****

**BAHIRDAR INSTITUTE OF TECHNOLOGY A**

**SOFTWARE ENGINEERING OPERATING SYSTEM**

**INDIVIDUAL ASSIGNMENT**

**NAME SILITU AGALU YENESEW**

**ID NUMBER BDU1602433**

**SECTION B**

**SUBMISSION DATE 16\08\2017 E.C**

**CONTENTS PAGE**

**1. Introduction……………………………….…………….3**

**2. Background………………………….…………………….3**

**3. Objective…………………………………………………..4**

**4. Requirments ………………………………………………4**

**5.issues and solution………………………………………...5**

**6. Installation steps ……………….…………………………5**

**i. Download the arch linux Iso…………….…………..6**

**Create a live USB or DVD……………………….….7**

**Boot up Arch Linux…………………………………9**

**Check internet connection…………….……………11**

**Enabling Networking Time Protocol(NTP)….…….11**

**Partition the Disk…………………………………..…..12**

**Create Filesystem……………………………..……….14**

**Check the Mirror List for an Application Mirror…..16**

**Install Arch Linux…………………………………..….17**

**Install G rub Bootloader………………………………..21**

**Create new user set up privileges………………………..24**

**xii.Exit Arch-Chroot Environment and Reboot……………..24**

**xii.Log in to Arch linux……………………………………..……25**

**7. Advantage………………………………………………………………………27**

**8. Disadvanage………………………………………………………………………28**

**9. Recommendation …………………………………………………………………..28**

**10. What,How and Why Virtualization in Modern OS……………………………..29**

**11. Implement system call…………………………………………………………..33**

**Documentation for Installation of Arch Linux in a Virtual Environment**

1. **Introduction (Background, Motivation)**

Arch [**Linux**](https://phoenixnap.com/kb/what-is-linux) is one of the most popular [**Linux distributions**](https://phoenixnap.com/glossary/what-is-a-linux-distribution) due to its versatility and low system requirements. It features a rolling release model, ensuring that users always have access to the latest Linux [**kernel**](https://phoenixnap.com/glossary/what-is-a-kernel) and software updates.

It is a lightweight and flexible Linux distribution that is known for its simplicity and customization options. It follows a rolling release model, which means it is continuously updated rather than requiring major version upgrades.

Arch Linux is an independently developed x86-64 general-purpose Linux distribution that strives to provide the latest stable versions of most software by following a rolling-release model. The default installation is intentionally minimal so that users can add only the packages they require.

pacman, a package manager written specifically for Arch Linux, is used to install, remove and update software packages. Arch Linux uses a rolling release model, meaning it has no major releases. An Arch Linux installation is kept up- to-date by regularly updating the individual pieces of software that it comprises. The only "releases" are snapshots of main system components released monthly to provide an up-to-date installation medium.

Inspired by CRUX, another minimalist distribution, Judd Vinet started the Arch Linux project in March 2002. The name was chosen because Vinet liked the word's meaning of "the principal," as in "arch-enemy". Originally only for 32-bit x86 CPUs, the first x86\_64 installation ISO was released in April 2006. Vinet led Arch Linux until 1 October 2007, when he stepped down due to lack of time, transferring control of the project to Aaron Griffin. The migration to systemd as its init system started in August 2012, and it became the default on new installations in October 2012. It replaced the SysV-style init system, used since the distribution's inception.

2017 Support for the i686 architecture was officially dropped, with the last ISO for this architecture released in February 2017. The community derivative Arch Linux 32 was created to support i686 hardware.

2020 On February 24, Aaron Griffin announced his intention to transfer control of the project to Levente Polyak, following a voting period. This change also introduced a new two-year term for the Project Leader position.

2021 Arch Linux began including a guided installation script by default in its installation images. Additionally, developers considered porting Arch Linux packages to x86\_64-v3, targeting Intel Haswell processors.

2022 The Arch Linux developers started offering debug packages to assist in troubleshooting and development.

2024 Valve partnered with Arch Linux developers to support ongoing development efforts, focusing on build service infrastructure and secure signing enclave.

The motivation behind installing Arch Linux in a virtual environment is to explore its features, understand its package management system (pacman), and gain experience in configuring a Linux system from the ground up.

**b. Objectives**

• To install Arch Linux in a virtual environment using tools like VMware Workstation or OracleVM VirtualBox.

• To learn about the installation process, configuration, and management of Arch Linux.

• To document the installation steps, issues encountered, and solutions found during the process**.**

**c. Requirements**

**i. Hardware**

• A computer with at least 4 GB of RAM (8 GB recommended for better performance).

• A minimum of 20 GB of free disk space.

• A compatible CPU with virtualization support (Intel VT-x or AMD-V).

**ii. Software**

• VMware Workstation or Oracle VM VirtualBox installed on the host machine

• Arch Linux ISO image, which can be downloaded from the official Arch Linux website**.**

* Internet connection.
* A [**USB**](https://phoenixnap.com/glossary/what-is-a-usb) drive or DVD with at least 2GB of free space for the ISO image.
* USB stands for **Universal Serial Bus**, a widely used industry standard for connecting electronic

devices, including computers and peripherals, for data transfer and power delivery

**Important:** The installation requires [**formatting our hard drive**](https://phoenixnap.com/kb/linux-format-disk). D**oin**g this removes all saved data. Make sure to back up relevant data before starting the installation process.

**The common issues:**

**1.partition issue**

Too Few Partitions: if we might not create enough partitions, which can limit our flexibility later. it could lead to problems later, like not having enough space for our files or even making our system unbootable.

• Wrong File System Type: we might choose a file system that isn’t supported or isn’t ideal for our needs.

Solution:

1. Plan our Partitions: Before we start, sketch out how we want our partitions.

2. Use a Partition Tool: During the installation, use tools like fdisk or parted to create our partitions. They provide a visual representation that can help us see how much space we’re allocating.

3. Check our Work: After creating partitions, double-check that they are set up as intended. Make sure we have the right file system (like ext4 for Linux) and that they are the correct sizes.

2.package installation error:

• Bad Internet Connection: If our connection drops, packages won’t download properly.

• Outdated Mirrors: Sometimes, the servers where packages are stored can be slow or outdated.

**Solution:**

1. Check our Internet Connection: Make sure we’re connected to the internet. Use commands like ping google.com to test connectivity.

2. Update our Mirror List: If the download is slow or failing, we might need to update the list of servers where packages are stored.

**How to Install Arch Linux**

Installing Arch Linux gives us complete control over what is installed, allowing us to tailor the system to our needs. The installation process also helps us understand how Linux works, from partitioning our disk to configuring the **[bootloader](https://phoenixnap.com/glossary/bootloader" \t "_blank)** ([**GRUB**](https://phoenixnap.com/kb/what-is-grub)).

Follow the steps outlined below to install Arch Linux.

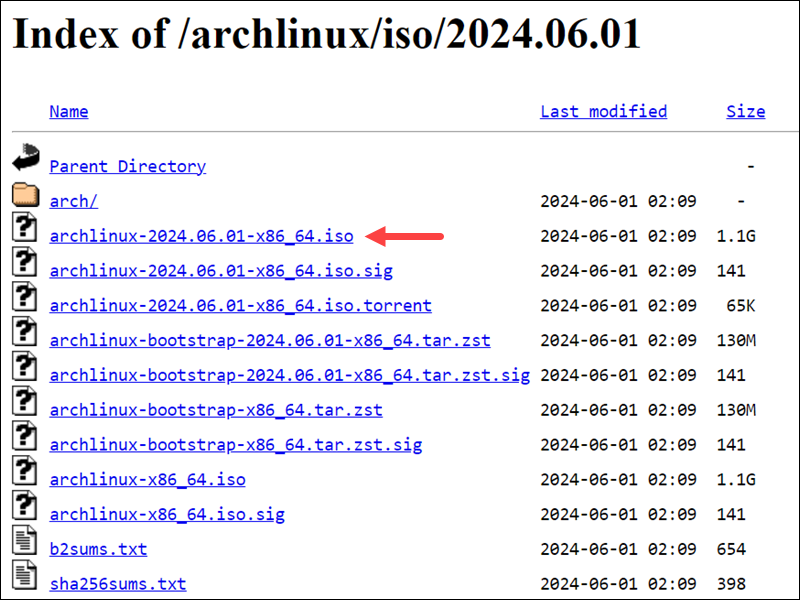
* [Step 1: Download the Arch Linux ISO](https://phoenixnap.com/kb/arch-linux-install#Step_1_Download_the_Arch_Linux_ISO)
* [Step 2: Create a Live USB or DVD](https://phoenixnap.com/kb/arch-linux-install#Step_2_Create_a_Live_USB_or_DVD)
  + [Create an Arch Linux Live USB](https://phoenixnap.com/kb/arch-linux-install#Create_an_Arch_Linux_Live_USB)
  + [Burn the Arch Linux ISO to a DVD](https://phoenixnap.com/kb/arch-linux-install#Burn_the_Arch_Linux_ISO_to_a_DVD)
* [Step 3: Boot up Arch Linux](https://phoenixnap.com/kb/arch-linux-install#Step_3_Boot_up_Arch_Linux)
* [Step 4: Check our Internet Connection](https://phoenixnap.com/kb/arch-linux-install#Step_5_Check_Your_Internet_Connection)
* [Step 5: Enable Network Time Protocols (NTP)](https://phoenixnap.com/kb/arch-linux-install#Step_6_Enable_Network_Time_Protocols_NTP)
* [Step 6: Partition the Disks](https://phoenixnap.com/kb/arch-linux-install#Step_7_Partition_the_Disks)
* [Step 7: Create Filesystem](https://phoenixnap.com/kb/arch-linux-install#Step_8_Create_Filesystem)
* [Step 8: Check the Mirror List for an Appropriate Mirror](https://phoenixnap.com/kb/arch-linux-install#Step_9_Check_the_Mirror_List_for_an_Appropriate_Mirror)
* [Step 9: Install Arch Linux](https://phoenixnap.com/kb/arch-linux-install#Step_10_Install_Arch_Linux)
* [Step 10: Configure Arch Linux](https://phoenixnap.com/kb/arch-linux-install#Step_11_Configure_Arch_Linux)
  + [Generate the fstab File](https://phoenixnap.com/kb/arch-linux-install#Generate_the_fstab_File)
  + [Use Arch-Chroot and Enter the Mounted Disk as Root](https://phoenixnap.com/kb/arch-linux-install#Use_Arch-Chroot_and_Enter_the_Mounted_Disk_as_Root)
  + [Set the Time Zone](https://phoenixnap.com/kb/arch-linux-install#Set_the_Time_Zone)
  + [Set the Locale](https://phoenixnap.com/kb/arch-linux-install#Set_the_Locale)
  + [Set the Hostname File](https://phoenixnap.com/kb/arch-linux-install#Set_the_Hostname_File)
  + [Set the Root Password](https://phoenixnap.com/kb/arch-linux-install#Set_the_Root_Password)
* [Step 11: Install Grub Bootloader](https://phoenixnap.com/kb/arch-linux-install#Step_12_Install_Grub_Bootloader)
  + [Install GRUB Bootloader on a Non-UEFI System](https://phoenixnap.com/kb/arch-linux-install#Install_GRUB_Bootloader_on_a_Non-UEFI_System)
  + [Install GRUB Bootloader on a UEFI System](https://phoenixnap.com/kb/arch-linux-install#Install_GRUB_Bootloader_on_a_UEFI_System)
* [Step 12: Create a New User and Set up Privileges](https://phoenixnap.com/kb/arch-linux-install#Step_13_Create_a_New_User_and_Set_up_Privileges)
* [Step 13: Exit Arch-Chroot Environment and Reboot](https://phoenixnap.com/kb/arch-linux-install#Step_14_Exit_Arch-Chroot_Environment_and_Reboot)
* [Step 14: Log in to Arch Linux](https://phoenixnap.com/kb/arch-linux-install#Step_15_Log_in_to_Arch_Linux)

**Step 1: Download the Arch Linux ISO**

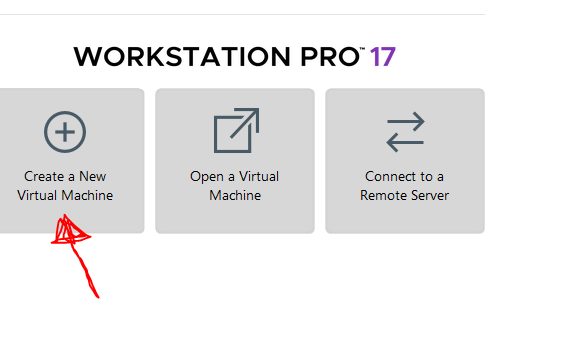
Download the ISO from the [**Arch Linux download page**](https://www.archlinux.org/download/). There are two ways to do so:

* Via BitTorrent
* As a direct download

To download the ISO via torrent, choose between adding a magnet link to our BitTorrent [**app**](https://phoenixnap.com/glossary/what-is-an-application) or downloading the torrent [**file**](https://phoenixnap.com/glossary/what-is-a-file). Alternatively, scroll down the page until we find a mirror closest to our current location and download the ISO file:

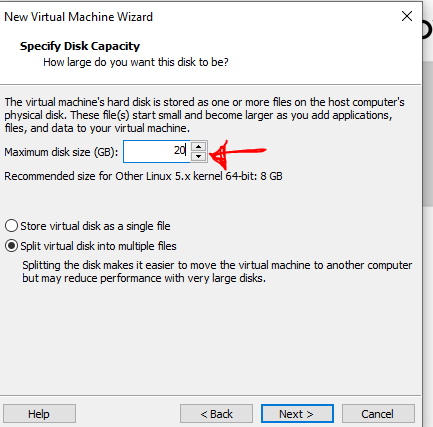


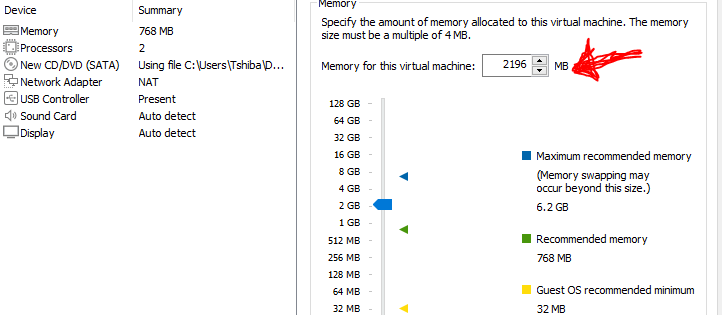
**Step 2: Create a Live USB or DVD**

****

Once we have the Arch Linux ISO file, create a live USB or burn it to a DVD.

**Create an Arch Linux Live USB**

****

****

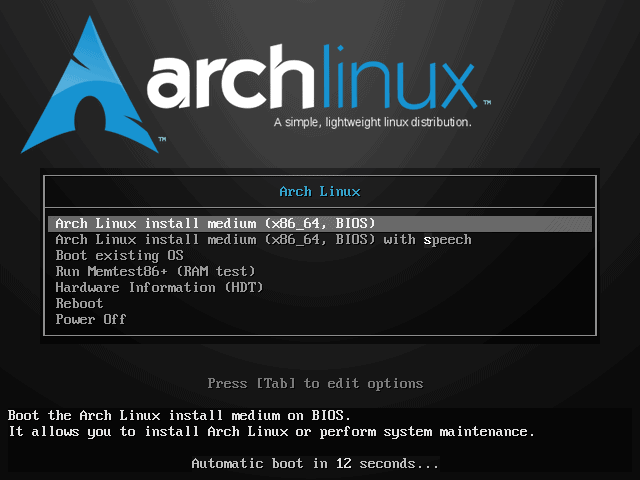
**Step 3: Boot up Arch Linux**

Start the installation process by [**booting**](https://phoenixnap.com/glossary/boot-definition) Arch Linux from the installation media. Follow the steps below:

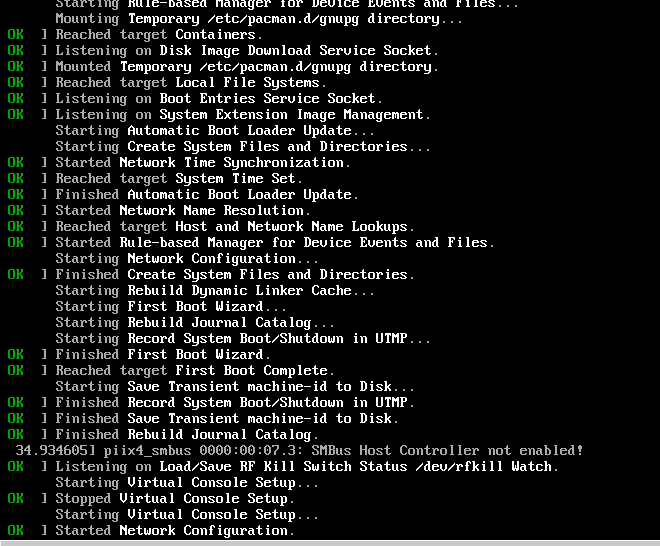
1. Insert the installation media into our machine and restart.

2. Depending on our system, press **F2**, **F10**, or **F12** to enter the boot menu and select the device from which the system will boot.

3. With the boot menu open, select the preferred installation media (live USB or DVD). The following screen shows up after Arch Linux boots:

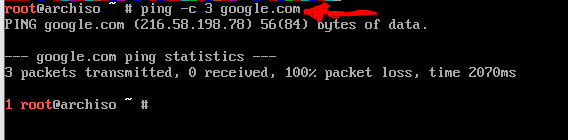


Select **Arch Linux install medium (x86\_64)** and press **Enter**to start the setup process.



**Step 4: Check our Internet Connection**

To check Internet connection using the [**ping command**](https://phoenixnap.com/kb/linux-ping-command-examples):



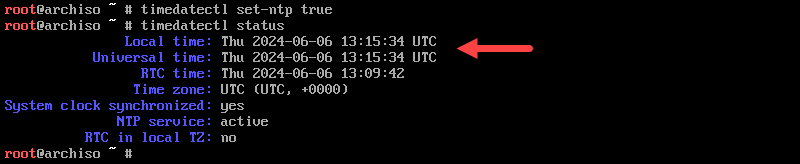
To install Arch Linux using a wireless Internet connection, consult the Arch Linux  **[wireless network configuration](https://wiki.archlinux.org/index.php/Wireless_network_configuration" \t "_blank)** section.

**Step 5: Enable Network Time Protocols (NTP)**

Next, enable [**Network Time Protocols (NTP)**](https://phoenixnap.com/glossary/network-time-protocol) and allow the system to update the time via the Internet:

NTP, or Network Time Protocol, is a networking protocol that synchronizes computer clocks across a network, ensuring accurate timekeeping for various applications.

To check the NTP service status, use:



**Step 6: Partition the Disks**

We will create an EFI and root partition. Other partition schemas are available, depending on the [**firmware**](https://phoenixnap.com/glossary/firmware) ([**BIOS or UEFI**](https://phoenixnap.com/kb/uefi-vs-bios)) and user preferences.

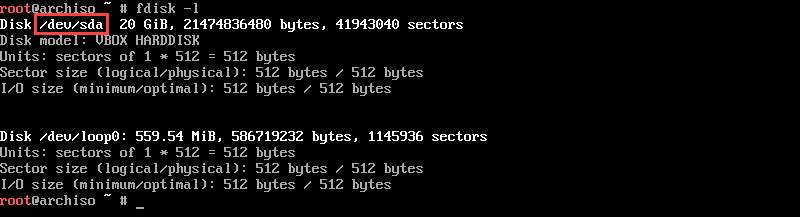
BIOS (Basic Input/Output System) is the older firmware, while UEFI (Unified Extensible Firmware Interface) is a newer, more advanced standard designed to replace it, offering features like faster boot times, a graphical interface, and improved security.

Follow the steps below:

1. Use the **fdisk** command to list all available disk drives:

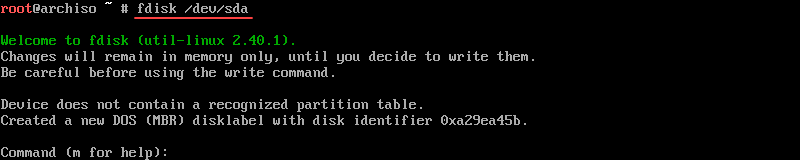
2. Find the name of the disk want to partition. In this context a "partition" refers to a logical division of a physical storage device (like a hard drive) into separate sections, each treated as a distinct drive by the operating system

The name is displayed in the **/dev/sda** format, where **a** is the drive letter.



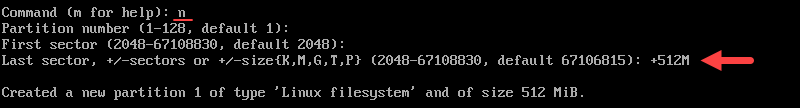
**Note:** When reviewing the list of available disk drives, ignore the ones ending in [**rom**](https://phoenixnap.com/glossary/what-is-rom) (read-only devices), **loop** (loopback devices), or **airoot** (root **[filesystem](https://phoenixnap.com/glossary/filesystem" \t "_blank)** used by the Arch Linux live environment).

3. Partition the drive by adding the drive name to **fdisk**. The syntax is:

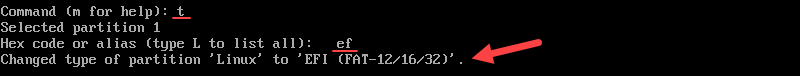


4. Create an EFI partition at the beginning of the disk. Type **n** and press **Enter**to [**create a new partition**](https://phoenixnap.com/kb/linux-create-partition). The system prompts you for the *disk number*, the *block size*, and the *partition size*. For the first two, keep the default values and press **Enter**.

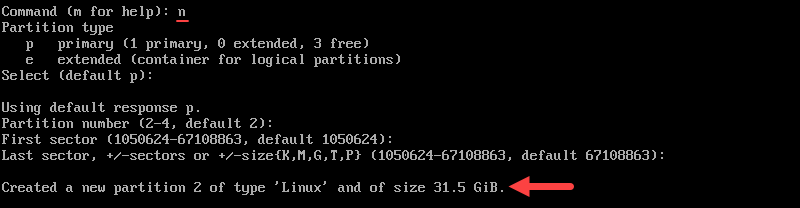
For the partition size, type **512M** and confirm with **Enter**. We can allocate more space, but 512M should be enough.



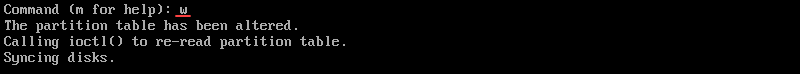
5. Change the EFI partition type from Linux filesystem to EFI system. Type **t** and press **Enter**. When prompted, type **ef** and press **Enter** to format the partition as EFI.



6. The next step is to create the root partition from the remaining free space. Type **n** and press **Enter**to create a new partition. Keep pressing **Enter**to accept all the default values until the system creates a new partition from the remaining free space.



7. Finally, when we finish partitioning the disk, type **w** and press **Enter**to write the changes to the disk.



To quit(exit or withdraw from) **fdisk**, type **q** and press **Enter**.

**Step 7: Create Filesystem**

The next step is to format the new partitions to install Arch Linux. To do this, create a file system for each of the partitions.

**ext4:**

* ext4 (Fourth Extended File System) is a widely used, general-purpose file system that has historically been the default for many Linux distributions, including Arch Linux.
* It's known for its extensibility and backward compatibility, meaning it can handle filesystems previously intended for ext2 and ext3.
* It offers features like journaling, which helps prevent data loss during crashes, and supports large files and disks.
* **Btrfs (Brfs):**
  + A modern, advanced file system primarily used in Linux, offering features like snapshots, data deduplication, and RAID support.
  + It's not as widely used as NTFS or FAT32, and it's not compatible with Windows.
  + Btrfs is known for its reliability and advanced features, making it a good choice for Linux systems that require those features.

**FAT32 (File Allocation Table 32):**

* An older file system, widely used for USB flash drives and digital cameras due to its compatibility with various operating systems.
* It has a limitation of not supporting files larger than 4 GB.
* It's a good choice for removable drives that need to be compatible with multiple operating systems, but it's less suitable for internal storage in Windows system

**NTFS (New Technology File System):**

* The default file system for Windows, offering features like advanced security, file journaling, and support for large files and partitions.
* It's generally faster and more reliable than FAT32, making it suitable for internal storage in Windows systems.
* However, NTFS is less compatible with other operating systems like macOS and Linux, which may require additional software or drivers for read/write access

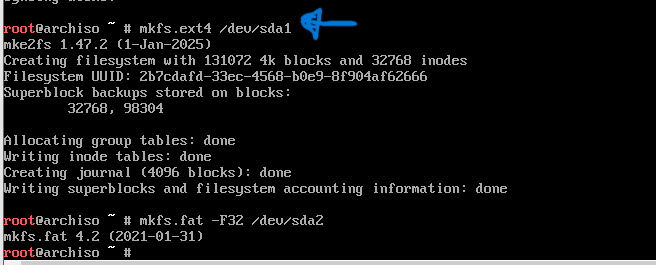
Follow the steps below:

1. Use the **mkfs** command to create a FAT32 filesystem for the EFI partition:

C:\Users\Tshiba\Pictures\FS.PNG

**a is** the drive letter of the disk to which the partition belongs.

2. Next, create an [**ext4**](https://phoenixnap.com/glossary/ext4) file system for the bootable partition

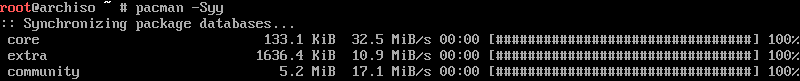


**Step 8: Check the Mirror List for an Appropriate Mirror**

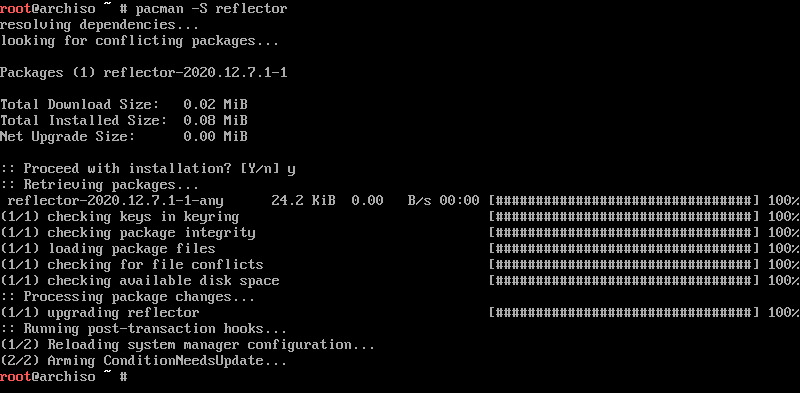
The Arch Linux installation downloads the necessary files through a mirror. Downloading files from a mirror far from our location slows down the process, which eventually causes the installation to fail.

To speed up the download, set up the mirror list so that the fastest mirrors are at the top.

1. Start by syncing the **pacman** [**repository**](https://phoenixnap.com/glossary/what-is-a-repository):



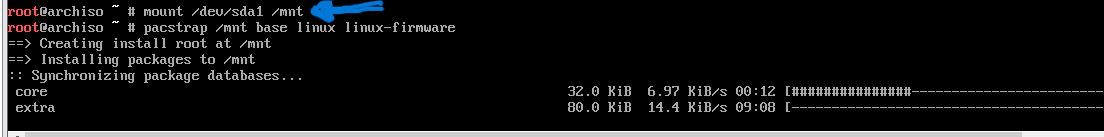
2. Install **reflector** to be able to update the mirrors and sort them by download speed. Add **reflector** by running:



**Step 9: Install Arch Linux**

Follow the steps below to install Arch Linux:

1. Mount the **root** partition with the following syntax:



**a** is our drive letter.

2. Next, use the **pacstrap** [**script**](https://phoenixnap.com/glossary/what-is-a-script) to install the necessary packages to the bootable partition:



Depending on our download speed, the installation process might take some time.

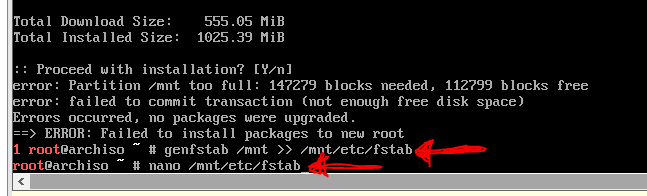
**Step 10: Configure Arch Linux**

Once the Arch Linux installation is complete, we must configure some settings.

**Generate the fstab File**

The **fstab** file defines the order in which disk partitions, block devices, remote devices, and other data sources are mounted.

Create a **fstab** file by running:

****

**Use Arch-Chroot and Enter the Mounted Disk as Root**

Change the **root** to the newly installed Arch Linux system with the **arch-chroot** command:

Switch to root in Arch Linux.

**Set the Time Zone**

Setting the correct time zone ensures the system clock reflects the accurate local time. Follow the steps below:

1. List all the available time zones:



we can also list the [**subdirectories**](https://phoenixnap.com/glossary/what-is-a-subdirectory) using the following syntax:

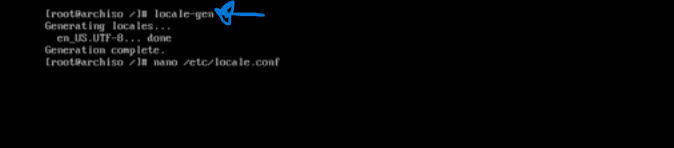
2. Find our time zone and make a note of the name.

3. Use the **ln** command to create a symbolic link from our desired timezone to */etc/localtime*. Replace **[Zone/SubZone]** with our actual timezone:

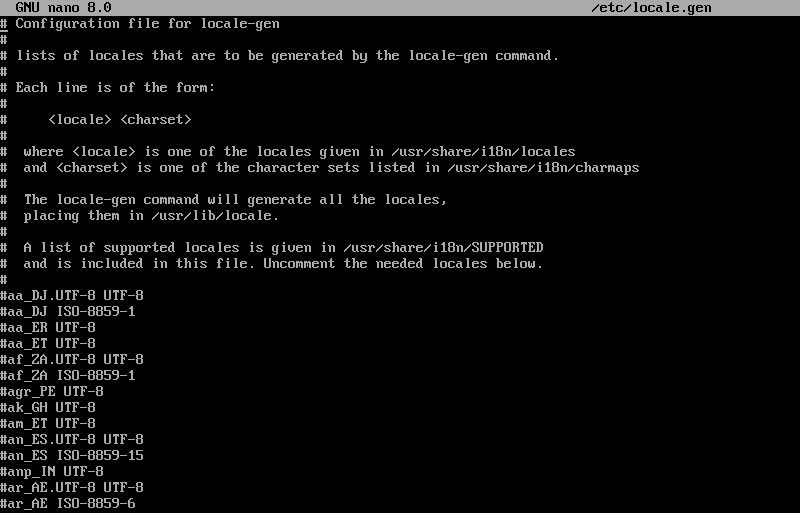
**Set the Locale**

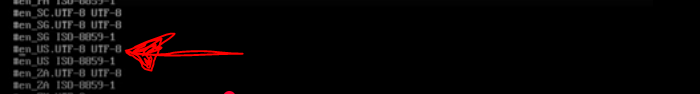
Set up the locale to determine the language, date, numbering, and currency format for our system. The **locale.gen** file contains a list of all available locales.

1. Open the file with **[nano](https://phoenixnap.com/kb/use-nano-text-editor-commands-linux" \t "_blank)** and find the name of our preferred locale:

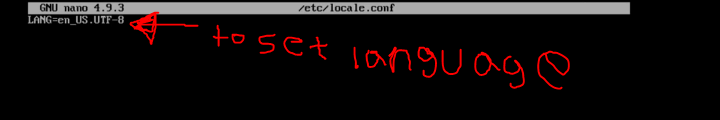


2. Uncomment the name (remove the **#**) of our preferred locale and any other we would like to use.





3. Press **Ctrl + X**to exit and type **Y** to save the changes.



4. Generate a locale [**configuration file**](https://phoenixnap.com/glossary/config-file) by running the following commands:

Replace **[locale\_name]** is the name of our preferred locale. For example:



**Note:** We can also change the time zone and locale later while using our Arch Linux system.

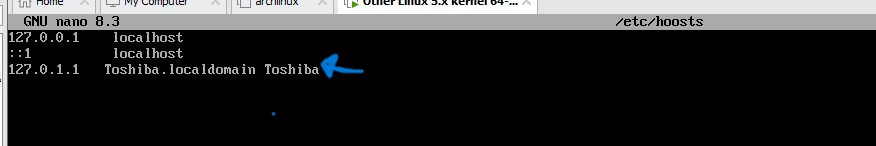
**Set the Hostname File**

The hostname is a unique identifier for a machine within a network. It is especially important in environments with multiple devices, as it helps differentiate between various systems.

1. Create a **hostname** file and add our hostname with the following syntax:

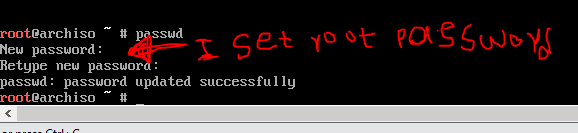
echo [our\_hostname] > /etc/hostname

2 [**Edit the hosts file**](https://phoenixnap.com/kb/how-to-edit-hosts-file-in-windows-mac-or-linux) and add the following content:



**Set the Root Password**

Set up a new root password with the **[passwd command](https://phoenixnap.com/kb/passwd-command-in-linux" \t "_blank)**:



Running this command prompts me to type and then retype my new password.

**Tip:** Simplify and speed up password creation with our [**strong password ideas**](https://phoenixnap.com/blog/strong-great-password-ideas) or our [**free password generator**](https://phoenixnap.com/kb/password-generator).

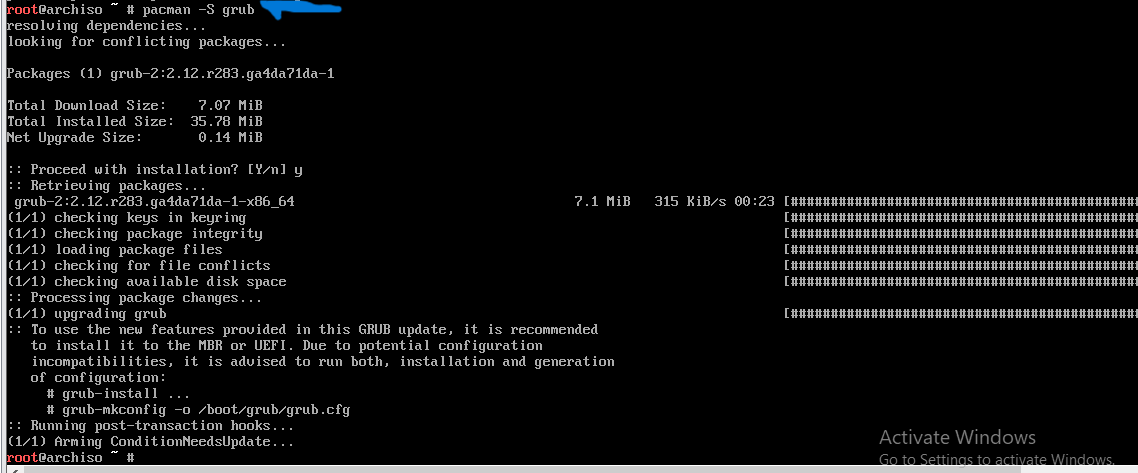
**Step 11: Install Grub Bootloader**

Next, install the GRUB bootloader. There are two ways to install GRUB, depending on whether we are using a non-UEFI or [**UEFI**](https://phoenixnap.com/glossary/uefi-unified-extensible-firmware-interface) system.

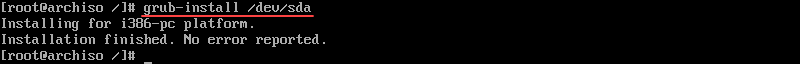
**Install GRUB Bootloader on a Non-UEFI System**

A non-UEFI system uses the traditional [**Basic Input/Output System (BIOS)**](https://phoenixnap.com/glossary/basic-input-output-system-bios) firmware interface for booting and system initialization, as opposed to the newer Unified Extensible Firmware Interface (UEFI) standard.

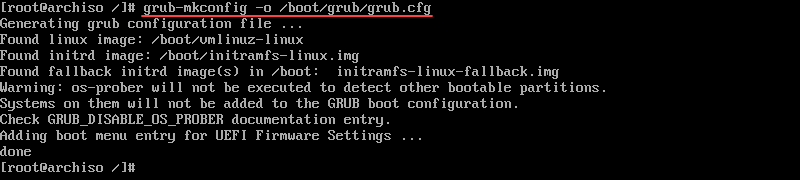
1. Add the GRUB bootloader packages with the **pacman** [**package manager**](https://phoenixnap.com/glossary/what-is-a-package-manager):



2. Install the GRUB bootloader:



3. Create a GRUB configuration file with:



**Install GRUB Bootloader on a UEFI System**

A UEFI system is a modern firmware standard that replaces the traditional BIOS. It provides enhanced boot, security, and system management capabilities.

1. Add the GRUB bootloader packages with the **pacman** manager:

pacman -S grub efibootmgr

2. Create a directory for the EFI partition:

mkdir /boot/efi

3. Mount the EFI partition to the directory we created:

mount /dev/sda1 /boot/efi

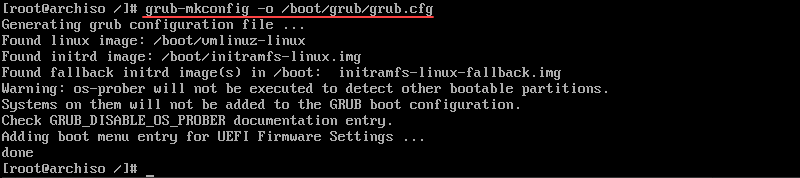
Replace **X** with the drive letter of the disk the partition belongs to.

4. Install GRUB with:

grub-install --target=x86\_64-efi --bootloader-id=GRUB --efi-directory=/boot/efi

Install GRUB on EFI Arch Linux system.

5. Finally, create a GRUB configuration file:



**Step 12: Create a New User and Set up Privileges**

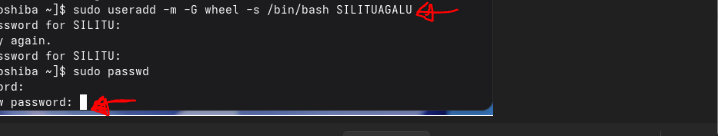
This step is optional but recommended. Having a system with only the root user does not require authentication for any changes, making it vulnerable and susceptible to failure.

Follow the steps below to [**create a new user**](https://phoenixnap.com/kb/linux-user-create):

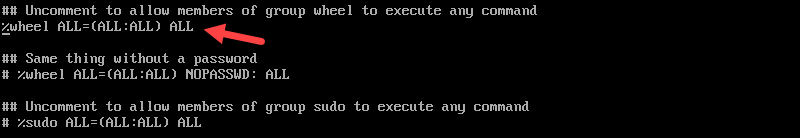
1. Install **[sudo](https://phoenixnap.com/kb/linux-sudo" \t "_blank)** before adding a new user:

2. Create a new user with this syntax:

3. Create a password



5. Lastly, edit the *visudo*file, specifically the line referring to the **wheel** group and uncomment it:



6. Save the changes and exit the file.

**Step 13: Exit Arch-Chroot Environment and Reboot**

Finally, [**reboot the machine**](https://phoenixnap.com/kb/restart-linux-using-command-prompt) to load the new Arch Linux installation.

1. Exit the **arch-chroot** environment:

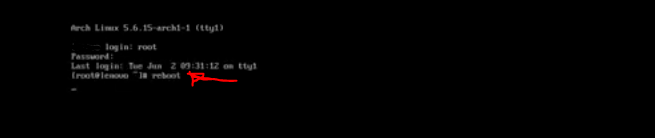
exit

2. Unmount the **root** partition with:

umount -l /mnt

3. Then, remove the USB or DVD and reboot the system with:

sudo reboot

****

**Step1 4: Log in to Arch Linux**



Once the system reboots, GRUB loads and allows us to load Arch Linux. Press **Enter** to load the system. Use the password we set up to log in.



****

****

****

**Advantages:**

* **Minimalist and Customizable:**

Arch Linux provides a bare-bones installation, allowing users to choose and install only the necessary software and packages, leading to a highly customized and efficient system.

* **Rolling Release:**

As a rolling release distribution, Arch Linux receives continuous updates, ensuring users always have the latest software and security patches, without the need for major reinstallation.

* **Large Repository:**

The Arch User Repository (AUR) provides access to a vast amount of community-maintained software, expanding the range of available applications.

* Excellent Documentation:

Arch Linux boasts extensive and well-maintained documentation, including the Arch Wiki, which serves as a valuable resource for users of all levels.

* **Fast and Lightweight:**

Due to its minimalist approach and optimized package management, Arch Linux tends to be fast and lightweight, even on older hardware.

**Disadvantages**:

* **Steep Learning Curve:**

Installing and configuring Arch Linux can be challenging, requiring a certain level of Linux knowledge and command-line proficiency, making it less beginner-friendly.

* **Potential Instability:**

As a rolling release distribution, updates can occasionally introduce instability or break existing software, requiring users to be vigilant and proactive in troubleshooting.

* **Manual Installation:**

The installation process is command-line based, requiring users to manually configure the system, including partitioning, package selection, and desktop environment installation.

* **Requires Active Maintenance:**

Users must regularly monitor and maintain their system, including updating packages and resolving any issues that may arise, which can be time-consuming.

* **Limited Commercial Support:**

Arch Linux is a community-driven project, and there is no official commercial support, meaning users must rely on community forums and documentation for assistance.

**CON**Recommendation

* **For Experienced Linux Users:**

Arch Linux is an excellent choice for users who enjoy customizing their systems and have a good understanding of Linux concepts.

* **For Developers:**

The ability to easily access the latest software and customize the system makes arch linux a popular choice for developers.

* **For Users Seeking a Minimal and Fast System:**

The minimal nature and rolling release model make Arch Linux ideal for users who want a fast and up-to-date system without unnecessary bloat.

* **For Absolute Beginners:**

Consider exploring other Linux distributions, such as Ubuntu or Linux Mint, before attempting to install Arch Linux.

* **For those who want to learn Linux:**

Arch Linux is a great way to learn about how Linux works, as it forces users to understand the system and its components.

**TA**

**Conclusion**

After following this guide, we have installed and configured Arch Linux on our computer.

Arch Linux is a highly customizable, minimalist, and up-to-date system that allows users full control over their operating environment and software.

It is ideal for advanced users who want a tailored computing experience.

Installing Arch Linux in a virtual environment provides valuable experience in understanding Linux systems.

The process reinforces knowledge of system configuration, package management, and troubleshooting**.**

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

**What**: Virtualization is the creation of a virtual version of a physical computer such as CPU,Memory and storage, enabling multiple operating systems or applications to run on the same hardware.

It achieves this by abstracting the hardware layer, allowing software to simulate hardware functionality and create virtual machines (VMs).

It is a technology that helps us to install different Operating Systems on hardware. They are completely separated and independent from each other.

Virtualization hides the physical characteristics of computing resources from their users, their applications or end users.

This includes making a single physical resource (such as a server, an operating system, an application or a storage device) appear to function as multiple virtual

resources.

It can also include making multiple physical resources (such as storage devices or servers) appear as a single virtual resources.

Virtualization in modern operating systems allows multiple operating systems or applications to run concurrently on a single physical machine by creating virtual environments, which enhances resource utilization and flexibility

**Why:** Virtualization offers several benefits, including:

**Resource Optimization:**

It allows better utilization of hardware resources by running multiple VMs on a single server, reducing hardware costs and improving efficiency.

**Flexibility and Scalability**

VMs can be easily created, moved, and scaled, providing flexibility in managing workloads and adapting to changing business needs.

**Cost Reduction:**

By consolidating hardware and reducing the number of physical servers, virtualization significantly lowers IT costs.

**Improved Disaster Recovery:**

Virtual machines can be easily backed up and restored, simplifying disaster recovery procedures.

**Enhanced Security:**

VMs can be isolated from each other, improving overall system security.

**How:** Virtualization is implemented through software, often using a hypervisor, which acts as a layer between the hardware and the virtual machines.

The hypervisor manages the hardware resources and allows multiple operating systems to run concurrently, each with its own virtualized hardware environment.

With virtualization, VMs are created to run their own OS and applications.

Virtualization allows multiple operating systems to run alongside each other and share the same virtualized hardware resources from a single physical machine. Containerization packages software code together into its own container.

Virtualization depends on 2 important concepts: virtual machines and hypervisors.

**Virtual machines**

A virtual machine (VM) is a computing environment that functions as an isolated system with its own CPU, operating system (OS), memory, network interface, and storage, created from a pool of hardware resources.

A VM can be defined by a single data file. As an isolated environment, it can be moved from 1 computer to another, opened in either, and be expected to work the same.Virtualization allows virtual machines with multiple different operating systems to run simultaneously on a single physical device—like running a MacOS or Windows environment on a Linux system.

Each operating system runs in the same way an OS or application normally would on the host hardware, so the end user’s experience is nearly identical to a real-time operating system experience running on a physical machine.

**Hypervisors**

Sometimes called a virtual machine monitor (VMM), a hypervisor is software that

separates a system’s physical resources and divides those resources so that virtual

environments can use them as needed.

A hypervisor takes physical resources (such as CPU, memory, and storage) from the hardware and allocates them to multiple VMs at once, enabling the creation of new VMs and the management of existing ones.

Hypervisors can sit on top of an operating system (like on a laptop) or be installed directly onto hardware (like a server).

The physical hardware, when used as a hypervisor, is called the host, while the many VMs that use its resources are guests.

When the virtual environment is running and a user or program issues an instruction that requires additional resources from the physical environment, the hypervisor relays the request to the physical system and stores the changes in a cache—which all happens at close to native speed.

There are 2 different types of hypervisors that allow virtualization to happen based on need.

**Type 1:** Also referred to as a native or bare-metal hypervisor, it runs directly on the host’s

hardware to manage guest operating systems.

It takes the place of a host operating system, and VM resources are scheduled directly to the hardware by the hypervisor. This type of hypervisor is most common in an enterprise datacenter or other server-based environments.

**Type 2**: Also known as a hosted hypervisor, it runs on a conventional operating system as a software layer or application. It works by abstracting guest operating systems from the host operating system.VM resources are scheduled against a host operating system, which is then executed against the hardware. This type is better for individual users who want to run multiple operating systems on a personal computer.

END!!!