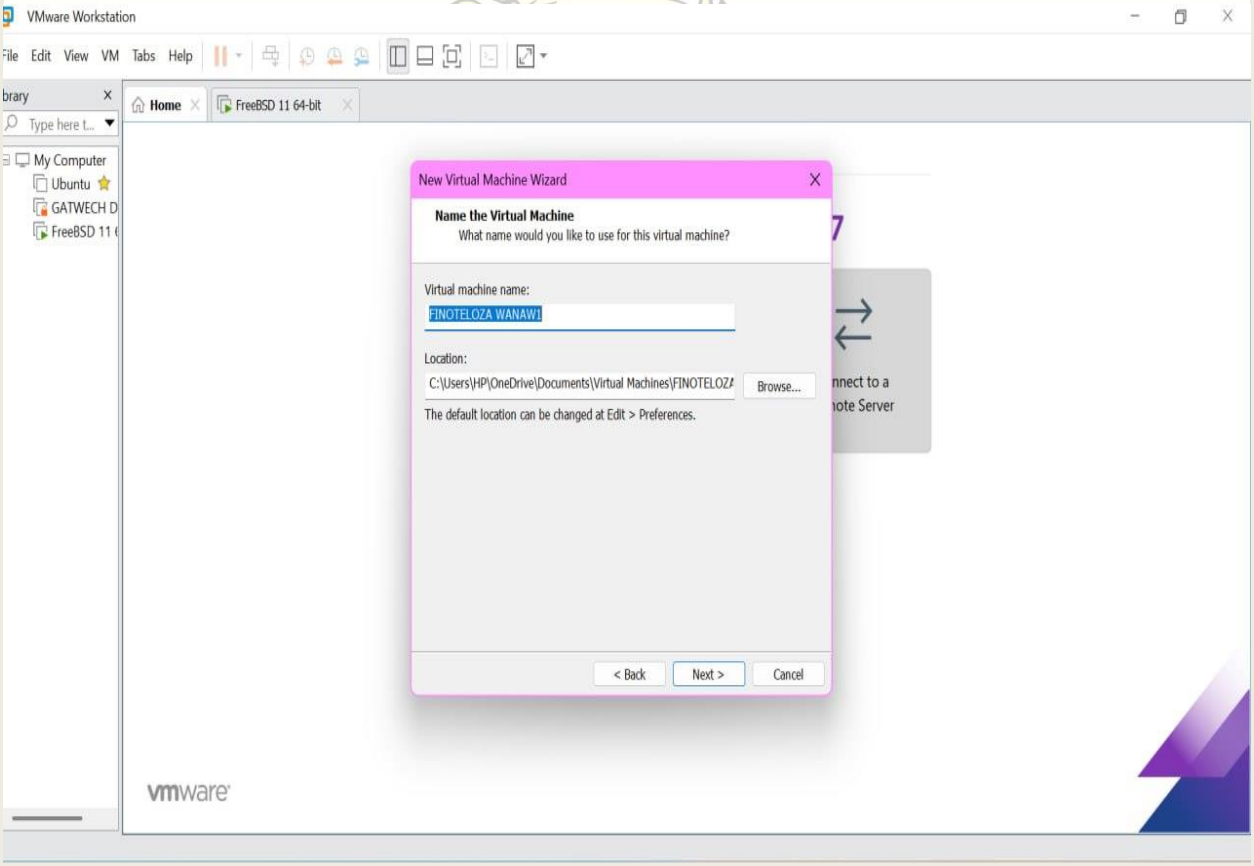


2. **Select "Installer disc image file (iso)"** → Browse and select Windows 11 ISO

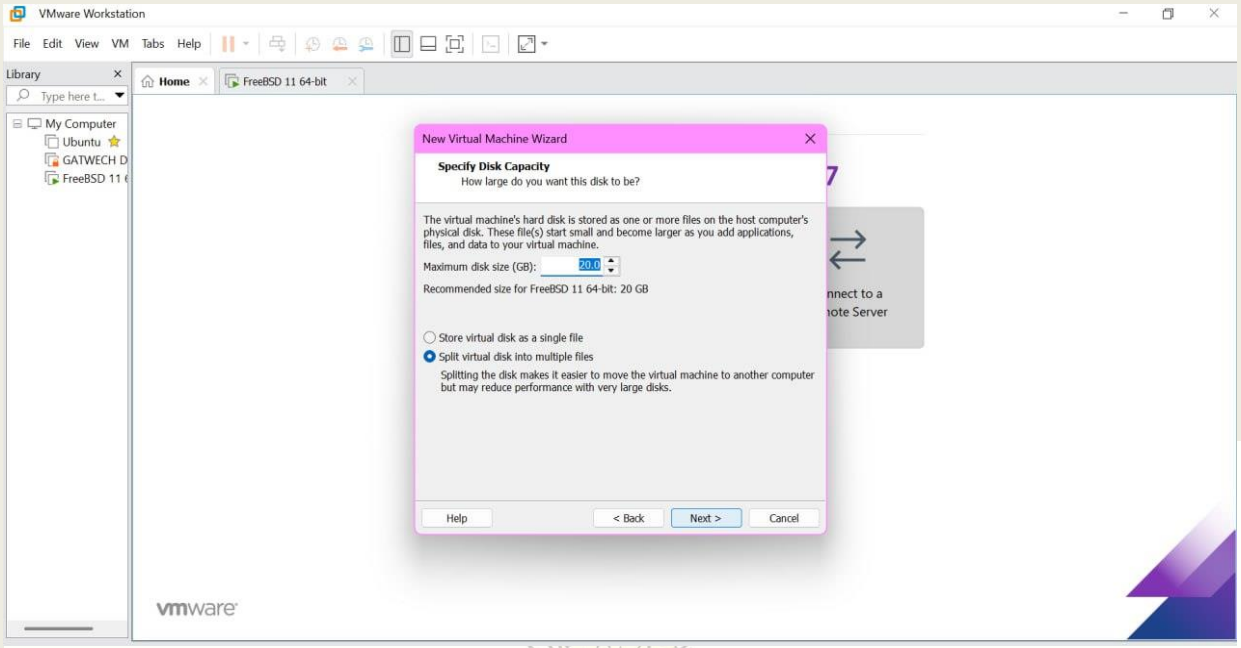




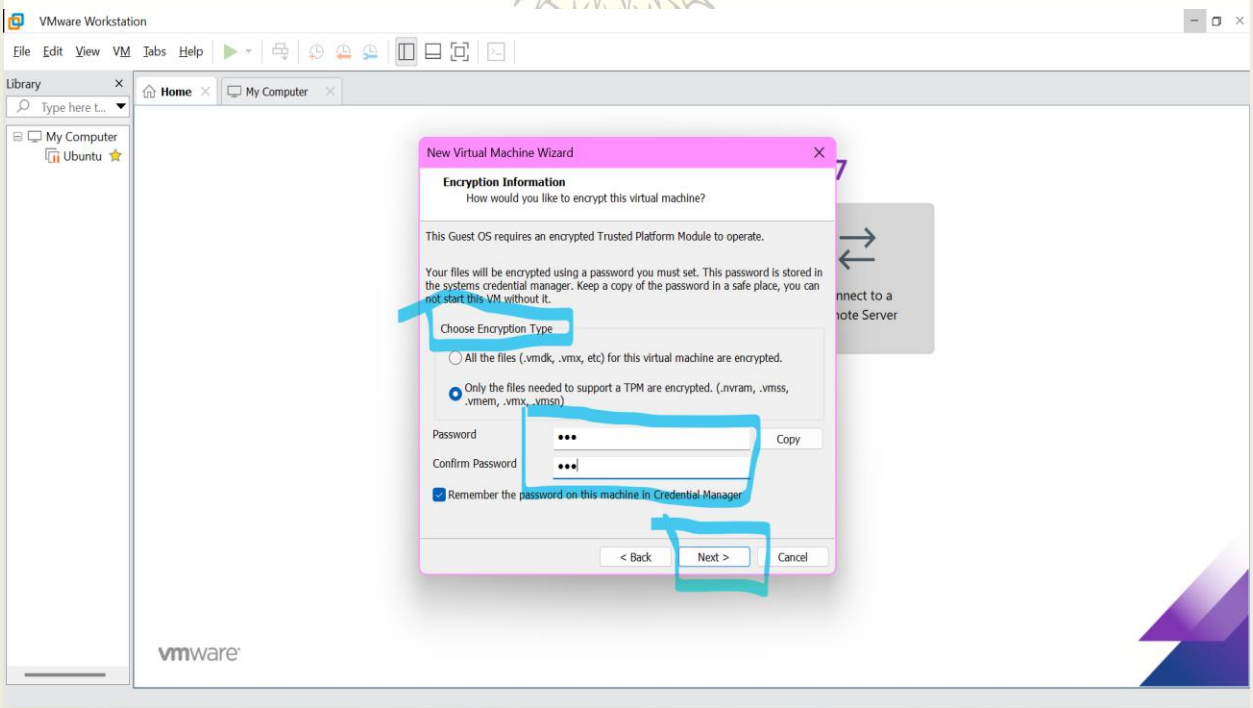
3. **Set VM name (FINOTELOZA WANAW)**



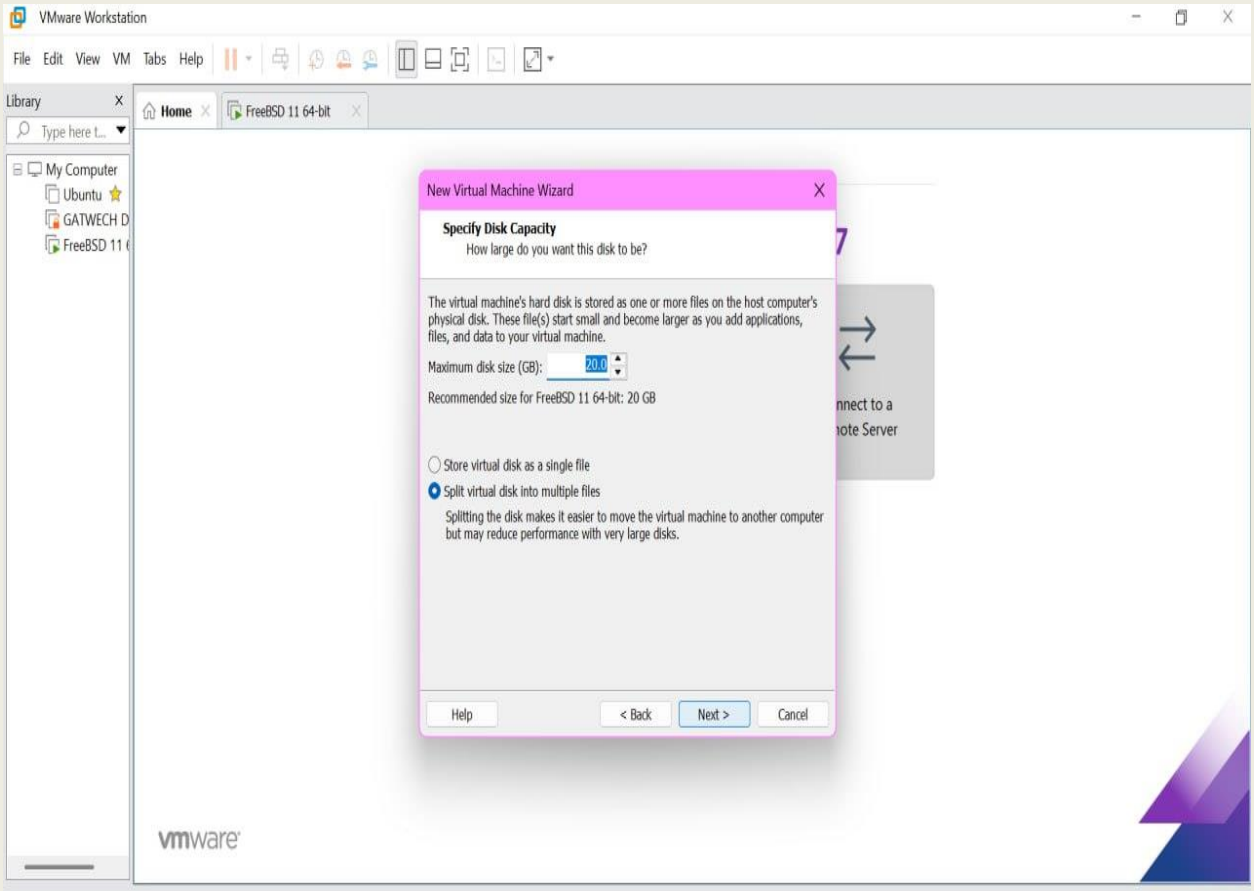
4. **Allocate RAM (4 GB or more)**



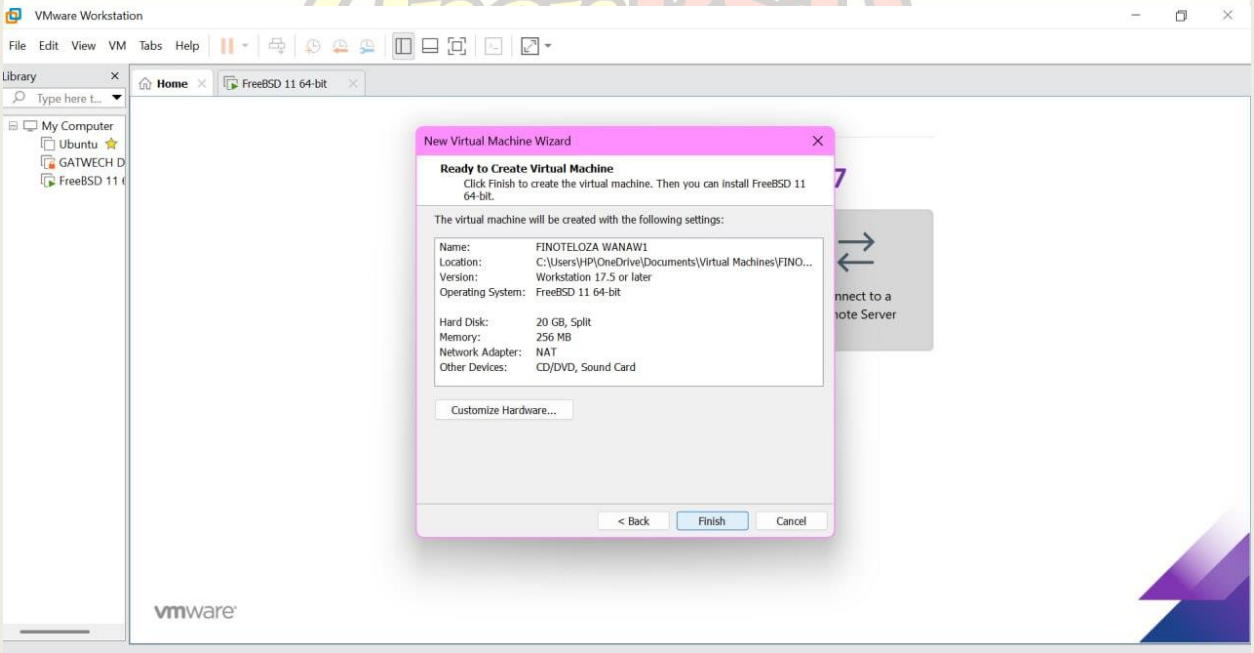
5. **Input password**



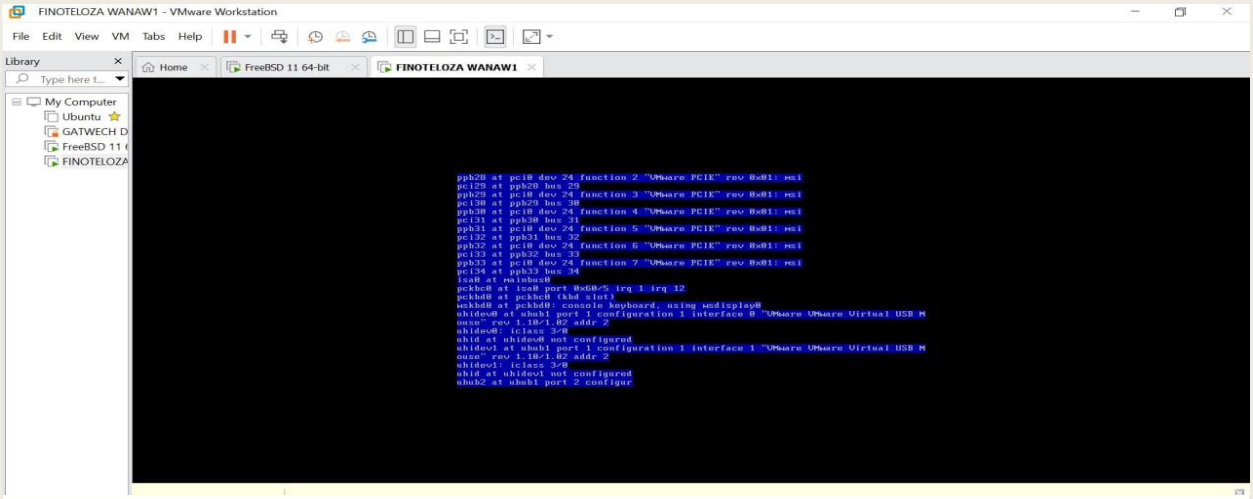
6. **Create a new virtual disk** → 20 GB recommended



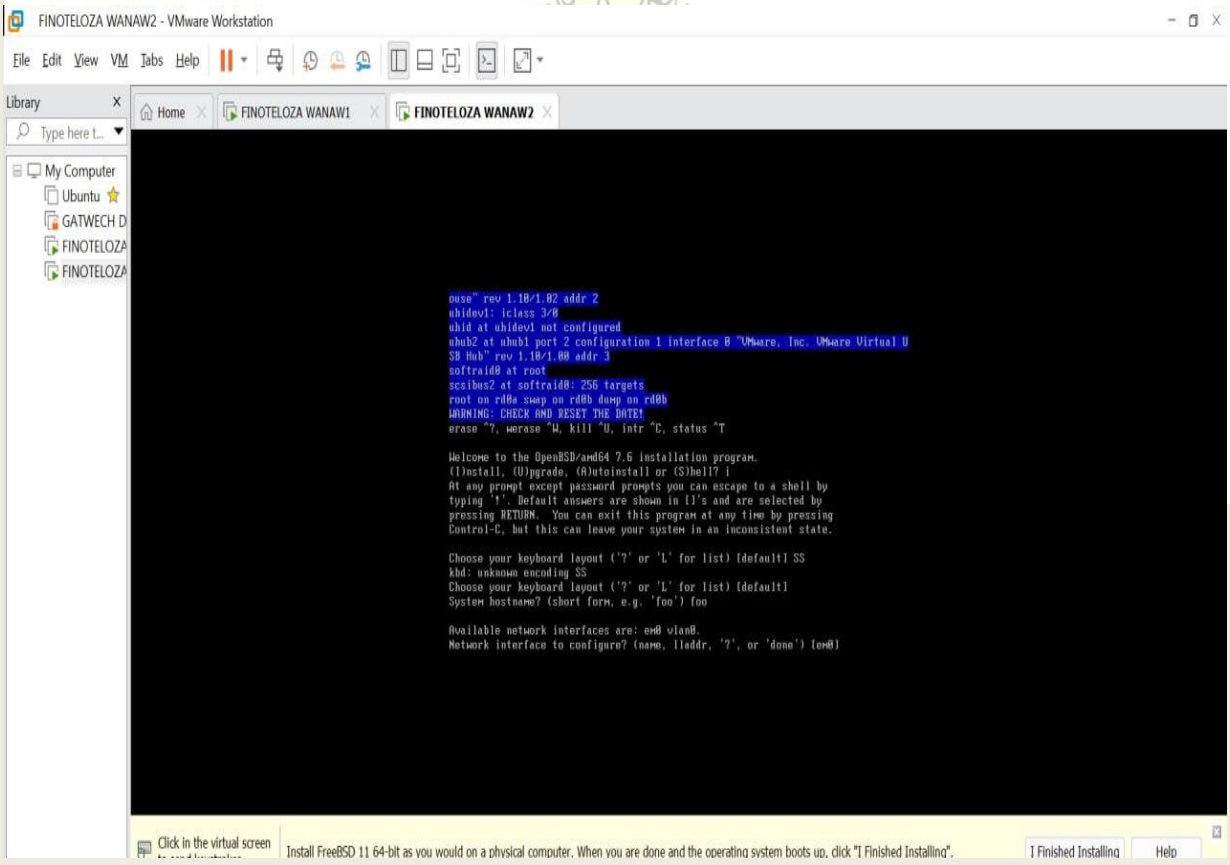
7. **Finish Setup** and power on VM



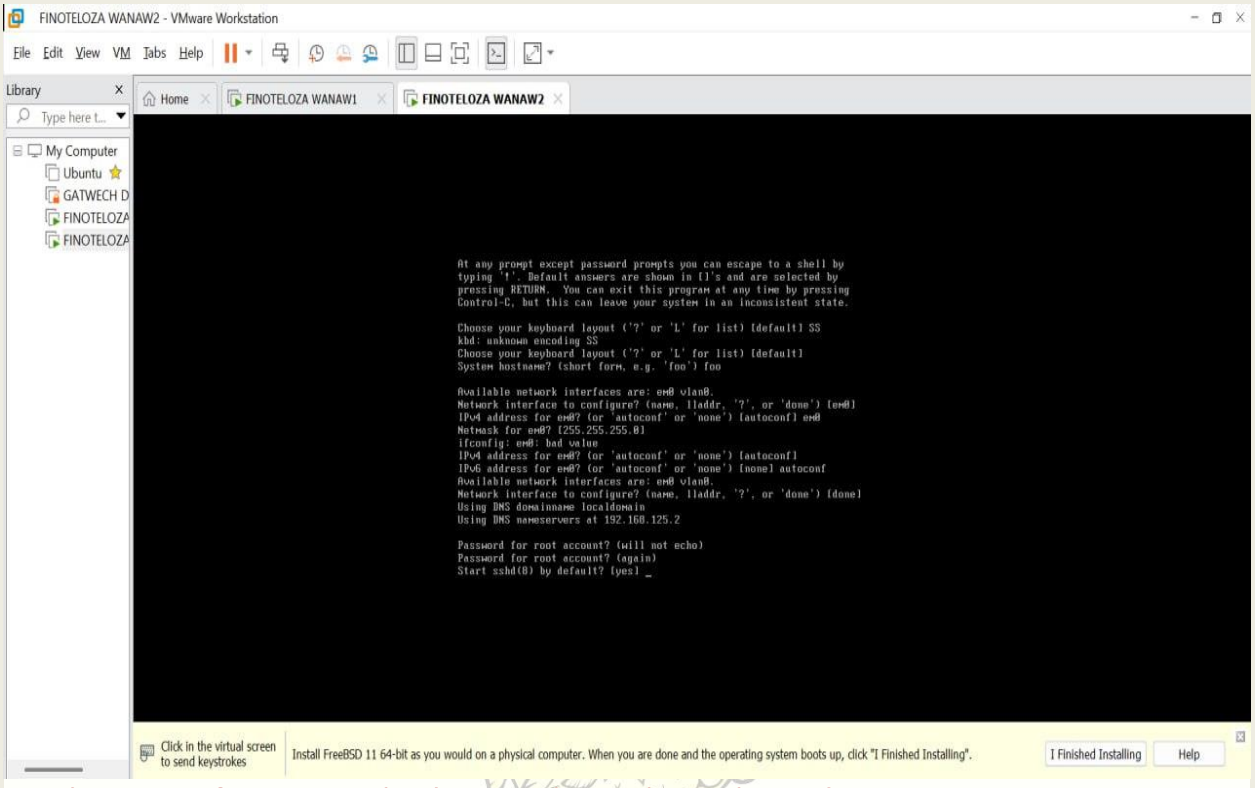
Now the OpenBSD is booting



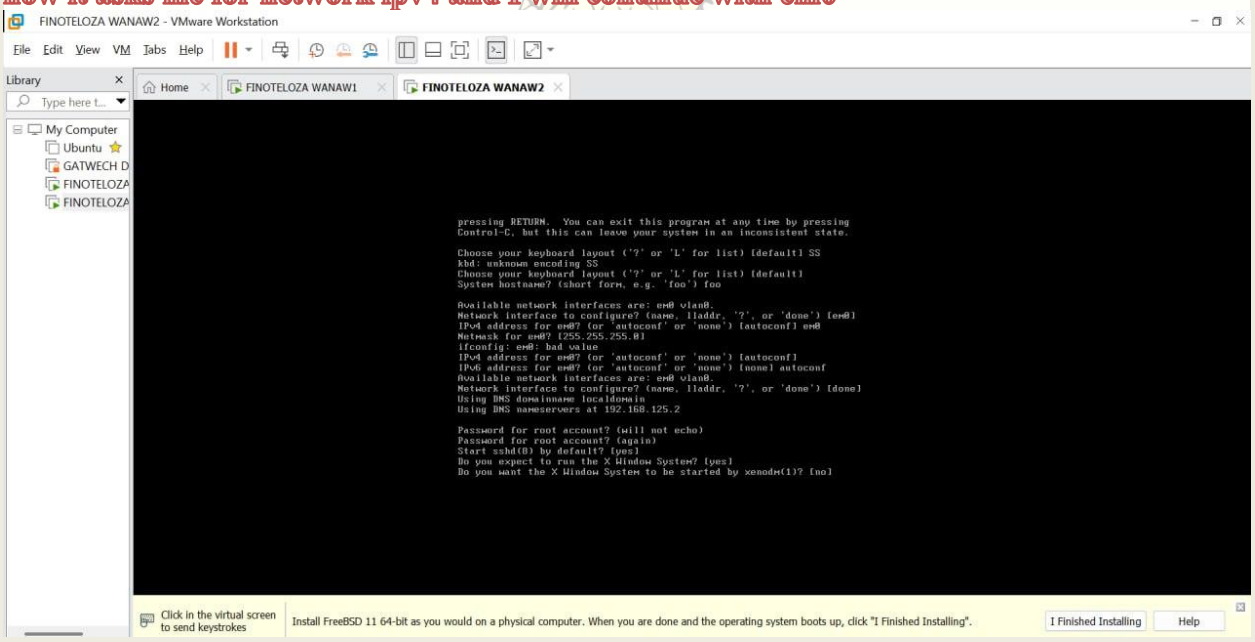
It will ask you to autoinstall and write I and click enter



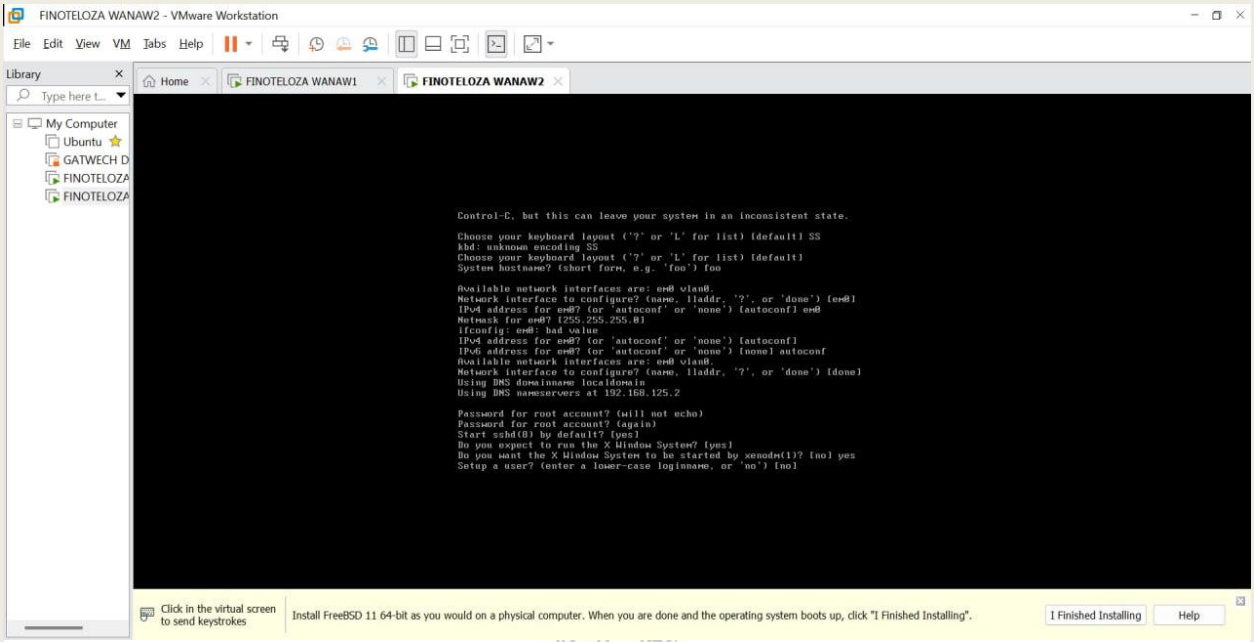
Now it ask me for the host name and will continue with default ("foo").



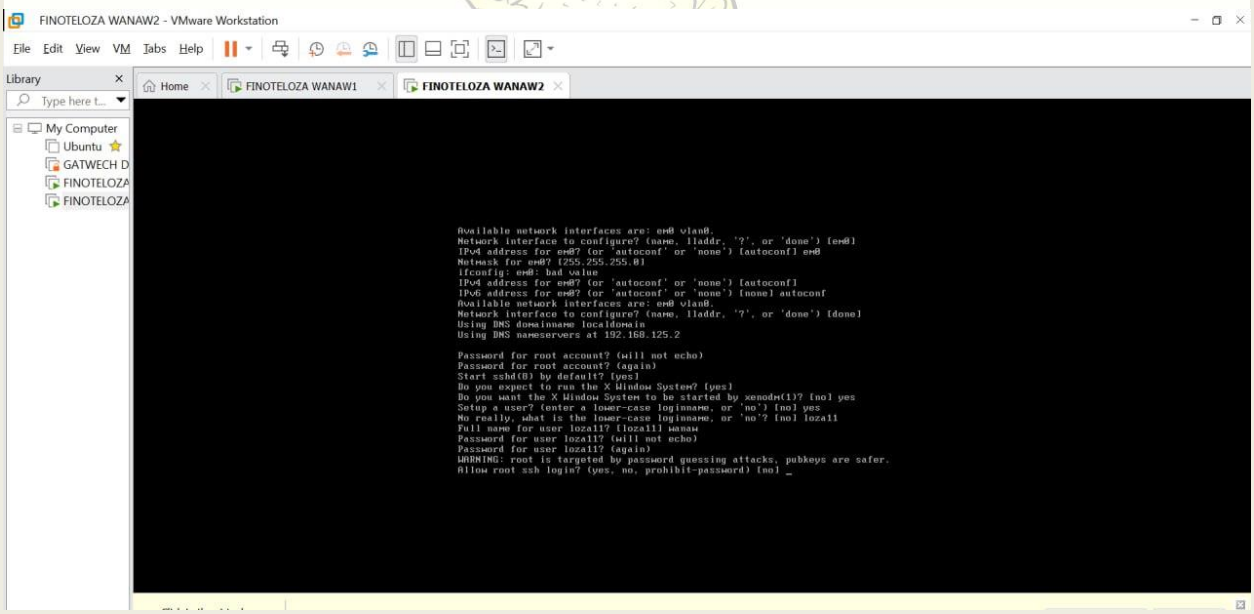
now it asks me for network ipv4 and I will continue with em0



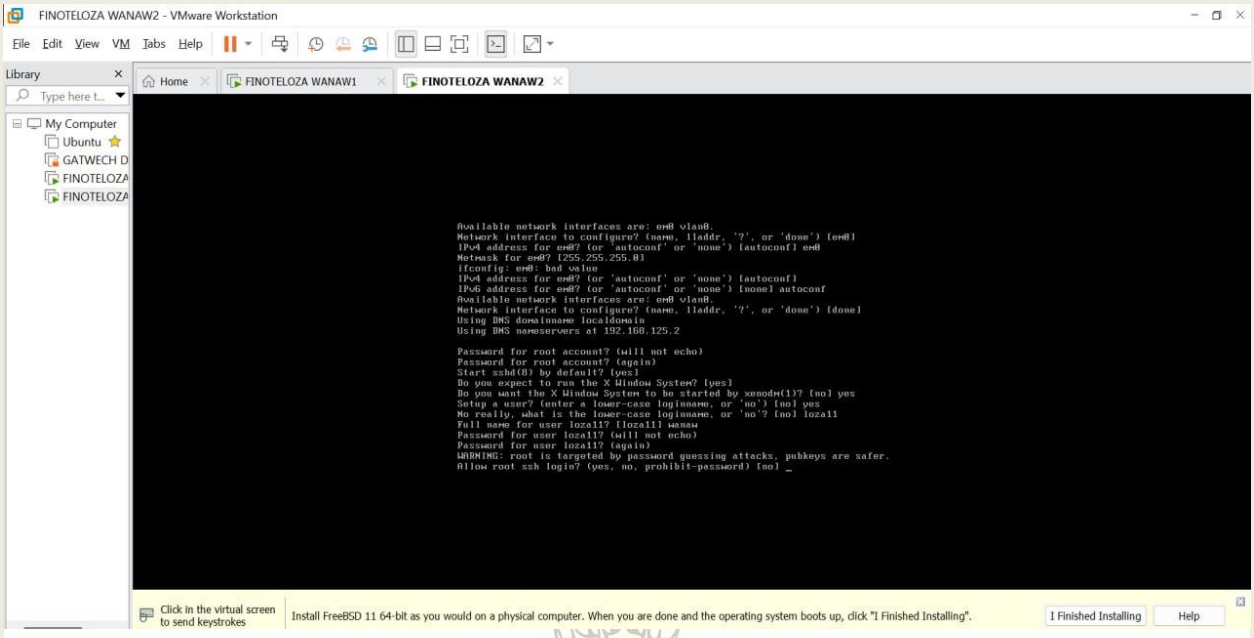
Now it ask me for ipv6 address and I will type the command autoconf and click enter



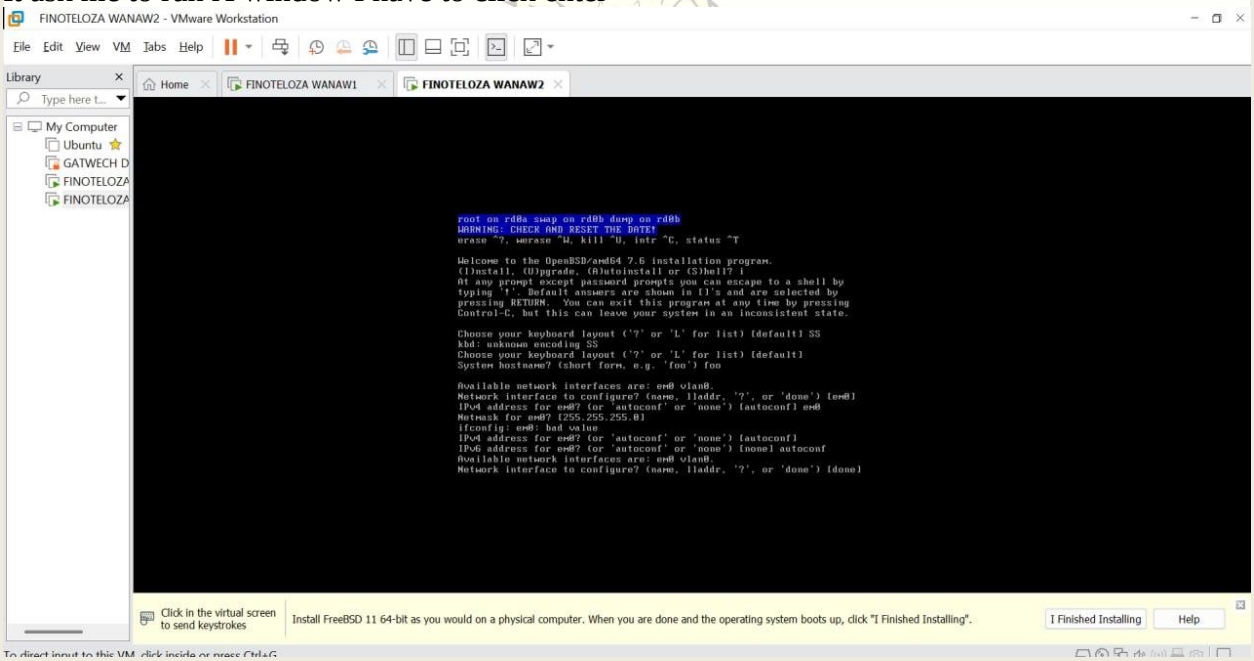
Now it ask me for the password of root account (I have to enter password)



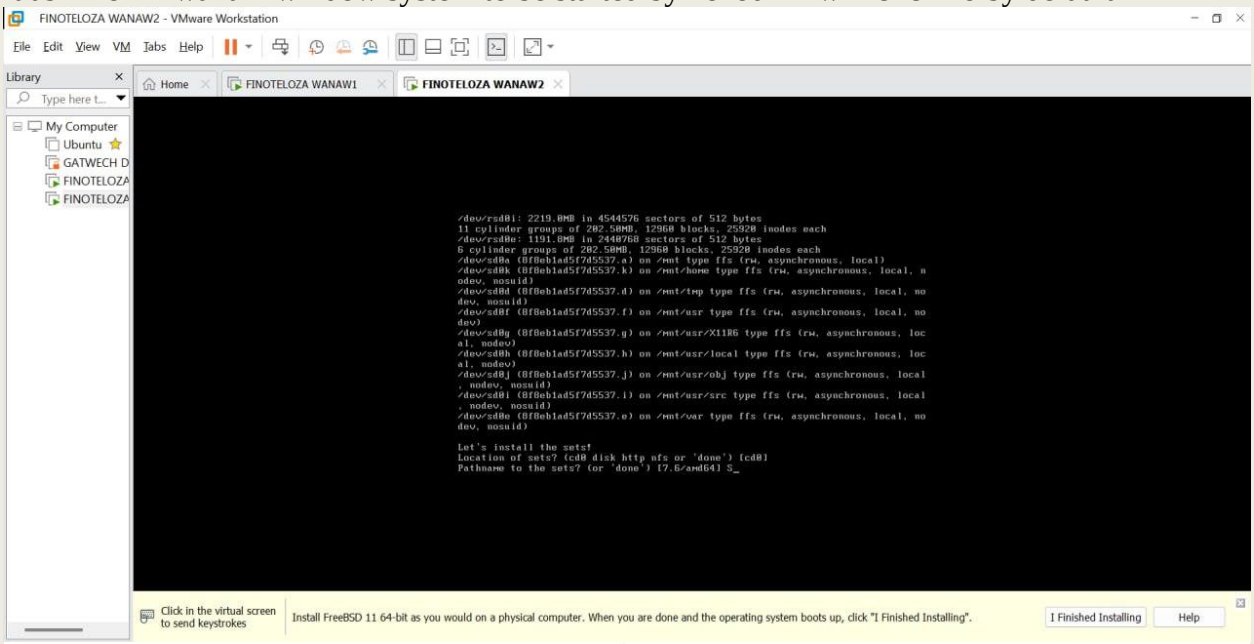
It ask me to start ssd by default and I have to click enter



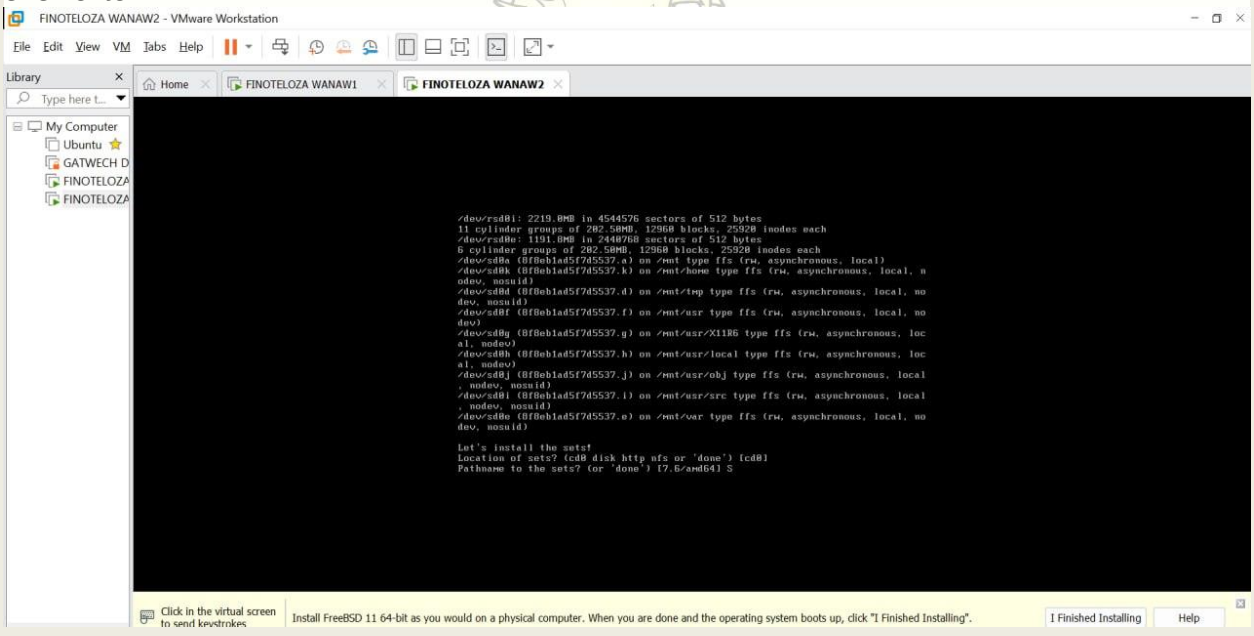
It ask me to run X window I have to click enter



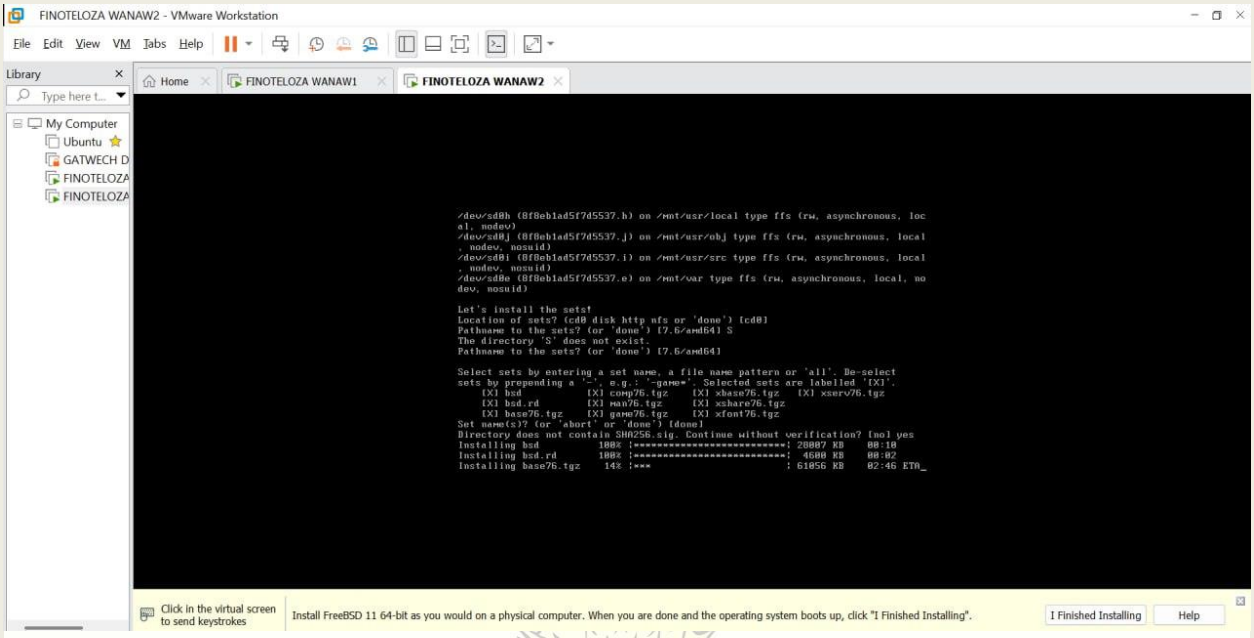
It ask me if I want X window system to be started by xenodm I will click no by default



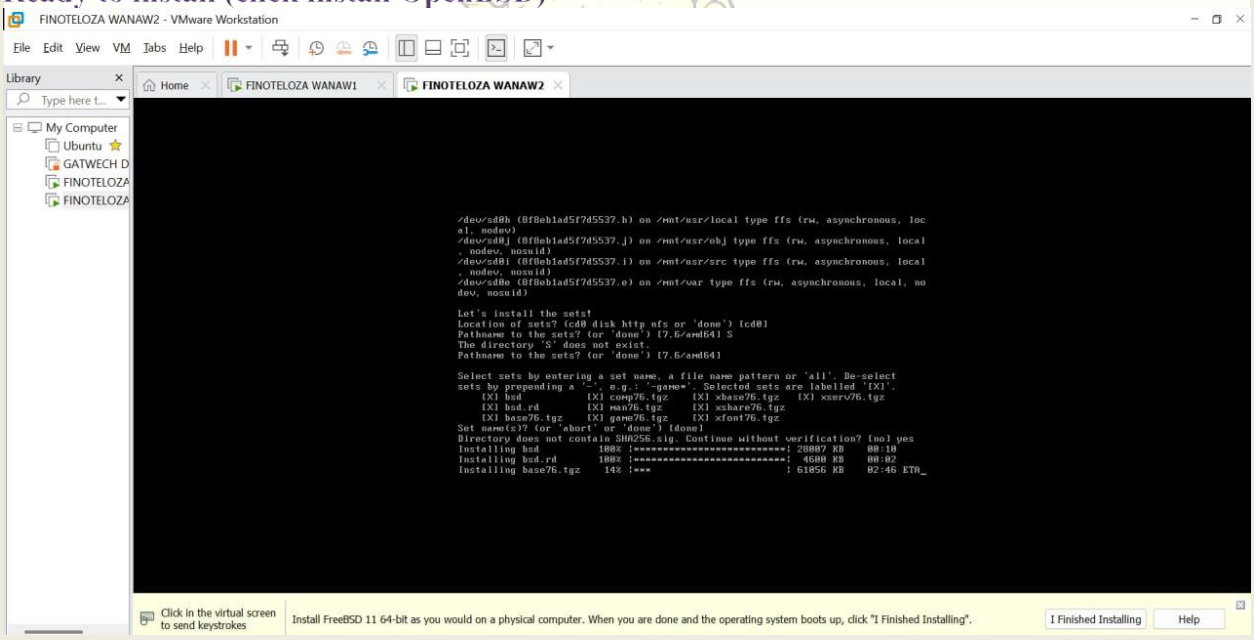
It ask me for the user name using lower case and I have to type finoteloza as my user name and click enter



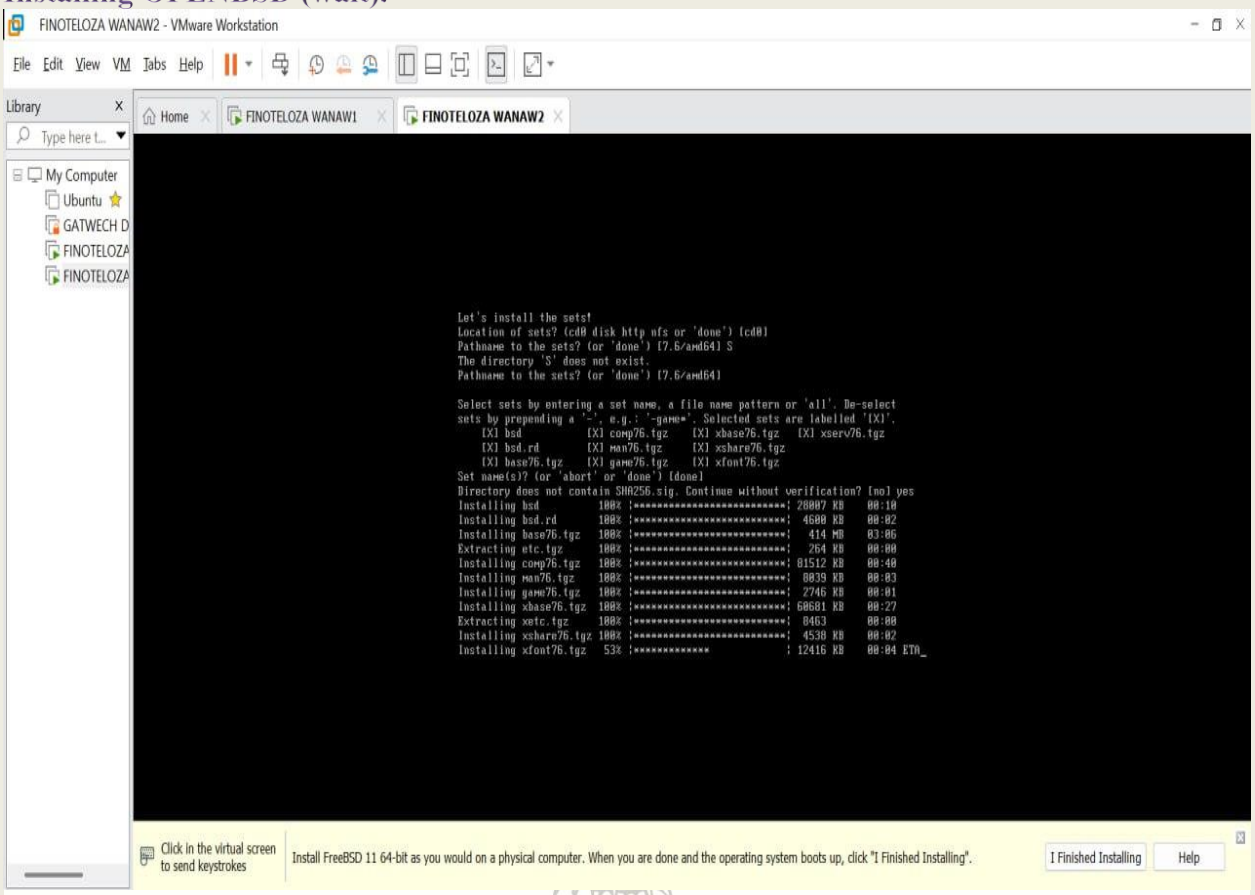
It ask for password of user name and I have to type my password (********) and click enter and now it is ready to install all application and supportive kernel interface



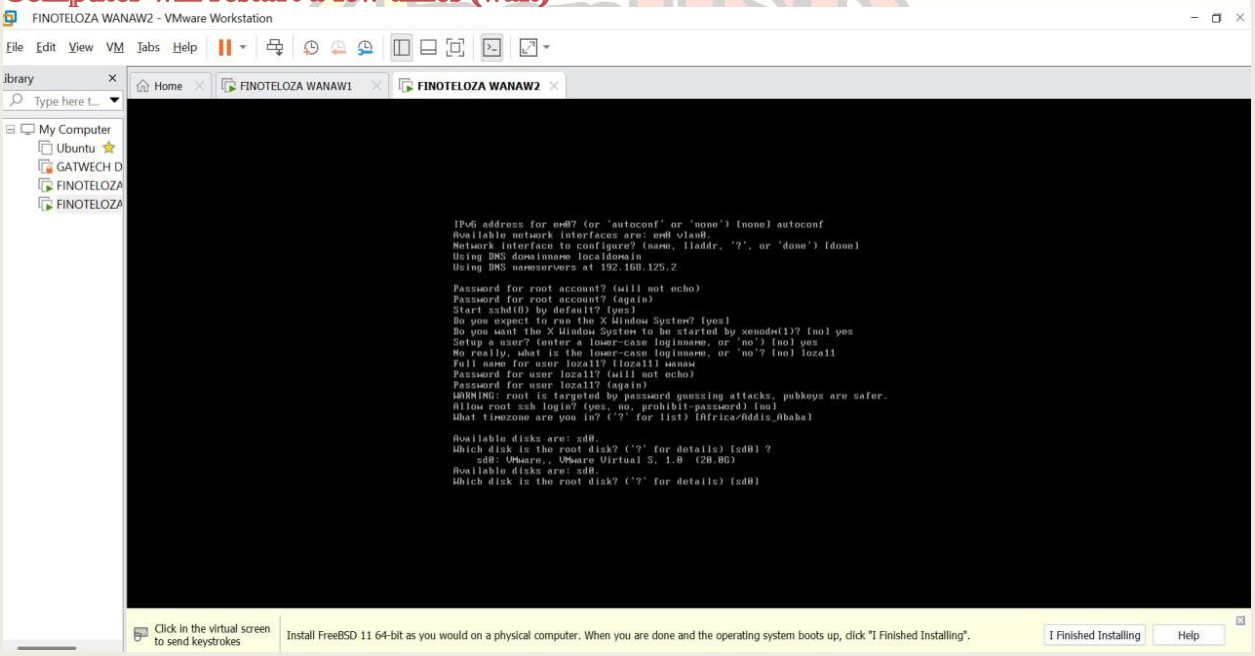
Ready to install (click install OpenBSD)



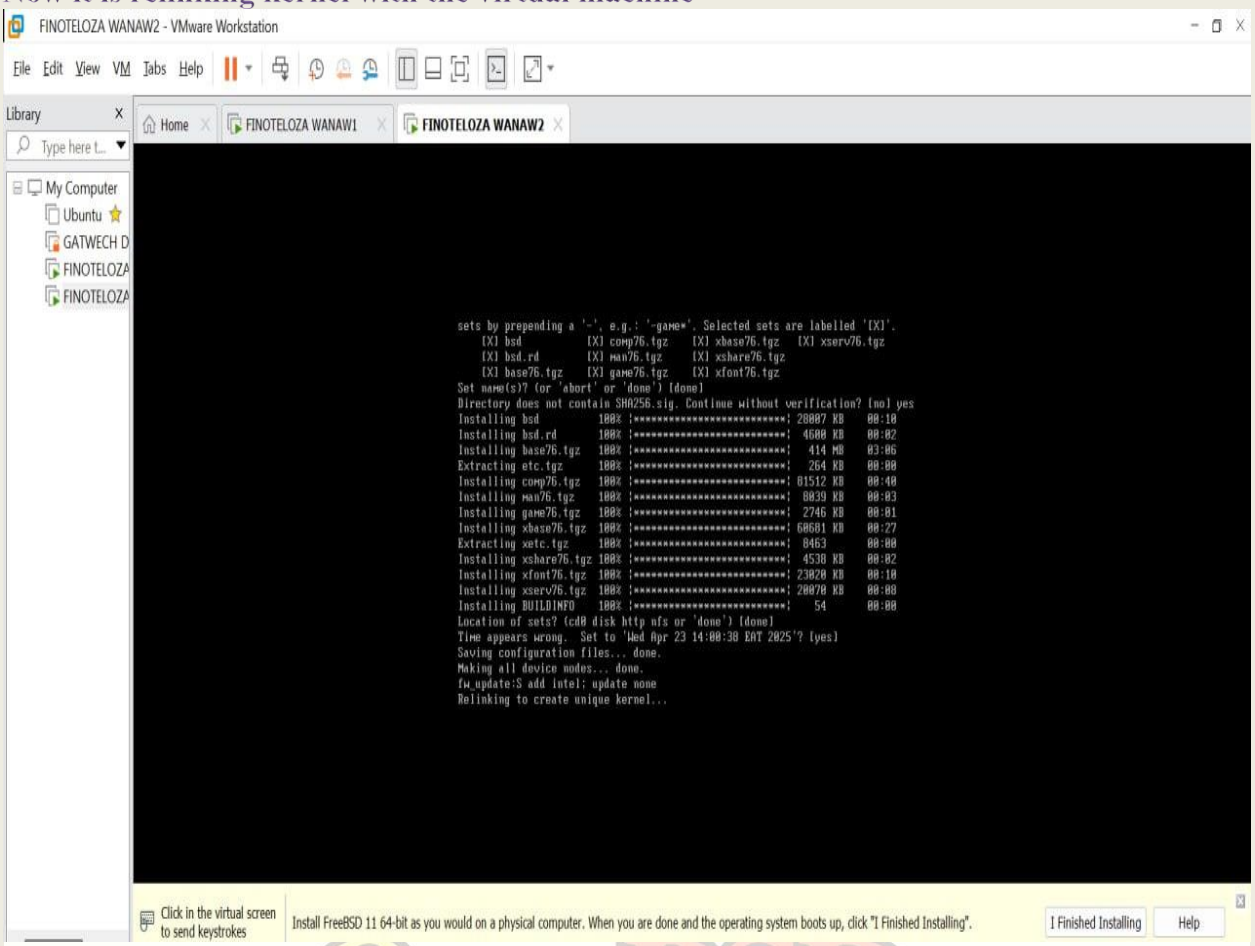
Installing OPENBSD (wait).



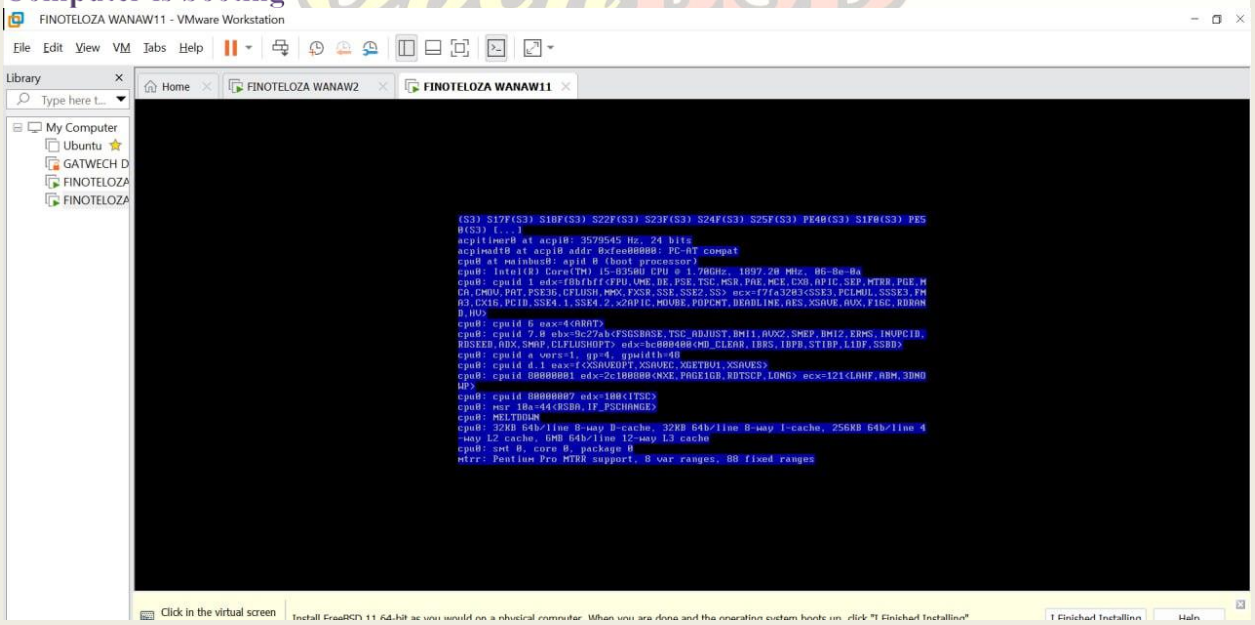
Computer will restart a few times (wait)



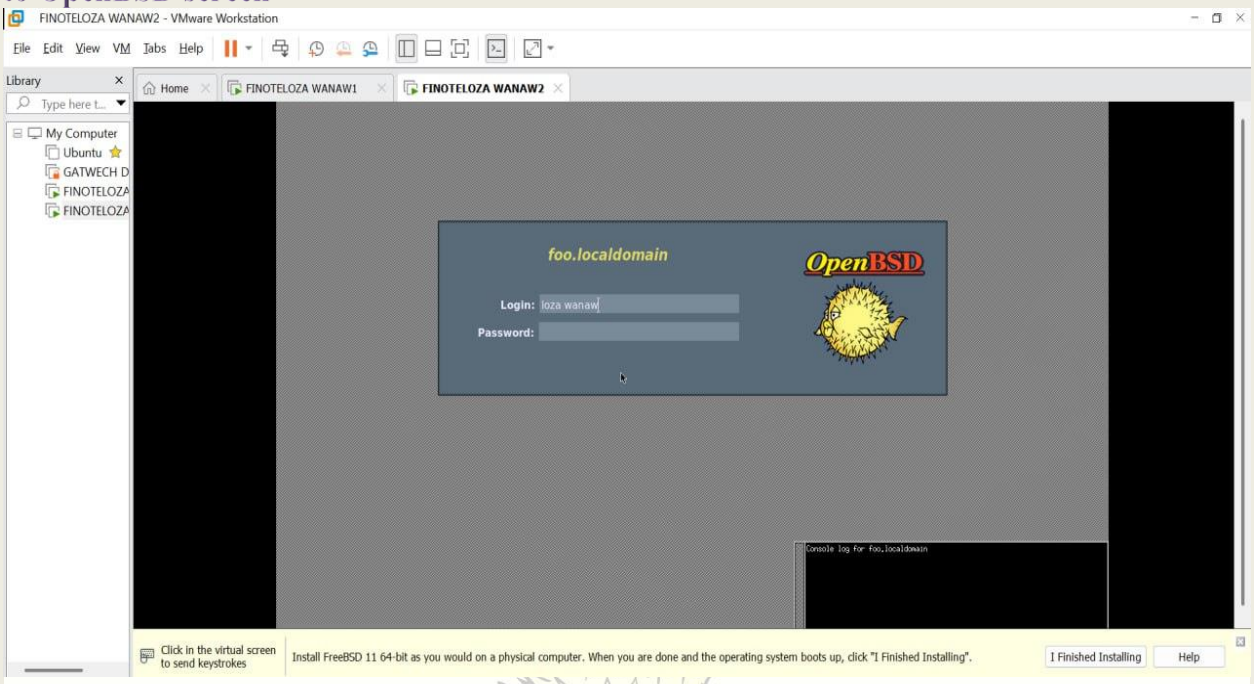
Now it is relinking kernel with the virtual machine



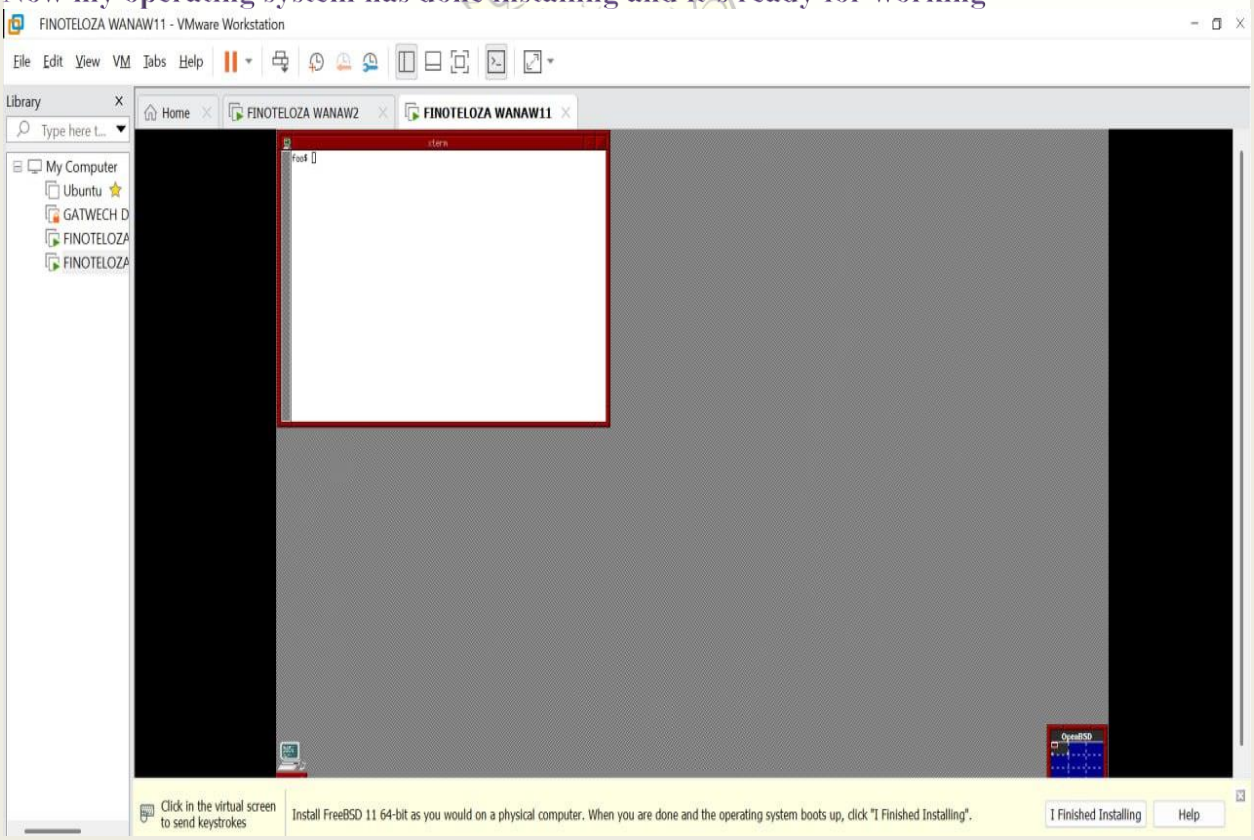
Computer is booting



It ask me for my user login info and password and I have to click enter and it will welcome to OpenBSD screen



Now my operating system has done installing and it's ready for working



E. ISSUES FACED

- ❗ Network adapter not detected during setup
- ❗ Keyboard layout confusion
- ❗ Unfamiliar disk partitioning UI

F. SOLUTIONS

- ☑ Network issue: Change network adapter settings to NAT or Bridged in VMware.
- ☑ Keyboard layout: Select correct layout or reconfigure post-install using kbd.
- ☑ Disk partitioning: Use auto layout or consult OpenBSD installation guide.

G. FILE SYSTEM SUPPORT

OpenBSD supports the following filesystems:

- **FFS (Fast File System)** – Default, fast, and stable.
- **MSDOS (FAT16/FAT32)** – Compatible with external drives and USBs.
- **ext2** – Read-only support for interoperability with Linux.

Why FFS?
FFS is tightly integrated into the OS and optimized for stability, performance, and compatibility.

H. FILE SYSTEM COMPARISON

Not Supported (or Limited Support)

Filesystem	Supported in OpenBSD?	Reason
NTFS	No (read-only with FUSE)	Not native; Microsoft proprietary
exFAT	No (needs FUSE)	Not supported out-of-the-box
ext4	No (partial read-only via ext2 driver)	Not reliable for modern Linux interoperability
Btrfs	No	Linux-specific, complex features not aligned with OpenBSD's simplicity
ZFS	No	Memory-heavy, more FreeBSD/Solaris focused
HFS+	No	Apple-specific, not widely needed
APFS	No	Apple's proprietary format, not open or portable

I. ADVANTAGES AND DISADVANTAGES

Advantages of OPENBSD on VMware:

- Strong security architecture
- Lightweight and fast
- Rich documentation and audit-friendly

Disadvantages:

- Limited GUI and third-party software
- Not beginner-friendly
- Less hardware compatibility out-of-the-box

J. CONCLUSION

OpenBSD installation in VMware is a great learning experience. Though the OS presents a steeper learning curve, it offers unmatched security and simplicity. With some guidance, it's an excellent platform for networking and system programming projects.

K. FUTURE OUTLOOK / RECOMMENDATION

- Use OpenBSD in firewall and security-focused environments
- Try using **VirtualBox** or **QEMU** as alternatives
- Contribute to documentation or build small projects using OpenBSD base

2. Virtualization – What, why, and How

What is Virtualization?

Virtualization is the creation of a virtual instance of hardware or software, allowing one physical system to run multiple isolated OS environments.

Why Use Virtualization?

- Testing without altering host system
- Efficient use of hardware
- Quick recovery and snapshotting
- Run different OS on one machine

How It Works:

A **hypervisor** (like VMware) abstracts physical resources and provides virtual hardware to guest OS. There are:

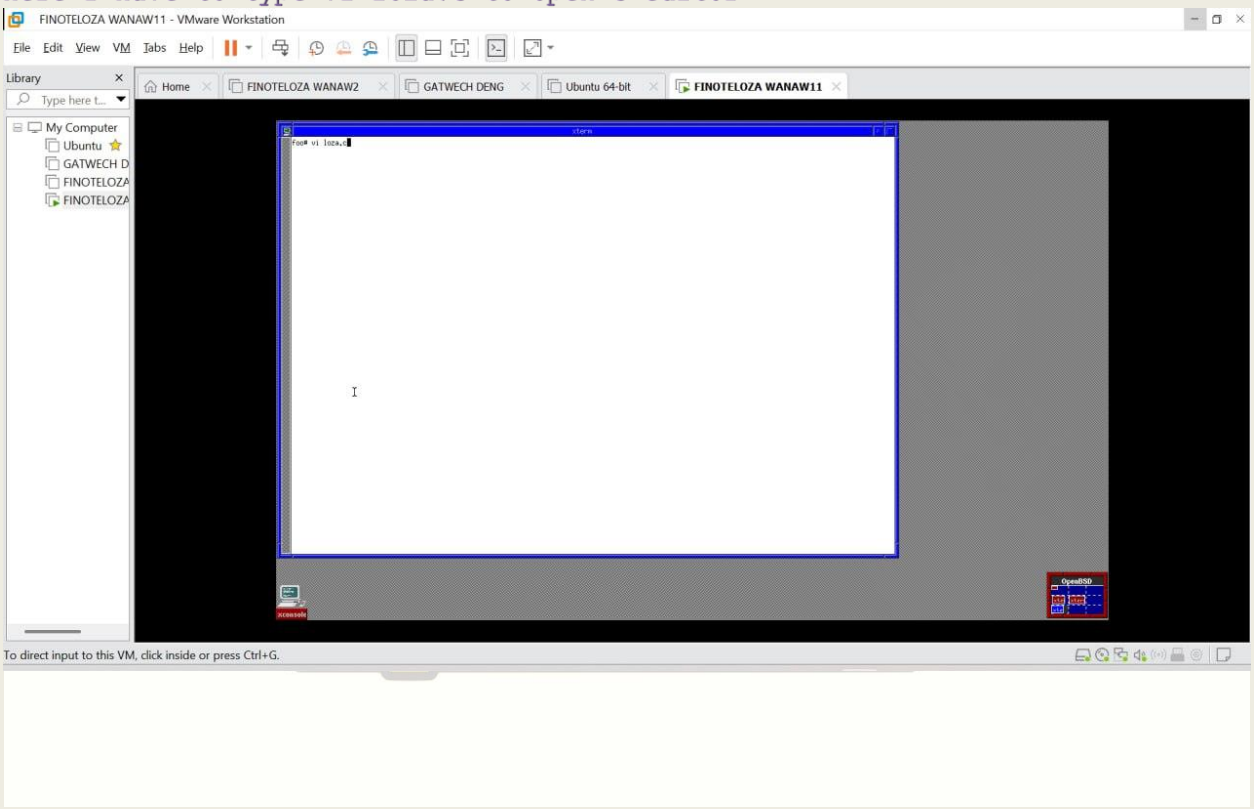
- **Type 1 hypervisors** (bare metal, like Hyper-V)

- **Type 2 hypervisors** (hosted, like VMware Workstation)

3. System Call Implementation

The `madvise()` system call provides the kernel with hints about memory usage patterns, helping optimize system performance.

here I have to type vi loza.c to open c-editor

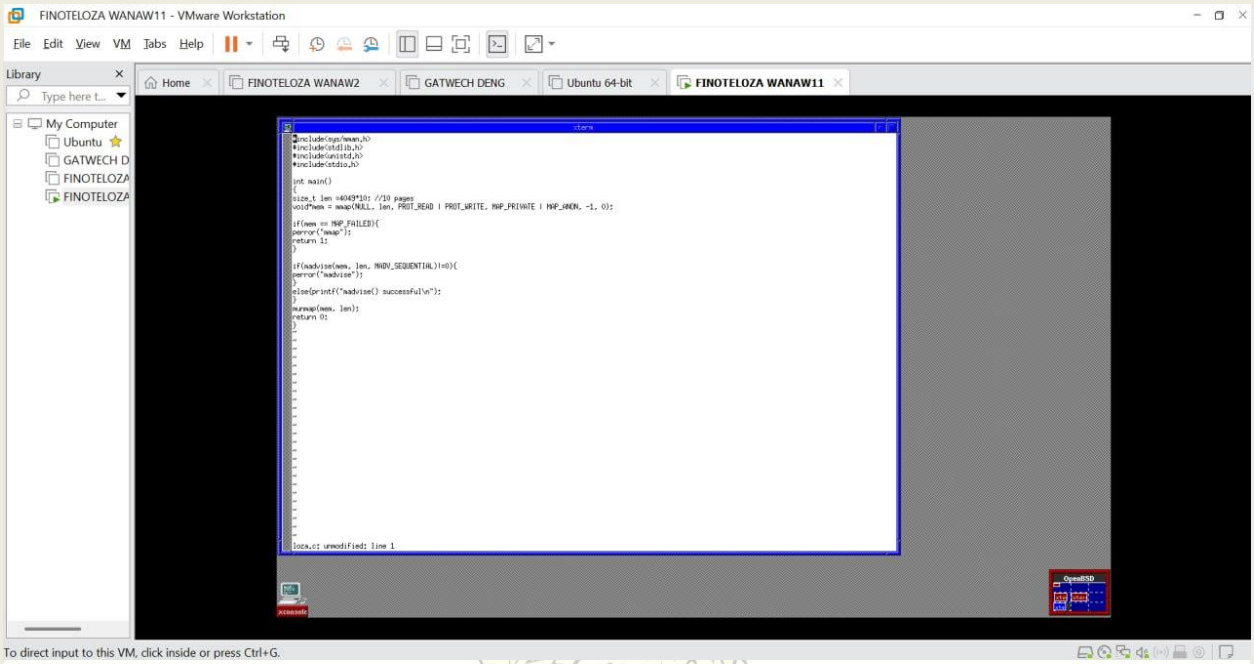


Parameters:

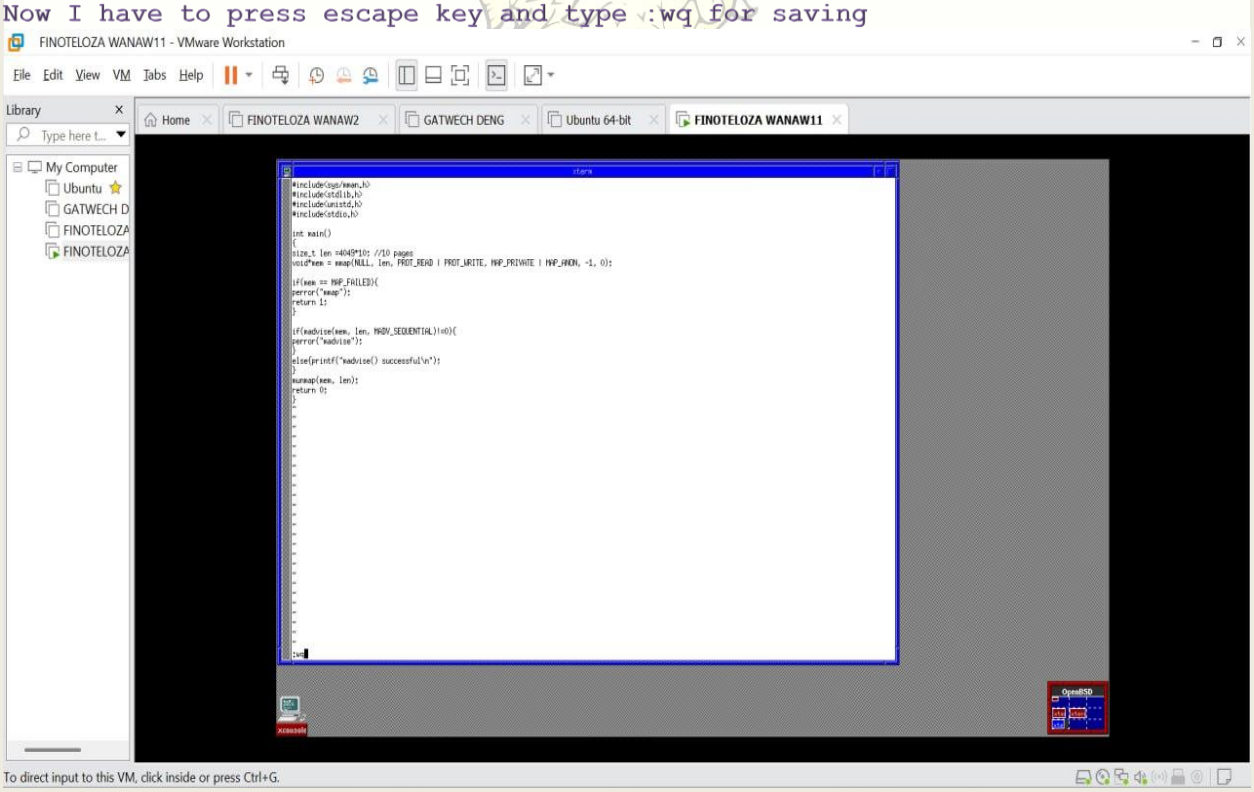
- `addr`: Starting address of the memory region
- `len`: Length in bytes
- `behav`: Expected access pattern (e.g., `MADV_SEQUENTIAL`, `MADV_RANDOM`)

Function Prototype:

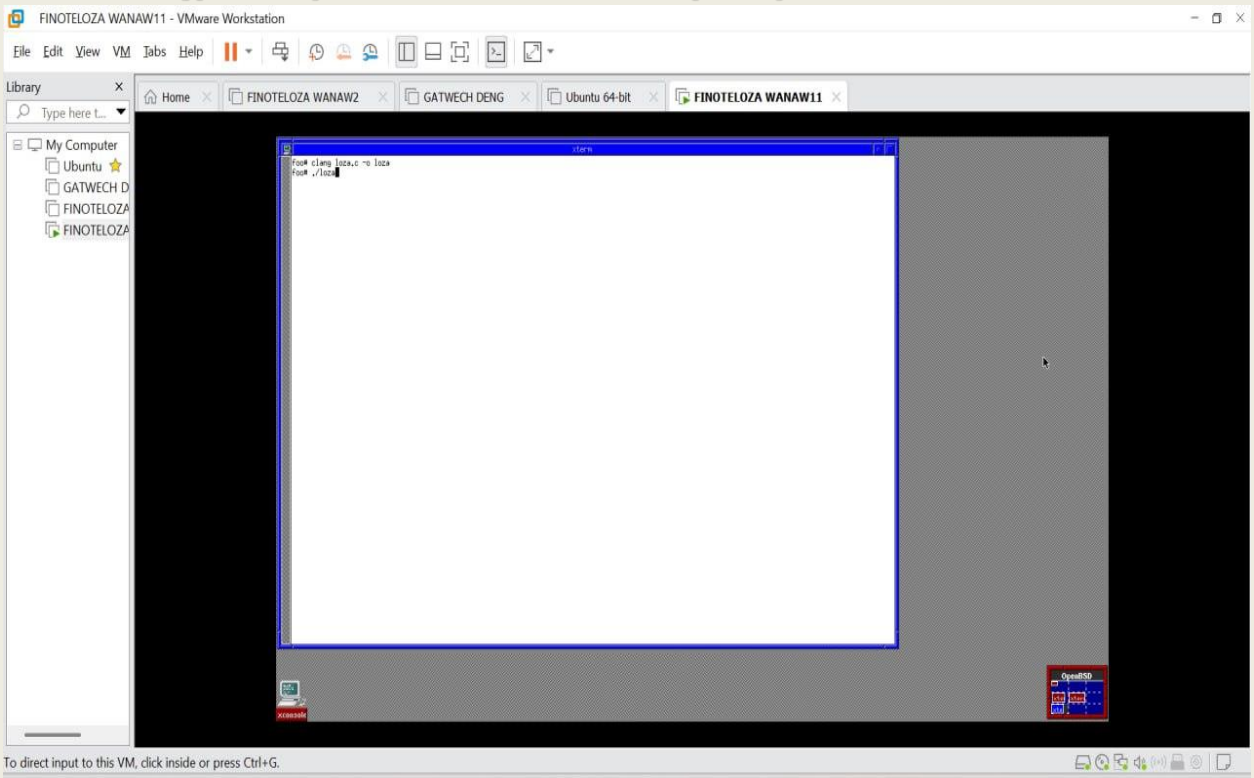
```
int madvise(void *addr, size_t len, int behav);
```



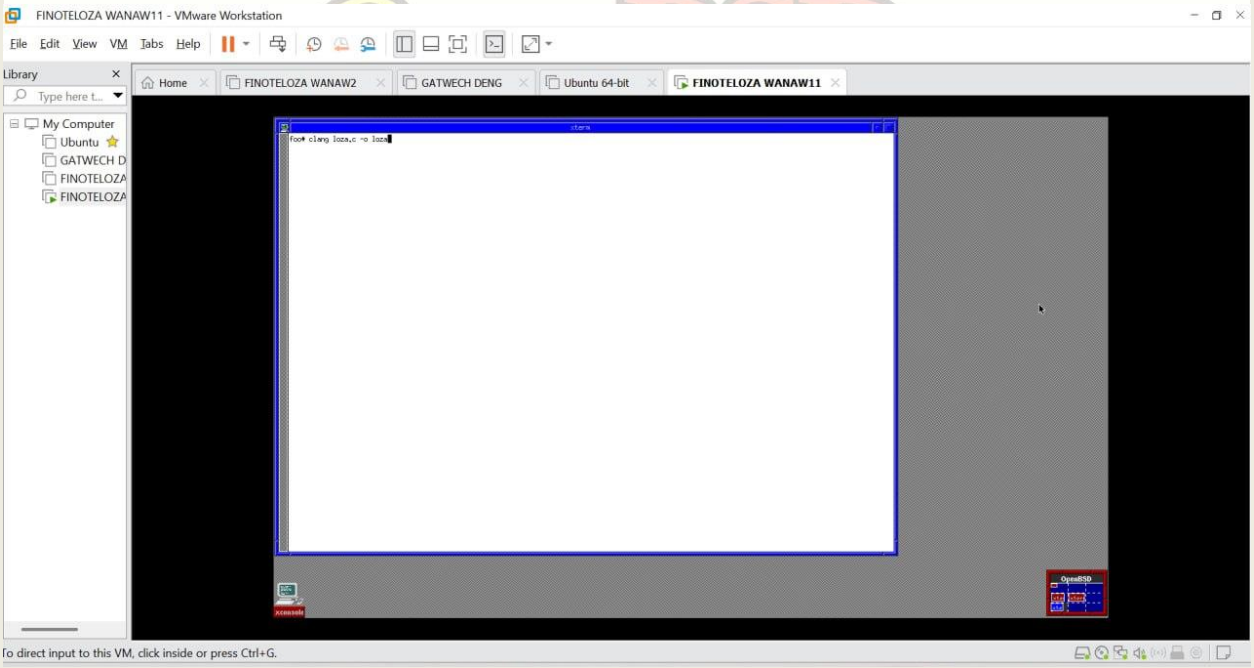
Now I have to press escape key and type :wq for saving



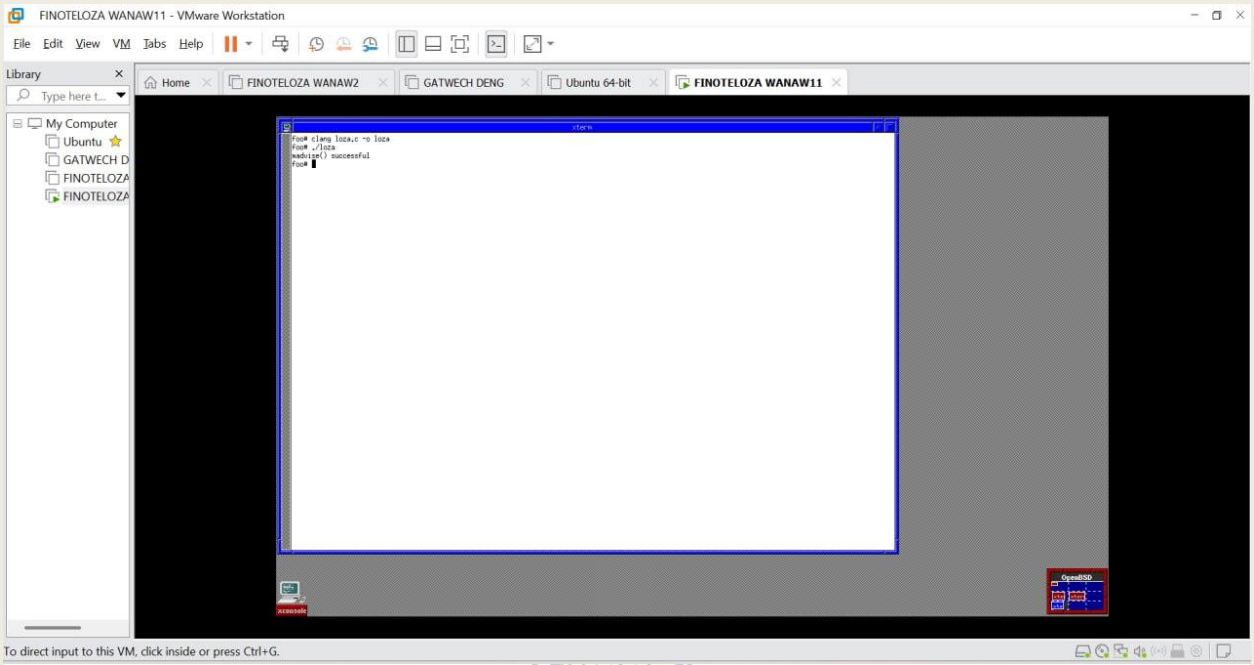
I have to type clang loza.c -o Loza for compiling



I have to type ./loza for running



Now it is successful (madvise() successful.



🔍 Internals:

OpenBSD validates memory regions, applies flags, and informs the VM system how to manage memory paging and prefetching more effectively.