BAHIR DAR UNIVERSITY

Software engineering

OPERATING SYSTEM

INDIVIDUAL ASSIGNMENT _ GUIX SYSTEM OS

Name	Id
Amanuel Gebrie	1600916
Section _ B	Submission Date_16/08/2017
	Submit to _ Lec . WendimuBaye

Table of content **Page**

A. Introduction (Background & motivation)

Background:

Guix System is an advanced, fully free and open-source **GNU/Linux distribution** that integrates the **GNU Guix** package manager at its core. It is known for its **declarative, functional approach** to both system configuration and package management, making it a unique option in the world of Linux distributions.

Developed by the **GNU Project**, Guix System emphasizes **reproducibility**, **flexibility**, and **freedom** for the user. Unlike traditional operating systems that typically offer imperative package management, Guix enables users to define their entire system configuration in a **declarative manner**, allowing for system builds that are fully reproducible across different machines.

Guix OS is a run-time GNU/Linux distribution with a focus on reproducible and declarative system configuration. It is based on the GNU Guix package manager and is appropriate for advanced users who value software freedom and total control over their system environment.

Motivation:

As computing platforms become increasingly dynamic and sophisticated, there's an increasing need for reproducible, adaptable, and trustworthy systems. Guix OS provides exactly this due to functional package management that aligns itself with modern Develops and infrastructure-as-code principles**.**

Guix provides a unique approach to system management with its declarative configuration and atomic upgrades, making it an excellent subject for studying modern operating system concepts**.**

B. Objectives

- **Install Guix OS in a virtual environment**
- **Understand its unique package management system**
- **Explore its file system and security features**
- **Compare with traditional Linux distributions**

Install and configure Guix OS in a virtual machine**.**

Configure user account "Amanuel Gebrie".

Record any installation problems encountered and how they were resolved.

Experiment with filesystem support in Guix OS.

Use a simple example with the clone() system call.

C. Requirements

i. Hardware:

CPU: x86_64 compatible processor

RAM: At least 2 GB (4 GB recommended)

Disk Space: At least 15 GB

Virtualization support enabled in BIOS

ii. Software:

Virtual Machine Software: Oracle VM VirtualBox (or VMware Workstation)

Guix OS ISO Image

Host OS (e.g., Windows/Linux for virtual environment)

In addition to this we should consider these requirements here below.

To successfully install and run **Guix OS** in a virtualized environment (such as **Virtual Box** or **VMware**), the following hardware and software requirements must be met:

1. Hardware Requirements

For Host Machine (Physical Computer Running the VM)

- **CPU**: 64-bit x86 processor (Intel/AMD) with **hardware virtualization support** (VT-x/AMD-V)
- **RAM**: Minimum **4GB** (8GB recommended for better performance)
- **Storage**: At least **20GB free disk space** (Guix itself is lightweight, but extra space is needed for software installations)
- **Internet Connection**: Required for downloading packages and updates

For Virtual Machine (Guix OS Installation)

- **Virtual CPU Cores**: At least **2 cores** allocated
- **RAM**: Minimum **2GB** (4GB recommended if running GUI applications)
- **Disk Space**: **20GB dynamically allocated** (ext4 or Btrfs recommended)
- **Graphics**: **VMSVGA** (for Virtual Box) or **VMware SVGA** (for VMware) for GUI support
- **Network**: **Bridged or NAT** (for internet access)

2. Software Requirements

For Host Machine

- **Hypervisor (Virtualization Software)**:
- **Oracle Virtual Box** (≥ 6.0)
- **VMware Workstation** (≥ 15.0) or **VMware Player** (Free)
- **QEMU/KVM** (Linux users)
- **Guix Installation Media**:
- **Download the latest Guix ISO** from https://guix.gnu.org/download/
- **USB/DVD writer** (if installing on real hardware)

For Guest VM (Guix OS)

- **Boot Mode**: **UEFI or Legacy BIOS** (Guix supports both, but UEFI is preferred)
- **File system**: **ext4** (default) or **Btrfs** (for snapshots)
- **Package Manager**: **Guix** (pre-installed)
- **Desktop Environment (Optional)**:
- **GNOME** (default)
- **Xfce** (lightweight alternative)
- **No GUI** (for server/CLI-only setups)

3. Additional Considerations

- **Secure Boot**: **Disabled** (Guix does not support Secure Boot by default).
- **Virtualization Extensions**: Must be **enabled in BIOS/UEFI** (VT-x/AMD-V).
- **Networking**: Ensure **NAT or Bridged** mode is configured for internet access during installation.
- **Guest Additions/VM Tools**: Not required (Guix uses open-source drivers).

Conclusion

❖ Guix OS has modest hardware requirements, making it suitable for virtualized environments. The key requirements are.

- A **64-bit CPU with virtualization support.**
- **4GB+ RAM** on the host machine.
- **20GB+ disk space.**
- A **compatible hypervisor** (Virtual Box, VMware, or QEMU).
- **Internet access** for package downloads.

Once these requirements are met, Guix can be installed and configured efficiently in a virtual machine for learning, development, or experimentation.

D. Installation Steps

1. first, Download Guix ISO: Get the installation image from https://guix.gnu.org/download/.

we get the guix OS on the offitial website https://guix.gnu.org/en/download.

2,Create a New Virtual Machine

To install Guix GNOME on a virtual machine, we create one with these parameters. We use AQEMU Virtual Machine.

•64-bit PC.

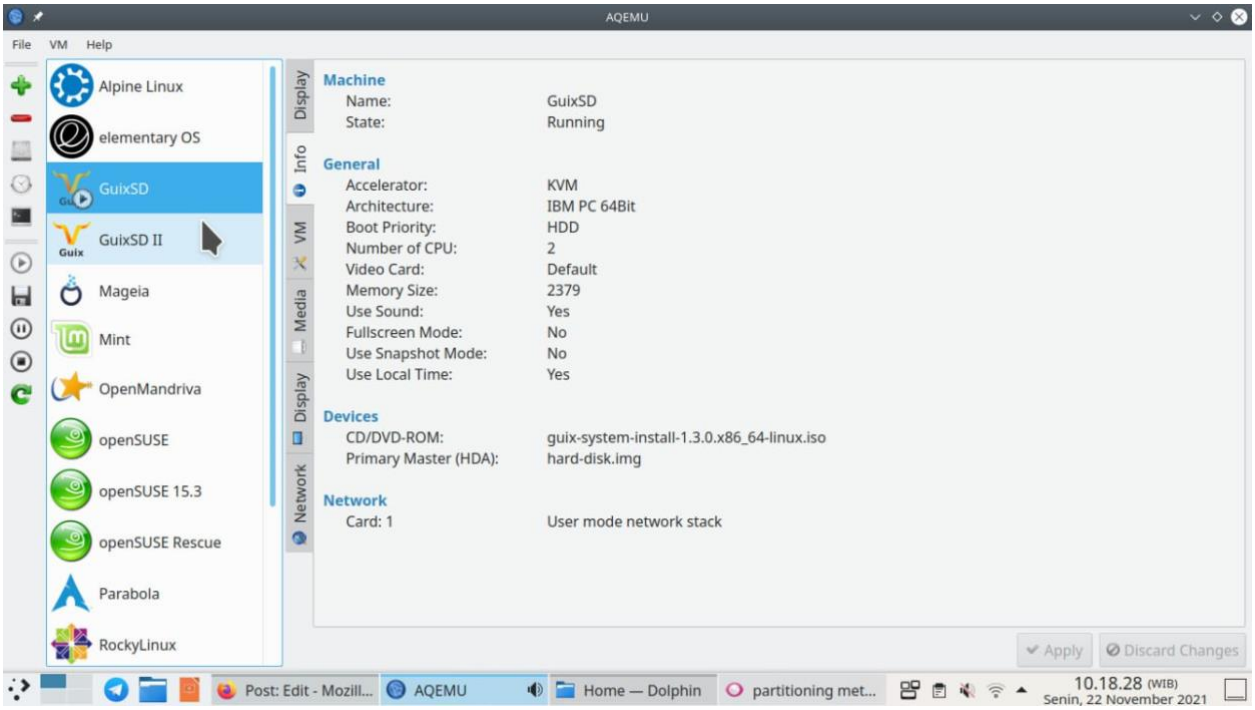
•2 GigaByte memory.

•20 GigaByte hard disk.

•Networking enabled.

•Default graphics option.

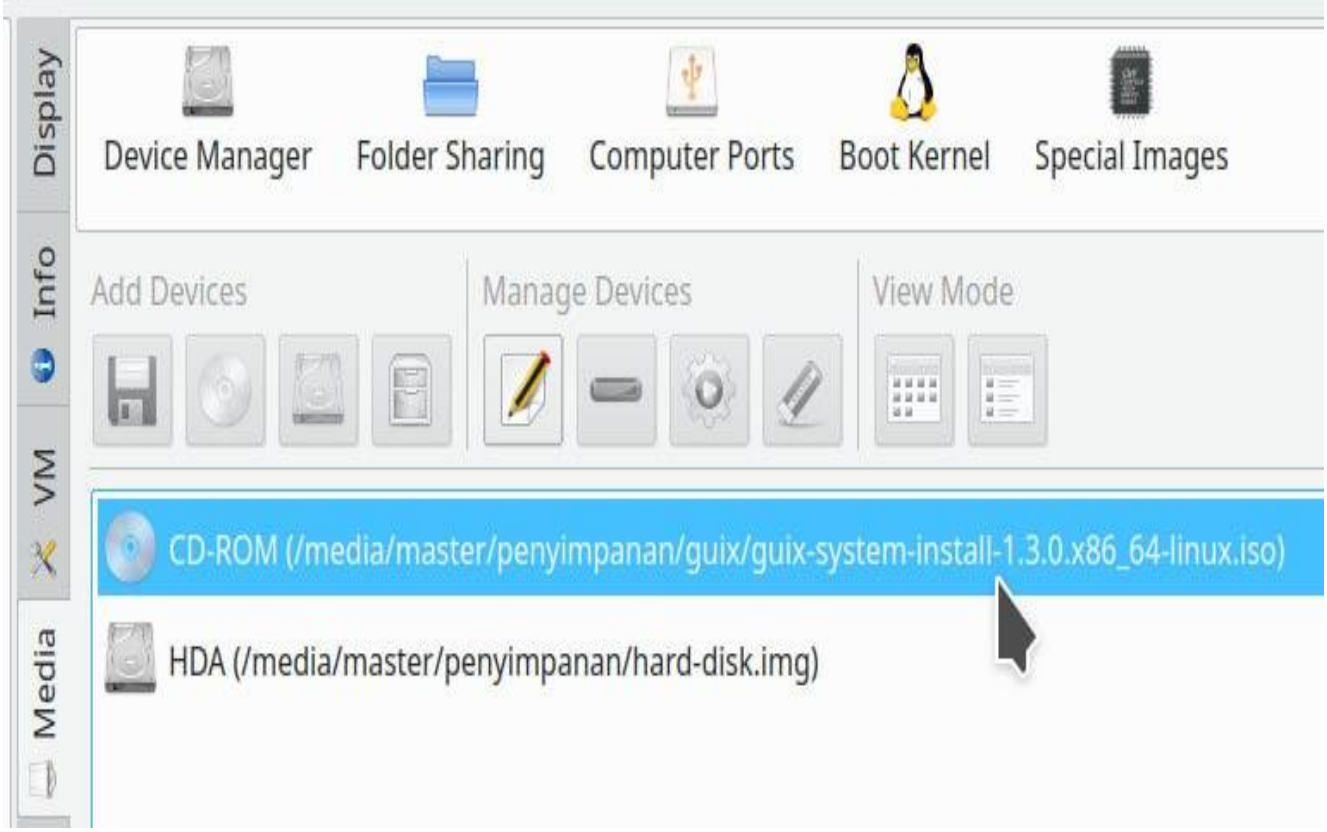
•BIOS Legacy booting mmode.



(AQEMU displaying the virtual machine configuration for Guix System)

3.Insert guix os file

Then, after we have created the virtual machine, next we insert the operating system file into it. To do so, on unstartedGuixvm, select Media, click Add CD/DVD ROM, insert the guix-system-*.iso file, image file inserted. Proceed to next section.



(ISO image preparation on the virtual machine)

4, Boot up guixOS: Once we put the image file in place,

We then boot the virtual computer to install Guix System installer. To do this, we double click on the Guixvm, click menubar VM, click Start, a display port will be opened as a monitor, Guix System installer will be shown, you are now ready to install Guix System. Proceed to next.

(Guix System when booting into its installer)

How To Play

Guix installer does not allow mouse click interactions. Thus, to navigate the system installer, press TAB and Shift+TAB to jump from one choice to another one and back, press SPACE to give check mark, and press ENTER to select an option.

5, user identity

- We Select language
- We Select territory
- We Select installation type
- We Select a timezone
- We Select a keyboard layout
- Name your computer

1 . Locale language: We choose English and continue.

2. Locale location: select United States and continue.

3. GNU Guix install: select Graphical install. option and continue.

4. Time zone: select a time zone e.g. Asia/Jakarta.

5. Keyboard layout: select English (US).

6. Hostname: select a name for your Guix computer e.g. we use 'master'. Continue to next step.

6. System set up

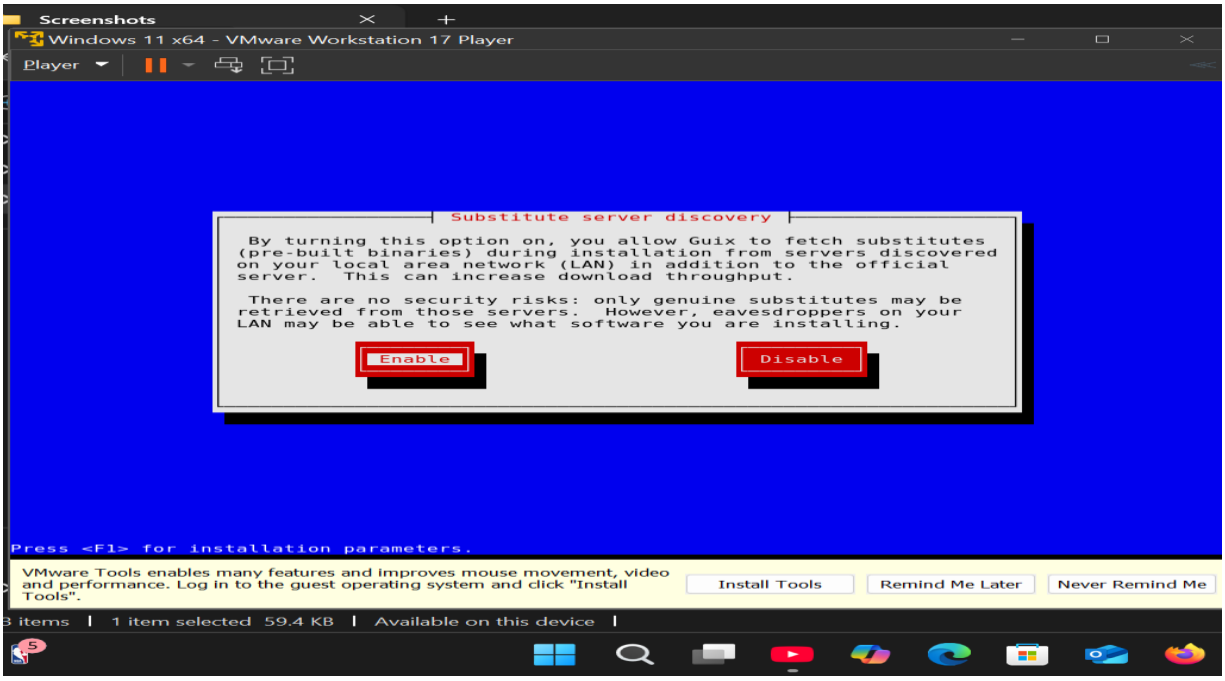
•Substitute server discovery

•Create administrator account

•Create user account

• Select a desktop environment

•\t Enable network



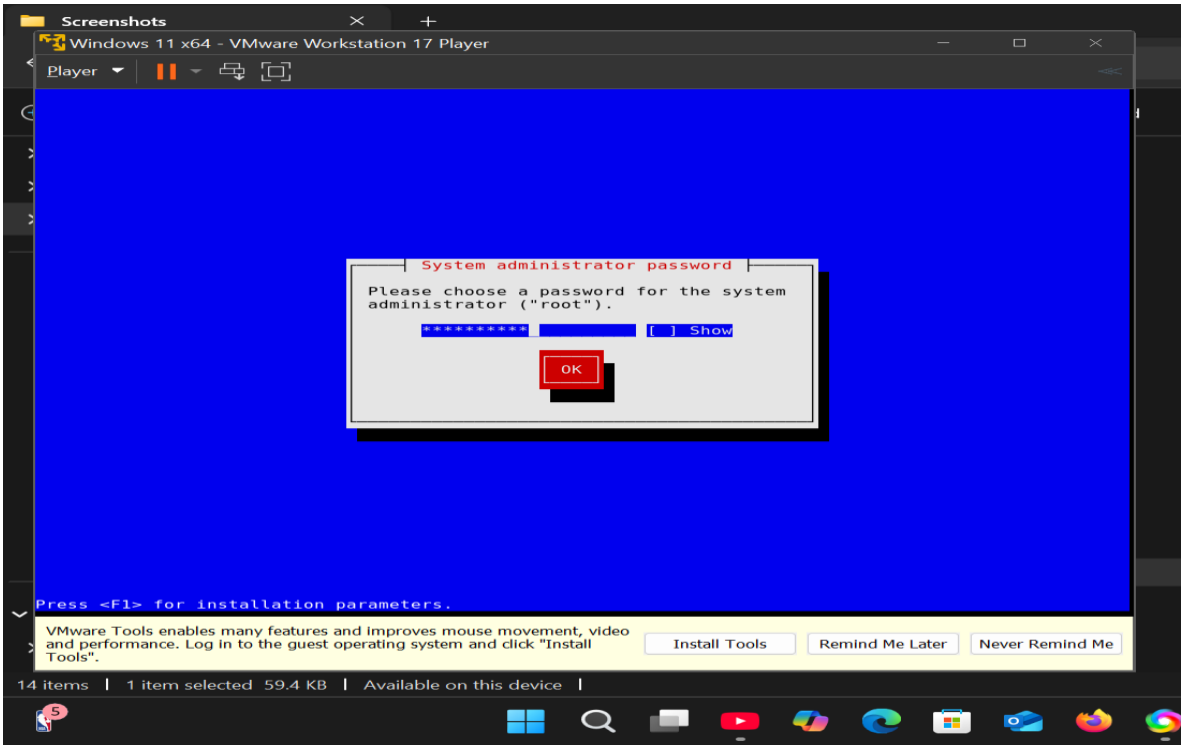
1. Substitute server discovery: select Enable and continue

2. System administrator password: create a root user password.

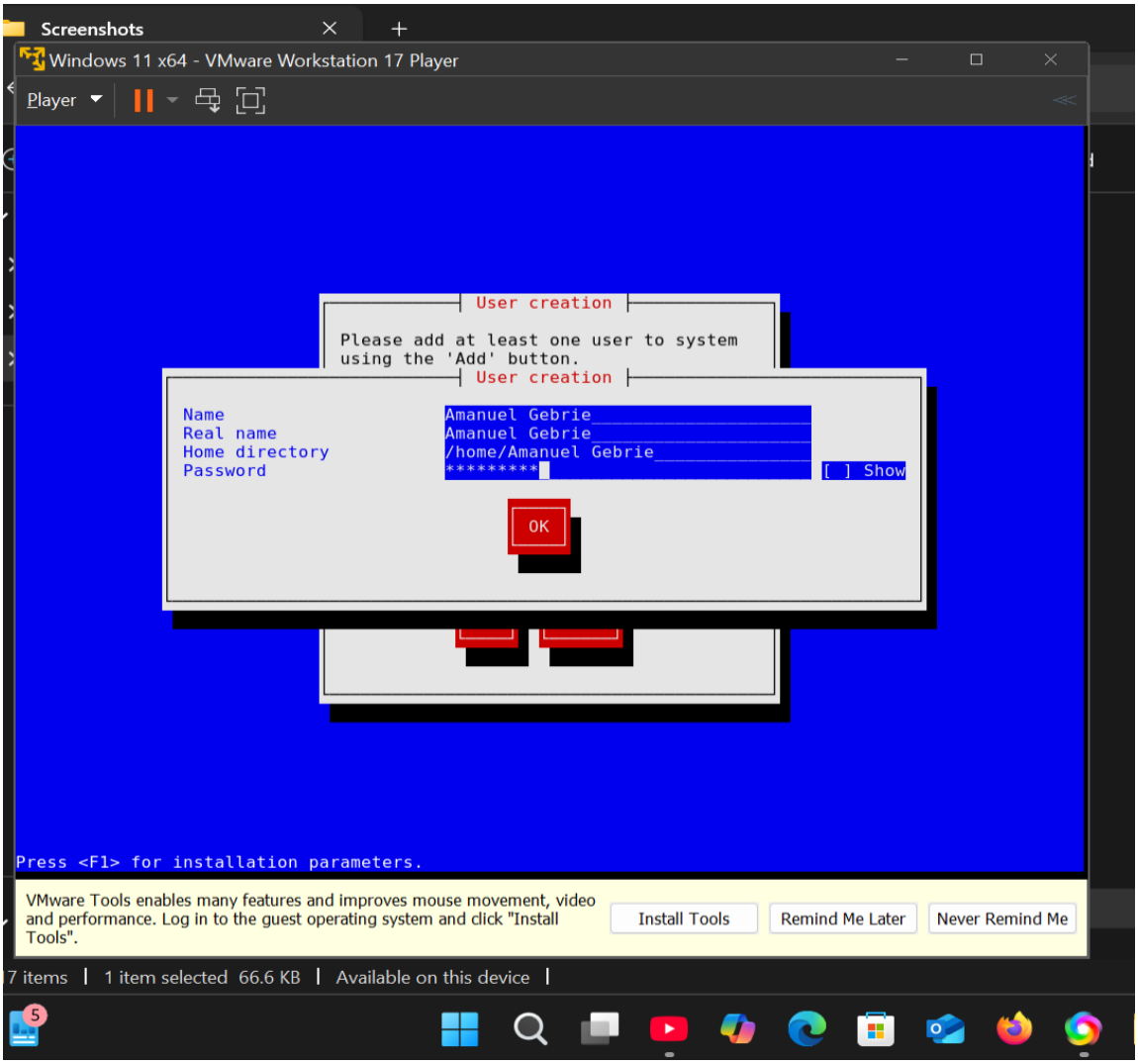
3. Building a user: assign yourself a username and a password.

4. Desktop environment: check box beside GNOME from the list.

5. Network service: select Mozilla NSS Certification and go to next step.

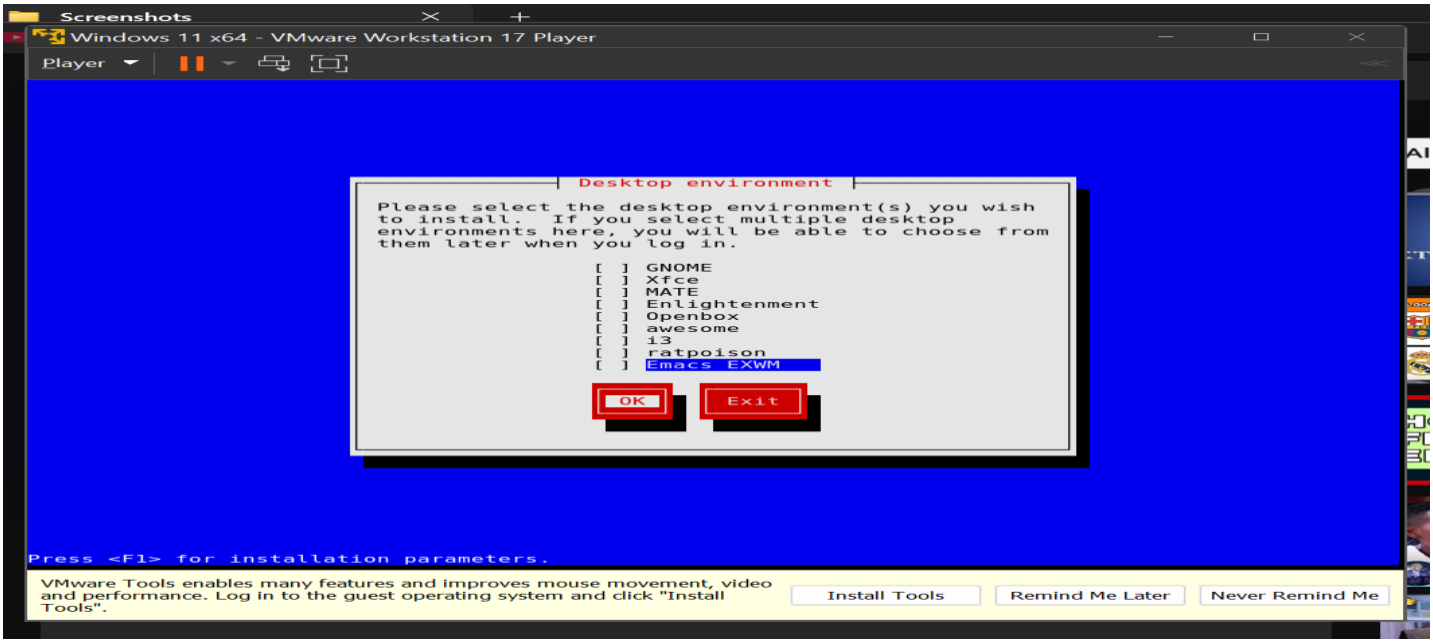


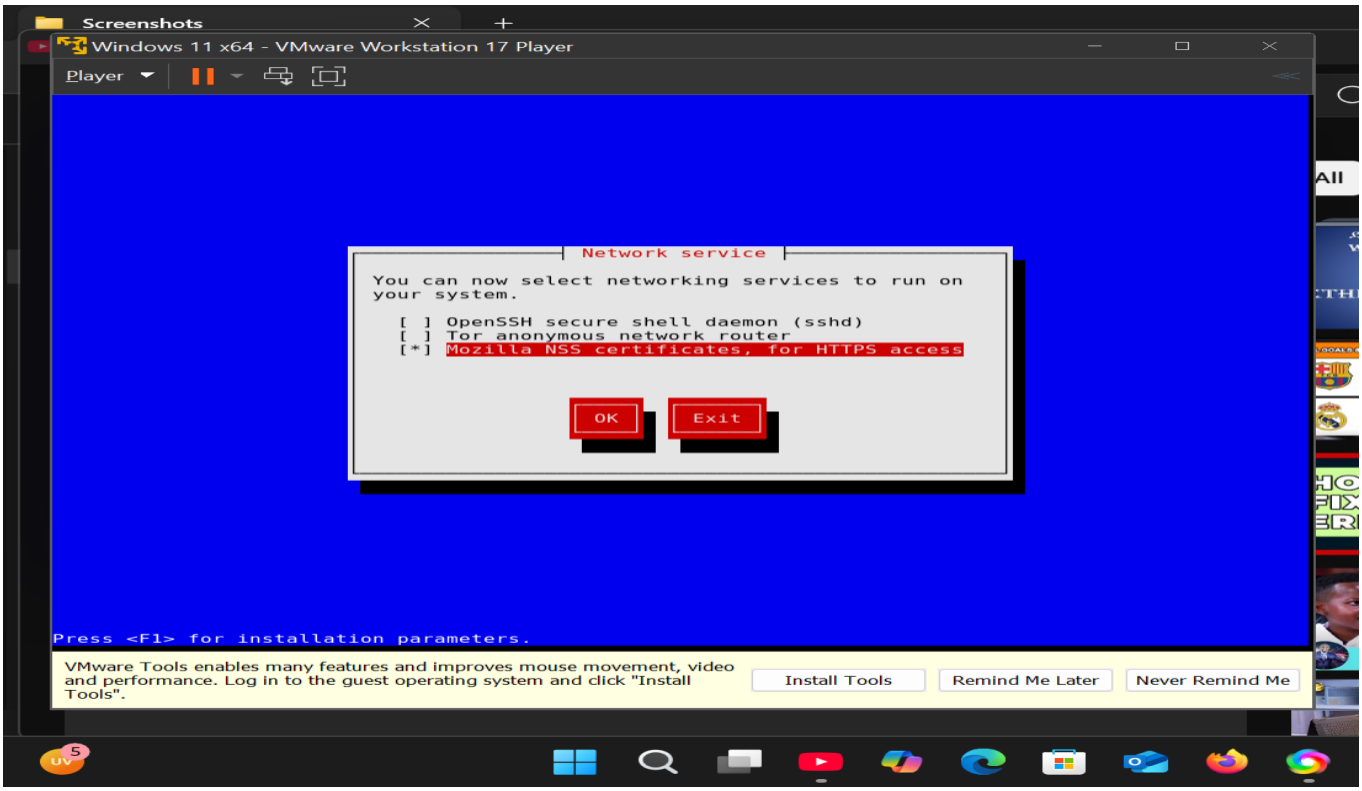
(System administrator password)



(User creation)

This is desktop environment





(This is the Network service)

7.partitioning,processing and finishing

•Stare Guided partitioning

•Select disk

•Select partitioning scheme

•Make partitions required

• Verify the summary in code form

•Start the real installation

•Wait patiently for the process to finish

1, Partitioning option: select Guided - using entire disk.

2. Disk: select the virtual machine's hard disk. In this scenario, we select ATA QEMU HARD DISK (20GB).

3 .attrition Partition scheme: select Everything is one partition.

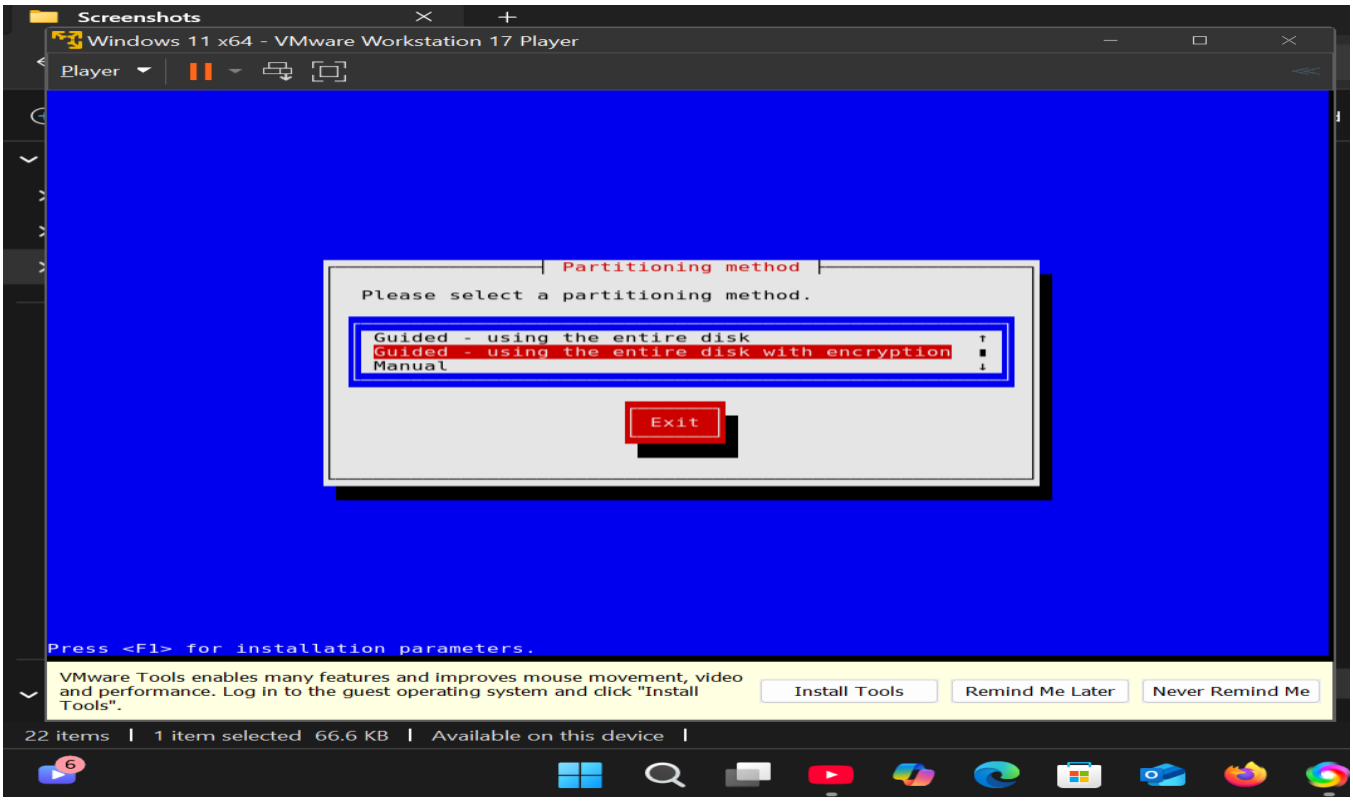
4. Guided partitioning: the installer will display a proposal of disk partitions that will be formatted. Accept it by selecting OK. Continue.

5. Format disk?: accept this by selecting OK.

6. Preparing partitions: please wait for the disk formatting process is taking place.

7. Configuration file: the installation plan will be displayed in form of code written in a language called Scheme. Accept this by selecting OK.

8. Waiting: the installation process will take place. If this process takes place, don't close your internet connection and don't shut down the virtual machine. Wait for a few minutes until you are faced with a final message saying "Please ENTER to continue".



Partitioning method

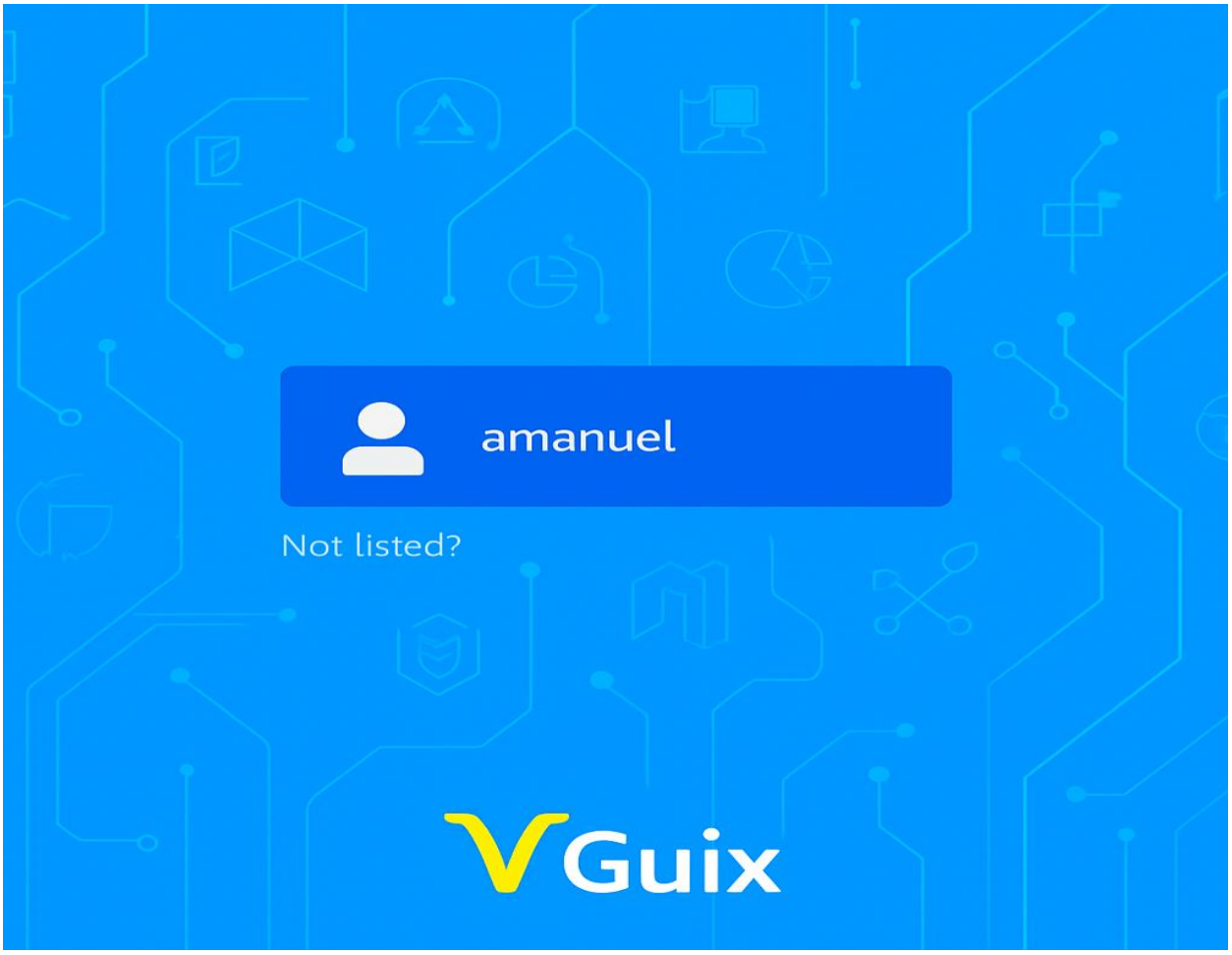
Guided partitioning

(here is Configuration file)

Here is the actual installation process

8. Restart and login

Once finished, the installer will ask you to Reboot then press Enter and you should be able to see Guix login screen like below.



E. Problems Faced

i. Default network adapter not working.

GRUB boot loader unable to detect boot partition.

F. Solution

i. Changed network settings to "Bridged Adapter" in Virtual Box.

Set GRUB installation location manually in final steps of installation.

G. File system Support

Guix primarily uses ext4 or btrfs for its root file system.

Guix OS has native support for ext4, Btrfs, and XFS. The default and recommended is ext4 because it is stable, widely supported, and supports journaling data integrity. Other file systems like Btrfs are feature-rich but less stable with minimal setups.

The choice is typically ext4 for its stability and journaling capabilities, which align with Guix's transactional approach to system management.

H. Advantages and Disadvantages

Advantages:

Extremely high customizability through functional configuration.

High free software emphasis.

Reproducible builds with Guix package manager.

Atomic upgrades and rollbacks.

Reproducible system configurations.

Declarative system management.

Unprivileged package management.

Disadvantages:

Mildly steeper learning curve.

Less hardware support compared to mainstream distributions.

Smaller support resources and community.

H. Conclusion

Guix offers a revolutionary approach to system management that emphasizes reproducibility and user control. While it may not be suitable for all users, it provides valuable insights into alternative operating system paradigms.

Installing and utilizing Guix OS provides more insight into modern-day operating systems and usable package management. Even though it is challenging, learning from it is worthwhile for the sake of appreciating reproducibility and declarative configuration in Linux.

J. Future Outlook / Recommendation

As functional package management gains popularity, Guix is well-positioned to influence mainstream distributions. Recommendations include improving hardware support and expanding the package repository

Guix OS is most appropriate for advanced users and researchers. It's appropriate for reproducible science, DevOps, or system administration experts. Its functional style is a vision of the future of system configuration and software management.

Virtualization in Modern Operating Systems

What is Virtualization?

Virtualization is a technology allowing you to create several simulated environments or committed resources from one physical hardware

configuration. This includes virtual machines (VMs), virtual storage, virtual networks, etc.

For operating systems, virtualization means running several copies of an OS—such as Linux, Windows, or macOS—on one physical machine simultaneously.

Every virtual machine runs independently with its own virtual CPU, memory, storage, and network interface, though they all share the physical resources of the host machine.

Why Use Virtualization?

Virtualization has become a necessity in today's computing for several critical reasons:

Resource Efficiency: Allows more effective use of hardware by running multiple OSes on a single machine.

Isolation: Every virtual machine is isolated from each other, improving security and stability.

Testing and Development: Applications can be tested on many different operating systems without needing several physical machines.

Disaster Recovery: VMs are easily recoverable and backed up in seconds.

Cost Savings: Less hardware and energy cost when there are fewer physical machines.

Scalability and Flexibility: Virtual machines are easy to deploy and manage in cloud and enterprise setups.

How Does Virtualization Work?

Virtualization is facilitated by a layer of software called a hypervisor. It sits between the hardware and the operating system(s) and manages the hardware resource allocation.

There are two main types of hypervisors:

Type 1 (Bare Metal):

Runs on bare hardware. Examples: VMware ESXi, Microsoft Hyper-V, Xen.

Type 2 (Hosted):

Runs on top of a host OS. Examples: Oracle VM VirtualBox, VMware Workstation.

Basic Workflow:

Hypervisor Installation:

The hypervisor is loaded onto a host computer (either on bare metal or on top of an OS).

Resource Allocation:

You define how much CPU, RAM, storage, etc., each virtual machine gets.

VM Creation:

You define virtual machines and install guest operating systems inside them.

Execution:

The hypervisor manages all interaction between the VM and the underlying hardware.

Example: Using Virtual Box to Run Guix OS

Host OS: Windows/Linux/macOS

Hypervisor: Virtual Box

Guest OS: Guix OS

Outcome: Guix runs as if it were installed on a real computer, although it is being run in a controlled virtual environment.