
# BAHIR DAR UNIVERSITY

**Faculty of computing**

**Software Engineering Department**

**Operating System and System Programing Individual Assignment**

**Name:** ***Nibretu Mengaw***

**ID:** ***BDU1602278***

**Section:** ***B***

*Submitted to: Lec. Wondmu B.*

*Submission date: 16/08/2017E.C*

*Bahirdar Ethiopia*

# Contents

# Documentation to install Manjaro Linux OS

## 1. Installation of Operating System in Virtual Environment

### a. Introduction (Background, Motivation)

**Background**:- Operating systems (OS) manage computer hardware and software resources, providing a platform for applications. Virtualization allow running multiple OS instances concurrently on a single physical machine using software like VirtualBox. Manjaro Linux is a user-friendly, Arch Linux-based distribution known for its accessibility, customizability, and access to the latest software via its rolling-release model and the Arch User Repository (AUR). This task involves installing Manjaro Linux within a virtual environment.

**Motivation:-** Installing Manjaro in a virtual machine (VM) provides a safe, isolated environment to explore its features, test software compatibility, and learn about Linux/Arch-based systems without affecting the host operating system. This practical exercise is crucial for understanding OS installation procedures and virtualization concepts within the OSSP course.

### b. Objectives

- ✓ Successfully install Manjaro Linux (XFCE Edition recommended for VMs) within VirtualBox
- ✓ Clearly document the installation process using the Calamares installer, including guiding image.
- ✓ Identify and document any potential issues encountered during setup or installation.
- ✓ Analyze and report the filesystem support, particularly the default choice (ext4).
- ✓ Understand the specific requirements, advantages, and disadvantages of Manjaro Linux.
- ✓ Gain hands-on experience with installing a rolling-release Arch-based Linux distribution.

### c. Requirements

#### i. Hardware (Host Machine)

- Processor: Minimum: 1 G
  Recommended: 2+ GHz, 2+ Cores.
- RAM: Minimum: 2 GB
  Recommended: 4 GB+ for VM.
- Storage: VM disk size: 20 GB
  Recommended: 30 GB+.
- Other: Hardware virtualization support (Intel VT-x or AMD-V) enabled in BIOS/UEFI.

#### ii. Software (Host Machine)

- Host OS: i.e, Windows 11 Pro , Window 10 etc.
- Virtualization Software: Select the latest version like Oracle VM VirtualBox 7.0.10.
- Installation Media: Manjaro Linux ISO file (e.g. manjaro-xfce-23.0.0-230824-linux65.iso). Download from the official Manjaro website: https://manjaro.org/products

  Select the XFCE edition recommended for general use/VMs.

## d. Installation Steps

### Step 1: Setting up the Virtual Machine in VirtualBox

- ✓ Open VirtualBox and click 'New'.
- ✓ Name: Manjaro Linux
- ✓ Type: Linux
- ✓ Version: Arch Linux (64-bit)
- ✓ Memory size: Allocate RAM (e.g., `4096` MB or more if available).
- ✓ Hard disk: Select 'Create a virtual hard disk now'. Choose VDI (VirtualBox Disk Image), 'Dynamically allocated', and set the size (e.g., `30` GB or more). Click 'Create'.

### Step 2: Configuring VM Settings

- ✓ Select the created 'Manjaro Linux' VM and click 'Settings'.
- ✓ System -> Processor: Allocate at least `2` CPU cores if possible.
- ✓ Display -> Screen: Increase Video Memory to `128` MB. Check 'Enable 3D Acceleration' (optional, may improve desktop performance).
- ✓ Storage: Select the 'Empty' optical drive under 'IDE Controller' or 'SATA Controller'. Click the disc icon on the right and 'Choose a disk file...'. Browse to and select your downloaded Manjaro ISO file.
- ✓ Network: Ensure 'Adapter 1' is 'Enabled' and 'Attached to': 'NAT' (default, usually works fine for internet access).
- ✓ Click 'OK'.

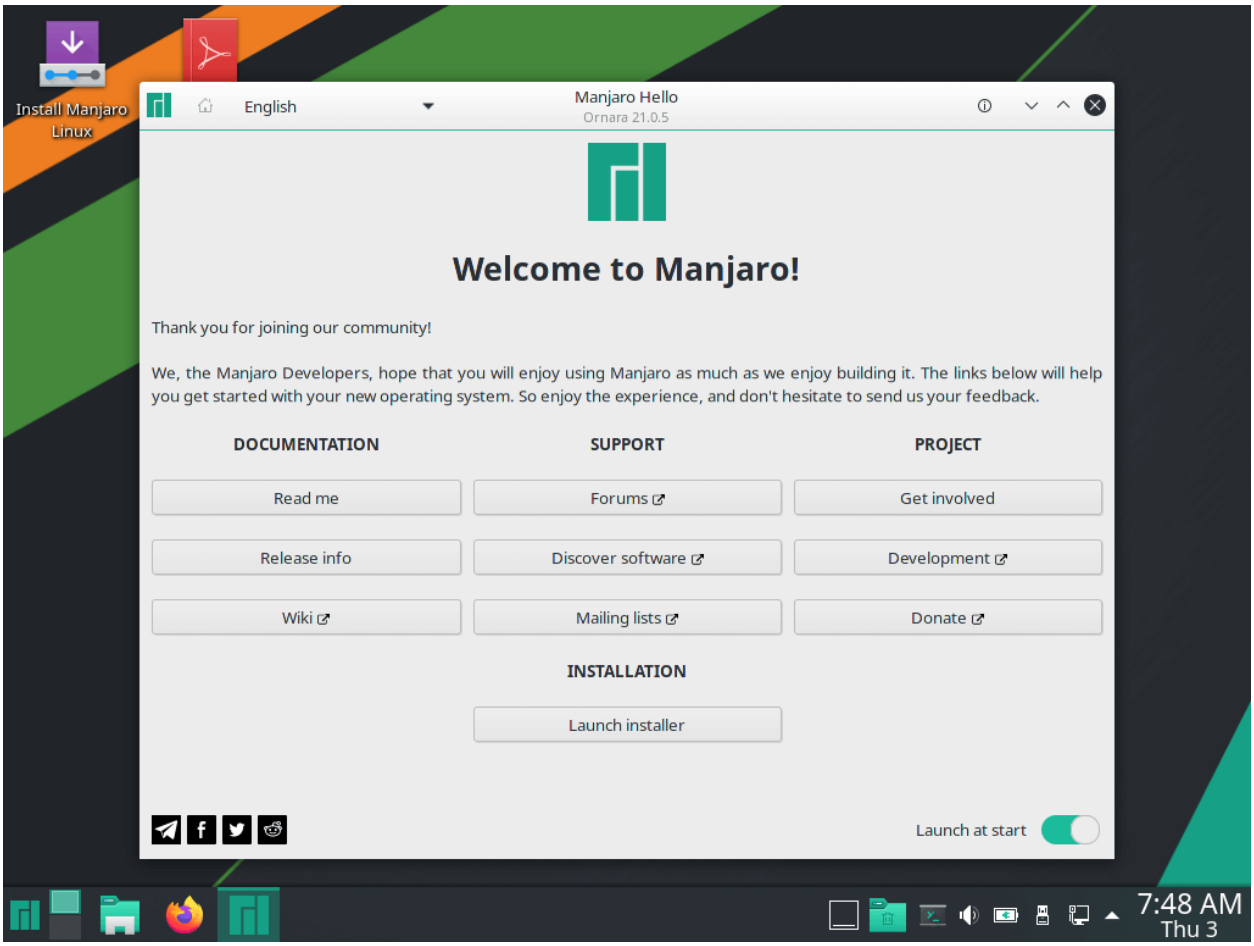
### Step 3: Starting the Installation

- ✓ Select the 'Manjaro Linux' VM and click 'Start'.
- ✓ The VM should boot from the ISO. Use arrow keys to select 'Boot with open source drivers' and press Enter.


✓

✓ Wait for the live environment to load. You will see the Manjaro XFCE desktop.

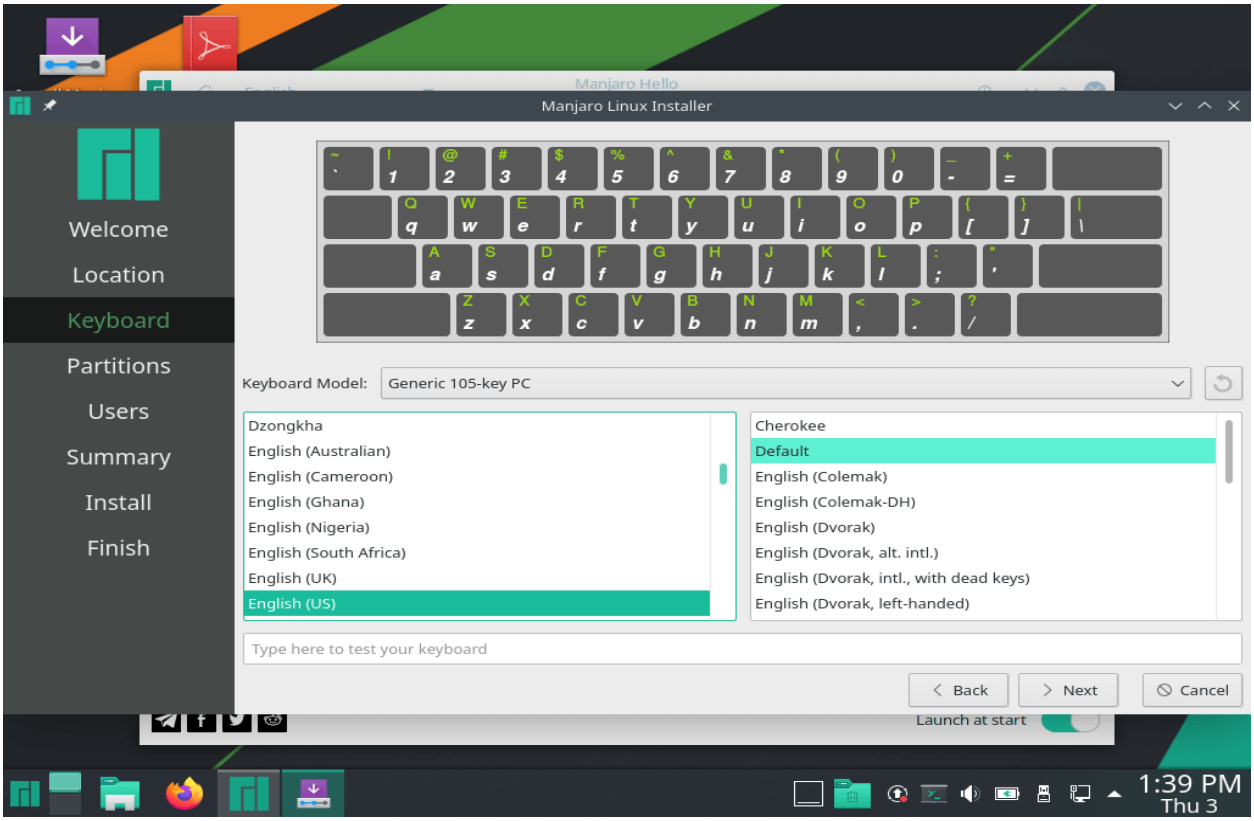
✓ Double-click the 'Launch installer' icon on the desktop.



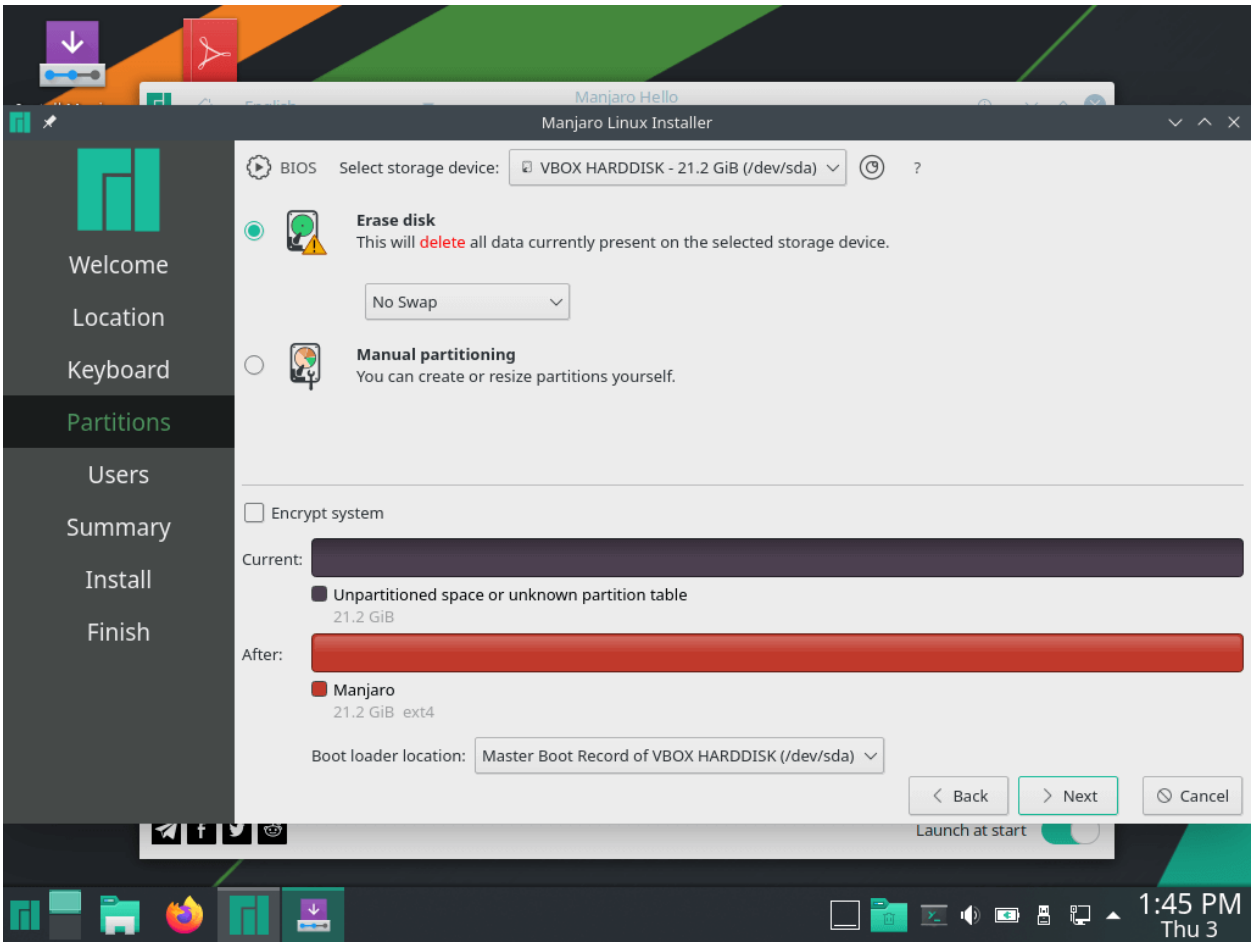
**Step 4: Following the Manjaro Ornara Installer**

- ✓ Welcome:- Select language and click 'Next'.


- ✓ Location:- Select your region and zone (e.g., Africa/Addis_Ababa) to set the timezone. Click 'Next'.
- ✓ Keyboard:- Select keyboard layout (e.g., `English (US)`) and click 'Next'.



- ✓ Partitions:- Choose 'Erase disk' . This is the simplest option for a VM. It will automatically partition the virtual disk using the default **ext4** filesystem and create a swap partition or swap file depending on Manjaro version.

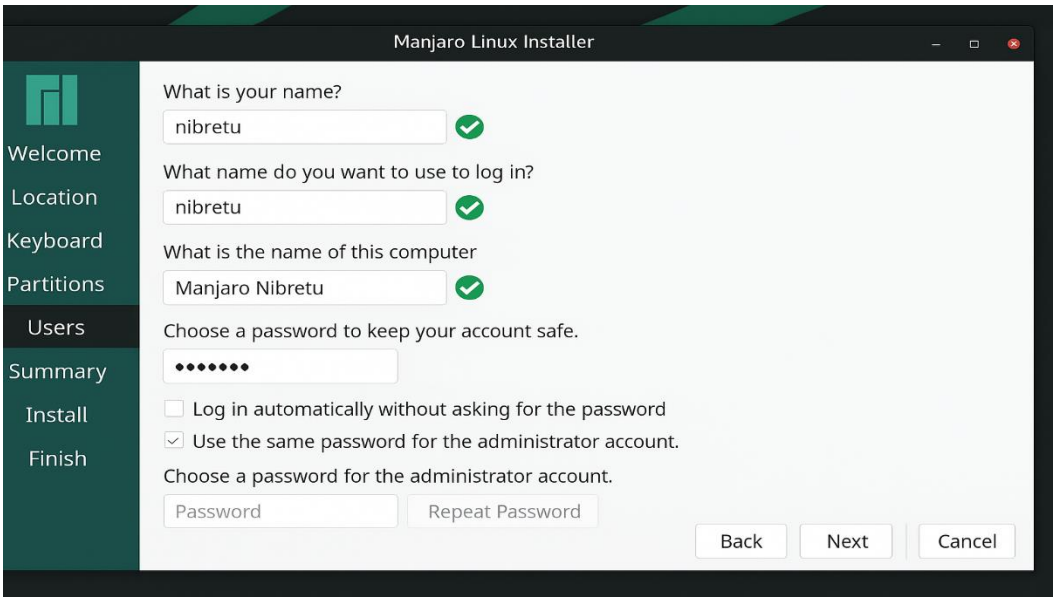



- ✓ Ensure the correct virtual disk is selected if you have multiple attached.
- ✓ Select Swap preference if offered (e.g., 'Swap (with Hibernate)' or 'Swap (no Hibernate)' or 'No Swap'). 'With Hibernate' is often default.

- ✓ Click 'Next'.

### Step 5: User Account Creation

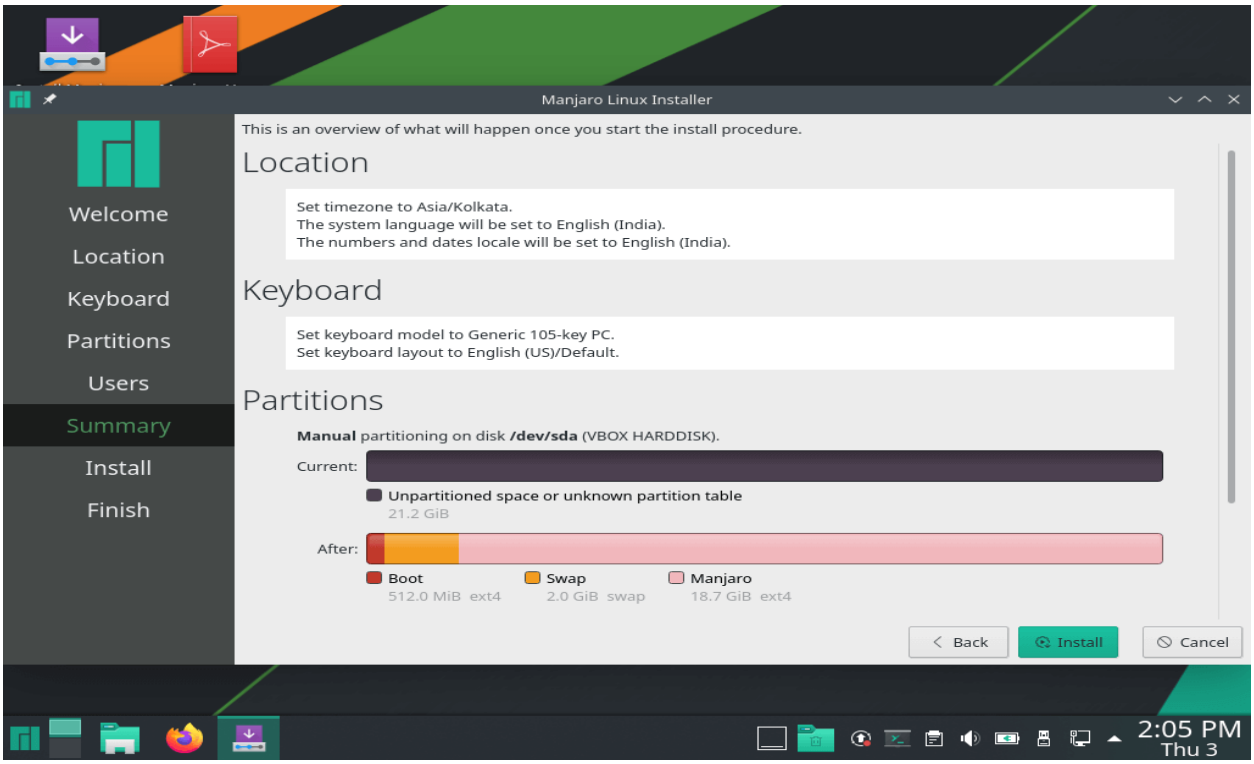
- ✓ Enter your name:
- ✓ Enter the username you want to use to log in.
- ✓ Enter the name for the computer (hostname): e.g., manjaro-vm
- ✓ Choose and confirm a password for your user account.
- ✓ Optionally check 'Use the same password for the administrator account'. Like this

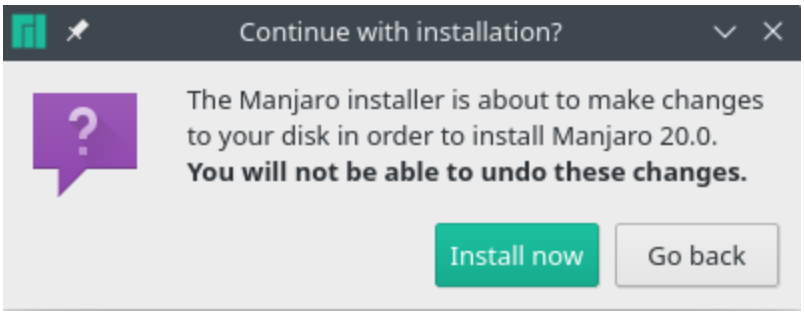



✓ Click 'Next'.

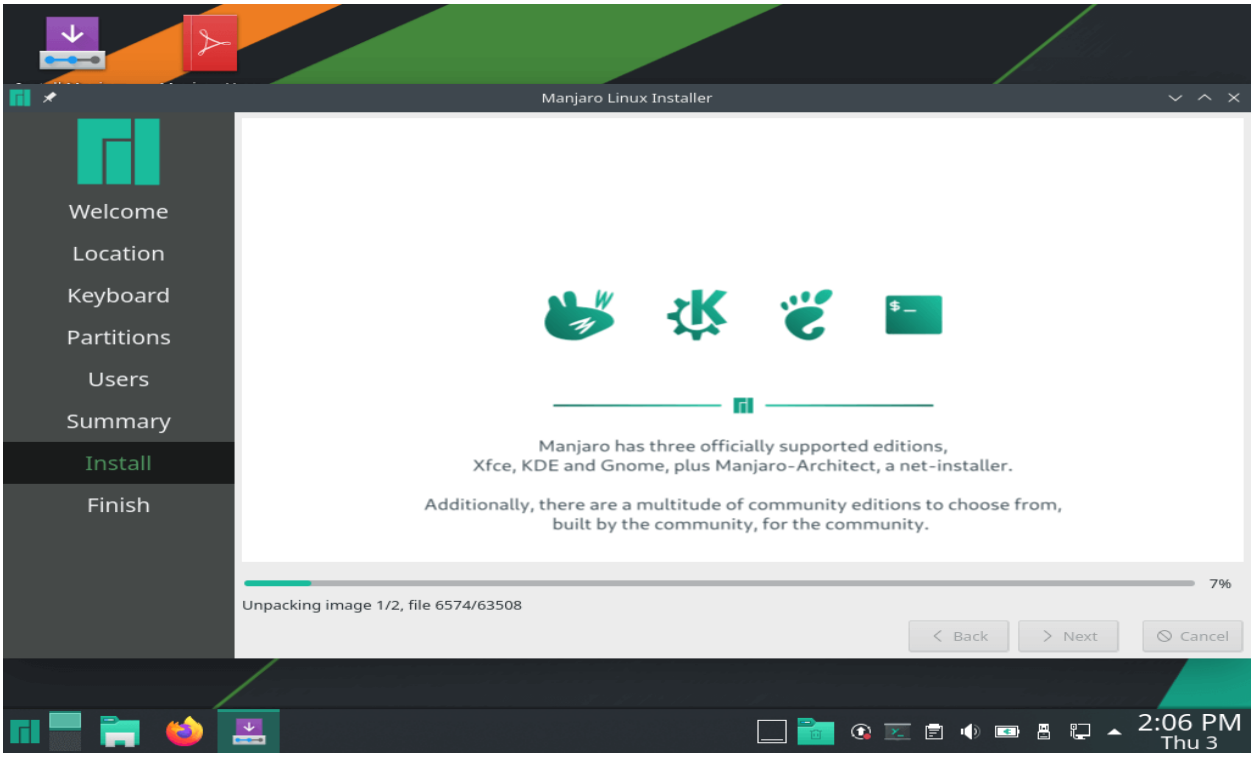
## Step 6: Summary and Installation

✓ Review the summary of installation settings. If correct, click 'Install'. Confirm the warning about erasing the disk by clicking 'Install' and then “install now”.

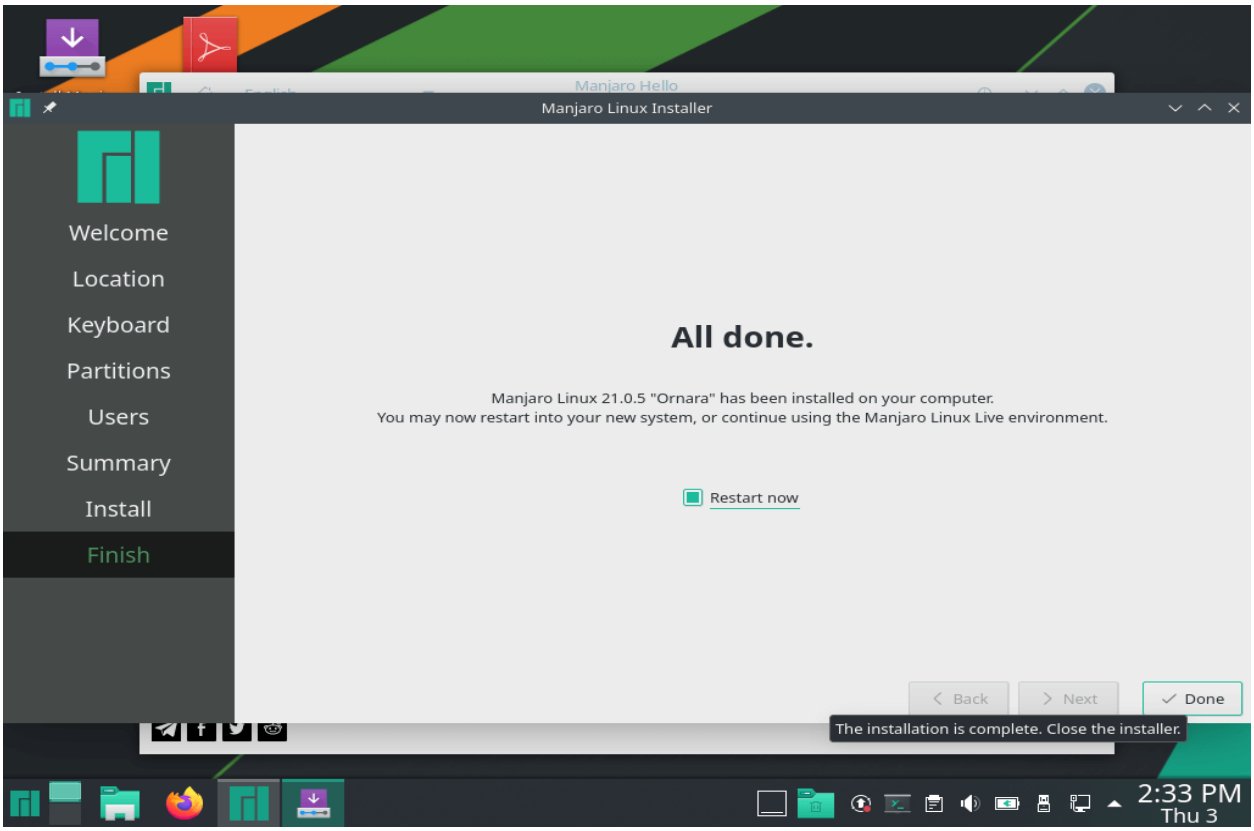




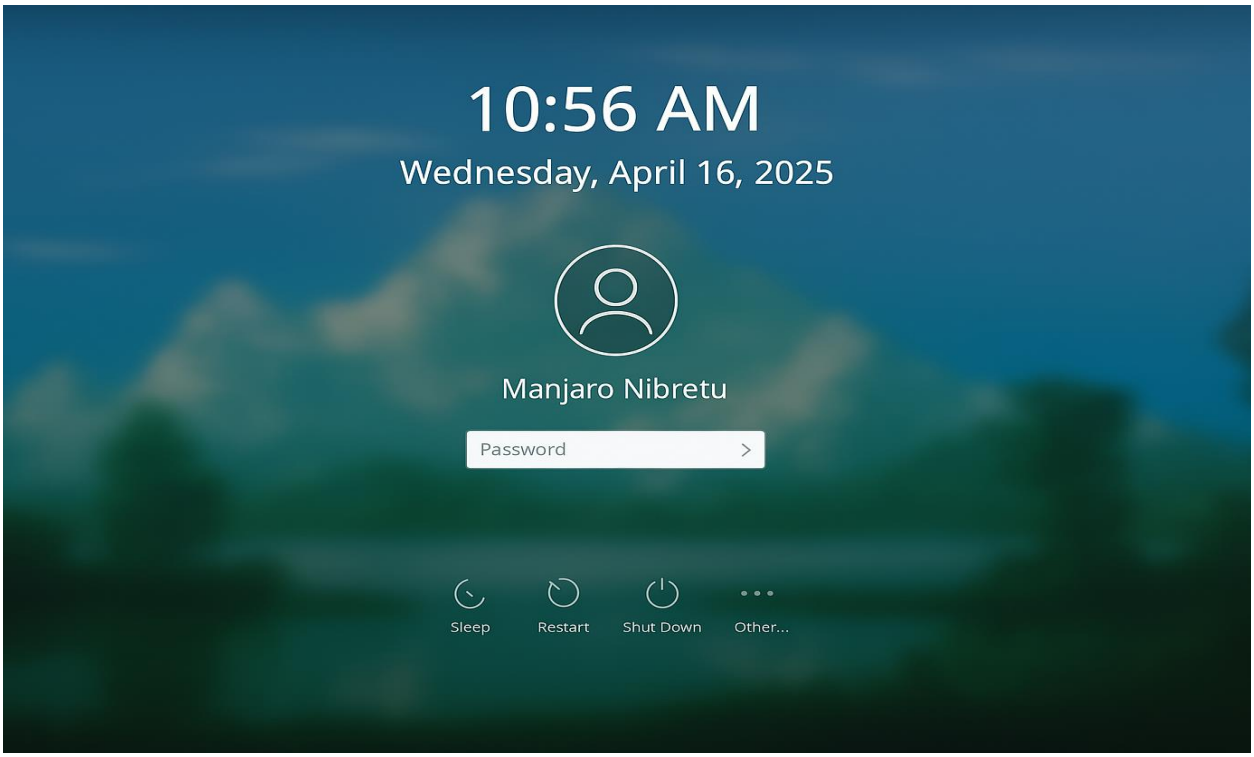

✓ The installation process will begin, copying files and configuring the system. Wait for it to complete.




✓ Once finished, ensure 'Restart now' is checked and click 'Done'.




- ✓ IMPORTANT: Before the VM restarts fully, go to the VM's 'Devices' -> 'Optical Drives' menu and click 'Remove disk from virtual drive' to unmount the ISO. If prompted, 'Force Unmount'. If you miss this, it might boot into the installer again.
- ✓ The VM will reboot into your installed Manjaro system. Login using the username and password you created.




**Step 7: Post-Installation (Install Guest Additions)**

- ✓ Open a Terminal in Manjaro (find it in the menu).
- ✓ First, update the system: `sudo pacman -Syu` (Enter your password when prompted. Type 'y' to proceed if asked).
- ✓ Install the VirtualBox Guest Additions utilities: `sudo pacman -S virtualbox-guest-utils` (Type 'y' to proceed).
- ✓ Enable the VirtualBox service: `sudo systemctl enable vboxservice`
- ✓ Reboot the VM: `reboot`
- ✓ After rebooting, features like automatic screen resizing, shared clipboard, and drag-and-drop should work (enable shared clipboard/drag-and-drop in the VM's 'Devices' menu).

## e. Issues (Problem Faced)

- **Potential Problem**
  - Screen resolution stuck at a low value (e.g., 800x600) after installation.
  - No network connection in the VM.

## f. Solution

- ✓ i. solution for screen resolution:- Install VirtualBox Guest Additions as described in Step 7 Post-Installation. After rebooting, the screen should resize automatically or allow selection of higher resolutions in Display settings.

- ✓ i. For network issue:- Check VM Network Settings (ensure Adapter 1 is enabled and set to NAT). Check host machine network connection. Restart the VM.
- ✓ Personally I would like to recommend If there is any other problem you faces while installing, fix them by searching on official website and read the documentation https://wiki.manjaro.org/index.php/Main_Page

## g. Filesystem Support

- ✓ Supported Filesystems:- Manjaro Linux, being Arch-based, has excellent filesystem support built into the kernel or available via packages. Native support includes:
  - ext4:- Very mature, stable, journaling filesystem. The common default for most Linux distributions, including Manjaro.
  - Btrfs:- Modern Copy-on-Write (CoW) filesystem with advanced features like snapshots, checksums, and built-in RAID. Often available as an option in the installer.
  - XFS:- High-performance journaling filesystem, good for large files and servers.
  - F2FS:- Filesystem optimized for flash-based storage (SSDs, NVMe).
  - FAT32/exFAT:- Supported for interoperability with Windows/macOS and external drives (requires `dosfstools`, `exfat-utils` packages, usually installed by default).
  - NTFS:- Supported for reading and writing Windows partitions via the `ntfs-3g` package (usually installed by default).
  - (Other filesystems like ZFS can often be installed but may require extra modules/setup).*
- ✓ Filesystem Chosen During Installation and Why:
  - ext4 was chosen automatically by selecting the 'Erase disk' option in the Calamares installer.
  - Why:- ext4 is the well-tested, reliable, and recommended default for Manjaro desktop installations. It offers good performance, stability, journaling (protects against data corruption from crashes), and is universally supported within the Linux ecosystem. While Btrfs offers more advanced features, ext4 is simpler and considered more robust for general desktop use, especially for users new to Linux or Arch-based systems.

## h. Advantage and Disadvantage (of Manjaro Linux)

- ➢ Advantages:
  - user-Friendly Arch:- Provides the power and software availability of Arch Linux (including the AUR) with an easier installation and setup process.
  - Rolling Release:- Access to up-to-date software packages shortly after they are released upstream.
  - Arch User Repository (AUR):- Vast community-maintained repository for software not in the official repos (use with caution).

- Good Hardware Detection:- Generally works well with a wide range of hardware out-of-the-box.
- Multiple Desktop Environments:- Official support for XFCE, KDE Plasma, and GNOME, plus community editions.

➢ Disadvantages:

- Rolling Release Stability: While Manjaro tests updates before release, rolling releases are inherently less stable than Point Release/LTS distributions (like Ubuntu LTS). Updates can occasionally cause issues.
- AUR Risks: Packages in the AUR are user-submitted; while often safe, they require more user scrutiny regarding security and stability.
- Can Require Troubleshooting: Being closer to Arch, users might occasionally need to use the command line or consult wikis (Arch Wiki is an excellent resource) to resolve issues, more so than beginner-focused distros.

### i. Conclusion

✓ Manjaro Linux (XFCE Edition) was successfully installed within VirtualBox following the steps outlined. The Calamares installer provided a straightforward process, using the default `ext4` filesystem on the virtual disk. Post-installation steps, specifically installing VirtualBox Guest Additions, were necessary for optimal integration and usability within the virtual environment. The process provided valuable hands-on experience with installing an Arch-based Linux distribution.

### j. Future Outlook / Recommendation

✓ This Manjaro VM can now be used to explore the Arch Linux ecosystem safely, experiment with software from the official repositories and the AUR, practice command-line usage, and customize the XFCE desktop environment.

✓ **Recommendation:-** Always keep the system updated using `sudo pacman -Syu`. When installing from the AUR, use an AUR helper (like `yay` or `pamac`) and review PKGBUILD files before installation. Ensure VirtualBox Guest Additions are kept up-to-date with kernel upgrades if necessary (usually handled by `virtualbox-guest-utils` package updates). Allocate sufficient RAM (4GB+) and disk space (30GB+) for a comfortable experience.

# 2. Virtualization in Modern Operating System

### a. What is Virtualization?

Define virtualization: Virtualization is the process of creating a virtual, rather than actual, version of computing resources. This can include virtual hardware platforms, operating systems, storage devices, or computer network resources. It allows for the abstraction of physical hardware, enabling multiple

operating systems or applications to run simultaneously on a single physical machine, isolated from one another.

- ✓ **Mention key concepts:**
  - **Host Machine**:- The physical computer hardware where the virtualization software runs.
  - **Guest Machine (VM):-** The virtual instance of an operating system and its applications running on the host machine.
  - **Hypervisor or Virtual Machine Monitor - VMM:-** The software, firmware, or hardware layer that creates, runs, and manages virtual machines. It allocates host resources (CPU, RAM, storage, network) to each guest VM.
- ✓ **Types of virtualization:**
  - **Hardware/Platform Virtualization**:- Creating VMs that simulate a complete hardware environment, allowing unmodified guest OSes to run (e.g., VMware Workstation/ESXi, VirtualBox, KVM, Hyper-V).
  - **OS-Level Virtualization (Containers):-** Virtualizing the operating system itself, allowing multiple isolated user-space instances (containers) to run on a single host kernel (e.g., Docker, LXC, Podman, Kubernetes). Does not simulate hardware.
  - **Desktop Virtualization:-** Running a user's desktop environment within a VM, often hosted remotely in a data center (VDI - Virtual Desktop Infrastructure).
  - **Network Virtualization (NV):-** Combining hardware and software network resources and functionality into a single, software-based administrative entity.
  - **Storage Virtualization:-** Pooling physical storage from multiple network storage devices into what appears to be a single storage device managed from a central console.

**b. Why is Virtualization Used?**

- ✓ Server Consolidation & Efficiency: Reduces the number of physical servers needed, leading to significant savings in hardware costs, power consumption, cooling, and physical space in data centers. Improves hardware utilization rates.
- ✓ Testing and Development: Provides isolated sandbox environments for developers and testers to build, test, and debug applications or OS updates without risking the stability of production systems or their primary workstations. Allows easy setup of specific configurations.
- ✓ Disaster Recovery & Business Continuity: Simplifies backup and recovery processes. Entire VMs (including OS, applications, and data) can be easily backed up, replicated to another site, and quickly restored on different hardware, minimizing downtime.
- ✓ Legacy Application Support: Enables running older applications that require outdated operating systems on modern hardware by encapsulating the legacy OS within a VM.
- ✓ Sandboxing & Security: Isolates applications or entire operating systems. If malware infects a guest VM, it is typically contained and cannot directly harm the host OS or other VMs. Useful for safely analyzing suspicious files or browsing untrusted websites.

- ✓ Running Multiple OSes: Allows users to run different operating systems (e.g., Windows, various Linux distributions, macOS) simultaneously on a single physical machine for compatibility, development, or learning purposes.
- ✓ Simplified Provisioning & Management:- VMs can be rapidly deployed from templates, cloned, and managed centrally using hypervisor management tools.
- ✓ Cloud Computing Foundation:- Virtualization is a fundamental technology underpinning Infrastructure as a Service (IaaS) cloud offerings, allowing cloud providers to offer scalable, on-demand computing resources.

**c. How is Virtualization Implemented in Modern Operating Systems?**

- ✓ **Hypervisors:-** The core enabling technology.
  - Type 1 (Bare Metal):- Installs directly onto the physical hardware, acting like a specialized OS itself (e.g., VMware ESXi, Microsoft Hyper-V Server, Xen, KVM within Linux). Offers high performance and efficiency, commonly used in data centers. KVM (Kernel-based Virtual Machine) is particularly notable as it's integrated directly into the Linux kernel, turning Linux itself into a hypervisor.
  - Type 2 (Hosted):- Runs as an application on top of a standard host operating system (e.g., VMware Workstation, Oracle VM VirtualBox, Parallels Desktop for Mac). Easier to set up and manage for desktop users but generally has slightly higher overhead than Type 1.
  - Hardware Assistance (CPU Virtualization Extensions):- Modern CPUs (Intel VT-x, AMD-V) include specific instructions and features that significantly accelerate virtualization. They allow the hypervisor to run guest OS code directly on the CPU in many cases, while efficiently trapping and emulating privileged instructions or sensitive operations that the guest OS tries to perform, improving performance and reducing hypervisor overhead. Technologies like Extended Page Tables (EPT by Intel) or Rapid Virtualization Indexing (RVI by AMD) improve memory management virtualization.
  - Paravirtualization (PV):- An approach where the guest operating system's kernel is modified to be "aware" that it's running in a virtualized environment. It makes specific calls (hypercalls) directly to the hypervisor for certain operations instead of relying on the hypervisor trapping hardware accesses. This can improve performance for certain tasks (like I/O) but requires guest OS modification. Modern hardware assistance has reduced the necessity of PV for performance, but PV drivers (e.g., VirtIO drivers for network and disk I/O) are still commonly used within fully virtualized guests for better performance.
- ✓ OS-Level Virtualization (Containers):

- Doesn't use a hypervisor in the traditional sense or simulate hardware. Instead, it leverages kernel features (like namespaces for isolation and cgroups for resource control in Linux) to run multiple isolated user-space environments directly on the host kernel.
- Examples include Docker, LXC, Podman, Kubernetes (for orchestration), FreeBSD Jails, Solaris Zones.
- Contrast with VMs: Containers share the host OS kernel, resulting in much lower overhead (less RAM/CPU usage, faster start times, higher density). However, they offer less isolation than VMs and all containers must run the same OS kernel as the host (e.g., you can't run a Windows container on a Linux host without a VM layer).

✓ Integration in Modern OSes:- Virtualization capabilities are increasingly integrated directly into mainstream operating systems:

- Windows:- Includes Hyper-V (can be enabled as a feature in Pro/Enterprise/Education editions), Windows Subsystem for Linux (WSL 2 uses a lightweight Hyper-V VM), Windows Sandbox.
- Linux:- KVM is built into the kernel. Various tools like QEMU, libvirt, virt-manager provide user-space components and management interfaces. Container technologies (LXC, Docker support) are native.
- macOS:- Includes a `Virtualization.framework` for developers to create VMs, and supports third-party hypervisors like VMware Fusion and Parallels Desktop.