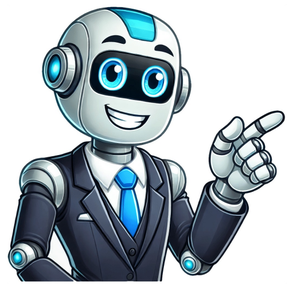
Click Here



Archlinux installation guide

Arch Linux is a widely used rolling release distribution among DIY enthusiasts and hardcore Linux users, offering a minimal base system that requires configuration by the user. Installing Arch Linux can be challenging but also serves as a learning opportunity for intermediate users. The installation process of Arch Linux has two options: using the guided installer (archinstaller) or configuring it manually without any assistance. However, this latter method may be more time-consuming and better suited for experienced users. Key requirements for installing Arch Linux include a compatible machine, sufficient RAM and disk space, an active internet connection, and familiarity with the Linux command line. It is essential to back up files before proceeding, as the installation process will wipe out existing operating systems. To create an Arch Linux live USB, one can use Etcher or the dd command. For those on Linux, using dd is a viable alternative. The process involves replacing the ISO file path and USB drive information in the command. For Etcher users who prefer a GUI, it's recommended to stick with that method. After creating the live USB, boot from it by pressing F2, F10, or F12 during startup, and select Boot from USB or removable disk. If secure boot is enabled, disable it first. Upon booting from the USB, select Arch Linux Boot Screen and choose Boot Arch Linux (x86_64). The system will proceed to the login prompt with the root user. To accommodate non-US keyboard layouts, list supported keymaps using ls /usr/share/kbd/keymaps/**/*.map.gz and change the layout using loadkeys command. For example, for German, use loadkeys de-latin1. Partitioning the disk involves using fdisk to list disks and partitions. Delete existing partitions using d, then create new ones with n. If UEFI mode is enabled, follow different steps that are clearly mentioned. For UEFI systems, create an ESP partition at the beginning of the disk. Otherwise, skip this step. To create an EFI System Partition, change its type to "EFI System" instead of Linux system. Enter 't' to change the type and 'L' to see all available partition types, then select the corresponding number for "EFI System". Create a root partition for both UEFI and legacy systems. A common practice is to create separate partitions for root, swap, and home, but you can also create a single root partition with a swapfile and home directory under it. When creating the disk partitioning in fdisk, press 'n' to create a new partition, which will automatically be given partition number 2. Allocate the entire remaining disk space to the root partition by keeping pressing enter. After finishing disk partitioning, enter 'w' to write changes to the disk and exit out of fdisk command. Next, create filesystems on your disk partitions using the following commands: - For UEFI systems: Create a FAT32 file system on the EFI type partition (mkfs.fat -F32 /dev/sda1) and an Ext4 filesystem on the root partition (mkfs.ext4 /dev/sda2). - For non-UEFI systems: Just make the single root partition ext4 (mkfs.ext4 /dev/sda1). Connect to WiFi interactively using the iwctl utility. Select your preferred wireless interface, scan for available networks, and connect to your target Wi-Fi. Finally, check if you're connected to the network by pinging google.com with the ping command. Firstly, ensure the top mirror is set correctly as it may not always be ideal. Fortunately, a solution exists. Begin by synchronizing the Pacman repository to download and install software: pacman -Syy Next, install Reflector to list fresh and fast mirrors in your country: pacman -S reflector Make a backup of the mirror list to avoid any issues: cp /etc/pacman.d/mirrorlist /etc/pacman.d/mirrorlist.bak Now, obtain the good mirror list with Reflector and save it to mirrorlist. You can change the country from US to your desired location: reflector -c "US" -f 12 -l 10 -n 12 --save /etc/pacman.d/mirrorlist All is now set for Arch Linux installation. Since everything is ready, proceed to install Arch Linux by mounting the root partition: mount /dev/sda2 /mnt For UEFI systems, use /dev/sda1 instead. After mounting the root partition, utilize the Pacstrap script to install necessary packages: pacstrap /mnt base linux linux-firmware vim nano It may take some time for these packages to download and install. If interrupted downloads resume with a command run again. The text editors Vim and Nano are added because you'll need to edit files post-installation. Visit our nano vs vim article for differences between these editors. Next, generate a fstab file to configure the installed Arch system: genfstab -U /mnt >> /mnt/etc/fstab Now enter the mounted disk as root using arch-chroot and make configuration changes to run the system properly on boot. Setting up the timezone involves timedatectl command. First, find your desired time zone and exit the list with Ctrl + C or q. Set it like this (replace Europe/Paris with your preferred time zone): timedatectl set-timezone Europe/Paris Finally, configure the locale to set language, numbering, date, and currency formats: Open /etc/locale.gen in Vim or Nano editor and uncomment the desired language. Use en_GB.UTF-8 as an example and hit Ctrl + X and then Y to save and continue. Generate the locale config with commands: locale-gen echo To configure Arch Linux, start by editing the `/etc/locale.conf` file and setting the `LANG` variable to `en_GB.UTF-8`. This can be changed later using the locale and timezone settings. Next, create a `/etc/hostname` file and add the hostname entry. In this example, the hostname is set to `myarch`, but you can choose any name you prefer: `echo myarch > /etc/hostname`. Create the hosts file with `touch /etc/hosts` and edit it using an editor like Vim or Nano. Add the following lines, replacing `myarch` with your chosen hostname: `127.0.0.1 localhost ::1 localhost 127.1.1 myarch` Set a password for the root account using `passwd`. This is crucial and may differ between UEFI and non-UEFI systems. For UEFI systems: * Install required packages with `pacman -S grub efibootmgr`. * Create the `/boot/efi` directory: `mkdir /boot/efi`. * Mount the ESP partition: `mount /dev/sda1 /boot/efi`. * Install GRUB using `grub-install --target=x86_64-efi --bootloader-id=GRUB --efi-directory=/boot/efi`. * Run `grub-mkconfig -o /boot/grub/grub.cfg`. For non-UEFI systems: * Install the `grub` package: `pacman -S grub`. * Install GRUB on the disk: `grub-install /dev/sda`. * Run `grub-mkconfig -o /boot/grub/grub.cfg`. Create an additional user with privileges to avoid booting into a system with only a root account. This will prevent unauthenticated changes that could damage your system. Install the `sudo` package: `pacman -S sudo`. Then, create a new user and give permissions using `useradd`, `passwd`, and `usermod`. Finally, edit the `visudo` file to uncomment the wheel group line. You can install GNOME desktop environment on Arch Linux using `pacman -S gnome`. This will give you a list of packages to choose from. You can handpick specific packages or use ranges, such as `1,2,4-8,9`, to select multiple packages at once. Be careful not to skip any packages. After installation, enable the display manager GDM and Network Manager using `systemctl enable gdm.service` and `systemctl enable NetworkManager.service`. Then, exit the chroot using `exit`. Next, unmount the root partition with `umount /mnt` or `umount -l /mnt`, and shut down your system with `shutdown now`. Before powering on the system again, remove the live USB. If everything goes well, you should see the Grub screen followed by the GNOME login screen. If you want to install KDE instead, follow a different tutorial. For a basic installation, ensure your machine has at least 512 MiB RAM. Although more memory is recommended for live system booting during installation. A minimal disk space of less than 2 GiB is required for installation. Having an active internet connection is necessary due to the need to download packages from a remote repository. Boot Options Menu See README.bootparams for a list of available options. A common example of manually defined boot parameters is setting the font size for better readability on HiDPI screens. Using fbcon=font:TER16x32 can help, but refer to HiDPI#Linux console (tty) for detailed explanations. To modify partition tables, it's essential to plan a long-term strategy to avoid complicated conversions or re-partitioning procedures in the future. If you need stacked block devices for LVM, system encryption, or RAID, do them now. Be cautious when booting from a disk with an existing EFI system partition; instead, use the existing one and create another if needed. You can set swap space on a swap file for certain file systems or avoid disk-based swap by setting up zram after installing the system. For UEFI, you'll need: - A 1 GiB EFI system partition - A Linux swap at least 4 GiB - An Linux root partition, taking the remainder of the device (at least 23–32 GiB) For BIOS, it's similar but uses MBR instead. Once partitions are created, format each one with a suitable file system. For example, using mkfs.ext4 on /dev/root_partition. If you created a swap partition, initialize it with mkswap. Format the EFI system partition to FAT32 using mkfs.fat(8) only if you created it during partitioning; reformatting can damage boot loaders of other systems. Mount file systems by mounting the root volume to /mnt, then any remaining mount points. Genfstab will later detect mounted file systems and swap space. mirrors should be sorted by download rate. The higher the mirror is in the list, the priority it gets when downloading a package. It's advisable to inspect the file and edit it if needed. For geographically closest mirrors at the top of the list, other factors must also be taken into account as these files will later be copied to the new system by pacstrap. Install essential packages Note: All software and configuration except /etc/pacman.d/mirrorlist are not transferred from the live environment to the installed system. Pacstrap(8) is used to install the base package, Linux kernel, and firmware for common hardware: # pacstrap -K /mnt base linux linux-firmware You can replace linux with a preferred kernel package or omit it entirely when installing in a container. When installing in a virtual machine or container, you can also omit the firmware package. The base package may not have all tools from the live environment, so additional packages may be required for a fully functional system. Edit /etc/locale.gen and enable en_US.UTF-8 locale by uncommenting it, then generate locales with `locale-gen`. Create a locale.conf file setting LANG=en_US.UTF-8. Also, set the console keyboard layout in vconsole.conf(5) if needed. Create a hostname file at /etc/hostname with yourhostname as its content. Complete network configuration for the newly installed environment, which may involve installing and configuring network management software. Creating an initramfs is usually not necessary since mkinitcpio was run during kernel package installation with pacstrap. However, for LVM, system encryption, or RAID configurations, modify mkinitcpio.conf(5) and recreate the initramfs image with `mkinitcpio -P`. Set a root password using `passwd` and choose a Linux-capable boot loader to install. Reboot your machine after exiting the chroot environment by typing exit or pressing Ctrl+d. Optionally, unmount all partitions manually with umount -R /mnt before restarting. Post-installation tasks include managing system resources as per General recommendations, setting up a graphical user interface, and configuring sound or touchpad functionality. Pre-installation preparation Get the installation image Visit the Download page and grab the ISO file or netboot image, along with its PGP signature. Verify the signature It's recommended to verify the image signature before use, especially when downloading from an HTTP mirror, where downloads might be intercepted. On a system with GnuPG installed, do this by downloading the ISO PGP signature, placing it in the ISO directory, and verifying it with `gpg --keyserver-options auto-key-retrieve --verify archlinux-version-x86_64.iso.sig`. Alternatively, from an existing Arch Linux installation run `pacman-key -v archlinux-version-x86_64.iso.sig`. Note that if you're downloading from a mirror site instead of archlinux.org, ensure the public key used to decode the signature is signed by another trustworthy key. The gpg command will output the fingerprint of the public key. Prepare an installation medium The ISO can be supplied to the target machine via USB flash drive, optical disc, or network with PXE: follow the appropriate article to prepare yourself an installation medium from the ISO file. For the netboot image, follow Netboot#Boot from a USB flash drive to prepare a USB flash drive for UEFI booting. Boot the live environment Arch Linux installation images don't support Secure Boot, so disable it before booting the installation medium. If desired, Secure Boot can be set up after completing the installation. Point the current boot device to the one with the Arch Linux installation medium by pressing a key during the POST phase, as indicated on the splash screen. When the installation medium's boot loader menu appears, select Arch Linux install medium (if using ISO) or choose a geographically close mirror from Mirror menu, then select Boot Arch Linux and press Enter. Using Virtual Console and Live Environment in Arch Linux Installation ---------- 1. **Switching between virtual consoles:** Press Alt+arrow keys to switch between the virtual console and other applications. 2. Editing configuration files: Use mcedit(1), nano, or vim to edit configuration files. 3. Setting keyboard layout: List available layouts with `localectl list-keymaps` and set the desired layout using `loadkeys(1)`. 4. Setting console font: Change the font by running `setfont(8)` without the path and file extension. 5. Verifying boot mode: Check the UEFI bitness with `cat /sys/firmware/efi/fw_platform_size` to determine if the system is booted in UEFI or BIOS mode. 6. Connecting to the internet: Configure your network connection using the following steps: * Ensure your network interface is listed and enabled, for example, with `ip link(8)`. * Connect to the wireless network using `iwctl` if you're using Wi-Fi. * Connect to mobile broadband modem using `mmcli utility`. 7. Updating system clock: Verify that the system clock is synchronized by running `timedatectl(1)` in the live environment. 8. Partitioning disks: Use a partitioning tool like fdisk to modify partition tables and create stacked block devices for LVM, system encryption or RAID if desired. EFI System Partition must be used instead of creating another one, and swap space can be set on a swap file for file systems supporting it, or zram after installing the system. Example layouts - UEFI with GPT: Mount point on installed system, Partition Partition type Suggested size /boot1 /dev/efi_system_partition EFI system partition 1 GiB [SWAP] /dev/swap_partition Linux swap At least 4 GiB / /dev/root_partition Linux x86-64 root (/) Remainder of device. At least 23–32 GiB. - BIOS with MBR: Mount point on installed system, Partition Partition type Suggested size [SWAP] /dev/swap_partition Linux swap At least 4 GiB / /dev/root_partition Linux Remainder of device. At least 23–32 GiB. File systems must be formatted after creating partitions. Use mkfs.ext4 for ext4 file system or mkswap(8) for swap, and mkfs.fat(8) for EFI system partition. Example: # mkfs.ext4 /dev/root_partition # mkswap /dev/swap_partition If you created an EFI system partition, format it with mkfs.fat(8). Mount the file systems under /mnt. Mount the root volume to /mnt and create any remaining mount points under /mnt. When copying files to a new system using pacstrap, it's crucial to pay attention to other necessary criteria. This file will later be duplicated by pacstrap, so precision is recommended. Essential software packages need to be installed: Note that no configuration or software (excluding /etc/pacman.d/mirrorlist) from the live environment gets transferred to the newly installed system. To install base packages, including the Linux kernel and firmware for common hardware, utilize the pacstrap(8) script with the following command: # pacstrap -K /mnt base linux linux-firmware Tip: Substitute linux with a preferred kernel package or omit it when setting up in a container. You may also choose to skip firmware installation if using a virtual machine or container. The base package does not include all tools from the live installation, so additional packages might be required for a fully functional base system. To install extra software or group packages, append their names (space-separated) to the pacstrap command above or use pacman while chrooted into the new system. Specifically, consider installing: microcode updates for hardware bugs and security patches (amd-ucode or intel-ucode), user-space utilities for file systems (e.g., fsck), RAID/LVM management tools if applicable, specific firmware packages not included in linux-firmware (e.g., sof-firmware for onboard audio), networking software (network manager or standalone DHCP client, Wi-Fi authentication software, ModemManager for mobile broadband connections), a console text editor (nano) to edit configuration files from the command line, and documentation access packages (man-db, man-pages, texinfo). To reference available packages in the live system, check pkglist.x86_64.txt. Next, configure the system by generating an fstab file using genfstab: # genfstab -U /mnt >> /mnt/etc/fstab Review the resulting fstab file and edit it if errors occur. Change root into the new system with arch-chroot: # arch-chroot /mnt Set the time zone with a symbolic link to /etc/localtime, then generate /etc/adjtime using hwclock(8): # hwclock --systohc This command assumes the hardware clock is set to UTC. For precise timekeeping and minimizing clock drift, use Network Time Protocol (NTP) clients like systemd-timesyncd for time synchronization. Edit locale settings by uncommenting required locales in /etc/locale.gen and running locale-gen. Create a locale.conf file with the following format: /etc/locale.conf LANG=en_US.UTF-8 If you've set your console keyboard layout, ensure it's persistent Post-installation steps to complete the network configuration for a newly installed environment include: Creating the hostname file at /etc/hostname and configuring suitable network management software. Enabling its systemd unit to start at boot. Note: A new initramfs is usually not required, as mkinitcpio was run during installation. However, if LVM, system encryption, or RAID are enabled, mkinitcpio.conf(5) needs to be modified and the initramfs image recreated. Set the root password with # passwd. Choose and install a Linux-capable boot loader. Exit the chroot environment by typing exit or pressing Ctrl+d. Optionally, manually unmount all partitions with umount -R /mnt before restarting. Final steps include: Removing the installation medium Logging into the new system with the root account

Arch linux installation guide 2024. Arch linux installation guide wifi. Arch linux installation guide script. Arch linux installation guide virtualbox. Arch linux installation guide vm. Arch linux installation guide. Arch linux installation guide 2023. Arch linux installation guide reddit. Arch linux installation guide vmware. Arch linux installation guide pdf. Arch linux installation guide dual boot. Arch linux installation guide kde. Arch linux installation guide github. Arch linux installation guide for beginners. Arch linux 32 installation guide.