

Forum:[Hack Forums](#)

Thread: [\[GUIDE \] Arch Linux installation guide](#)

Irritable - November 11, 2021, 3:52 am

Arch Linux installation guide

The intended audience(s) for this guide are as follows:

Average users (social media website usage, e-mails, office or college work, etc)

Curiosity consumed individuals

Computer enthusiasts

Individuals who believe in the pursuit of: Free information, software and privacy

Computer programmers

Linux users who desire to garnish more from their OS

Approximate results at the end of the installation (LXDE DE, LXDM greeter and OpenBox WM):

Harddrive consumption (usually 'high priced real-estate value' for very small medium sources (such as a high end SSD))/dev/sda1 (EFI): 512M

/dev/sda2 (root): ~3.8G

/dev/sda3 (home): ~99M

TOTAL HARDDRIVE USAGE: ~3.9G

RAM USAGE: 175M (excluding cached RAM; 350M if caches are allowed); ~3G of RAM free for use with anything

CPU USAGE: 1%

TOTAL BOOT TIME (cold-boot to desktop idle): 20.200s (a very old 5k RPM HDD)

How this guide is formatted:

Sections - "Sections" are a quick reference to jump around with. The title summarizes the procedure performed.

Notes - "Notes" contain miscellaneous information that I deemed essentially important.

Options - "Options" behave as a red-black tree; Linux typically has far more than a single venue to approaching a solution to any given problem.

Steps - "Steps" explain how to solve a given section in a chronological order; steps may sometimes be broken up into sub-components (EG: "cfdisk"s usage may not come naturally for all users).

Examples - "Examples" summarize what command(s) might be used (as your system and needs might vary: You cannot rely on them without reading the full text of the section).

Commands used - "Commands used" is merely a highly cohesive conglomerate of all aforementioned examples.

Section overview:Downloading the Arch Linux live ISO

Formatting a USB

Installing the image file (ISO) to the USB

Booting the live media (USB drive; Arch Linux live ISO)

Securely wiping the harddisk

Selecting a disk label type and creating partitions

Formatting the partition's filesystems (FAT32, ext4, swap)

Mounting the filesystems

Learning a bit about self-help with electronic manuals (the man pages)

Connecting to the internet (to install programs and the most recent version of the Linux kernel, drivers, etc)

Installing the base operating system (Arch Linux "base," "base-devel," "linux" and "linux-firmware")

Creating the fstab file
Defragment the filesystems
Chrooting to the system (root partition)
Setting and generating a locale (default language used system-wide)
Setting a time-zone
Setting the clock
Creating a hostname (machine name for local networks)
Creating a hosts file
Creating a password for the "root" user
Installing packages (programs)
Enabling systemd units
Installing the GRUB bootloader
Updating the GRUB configuration file
Installing an EFI stub for the motherboard
Installing an EFI start-up script (read by the motherboard after power-up)
Finishing up before rebooting
Adding new users and passwords
Enabling sudo privileges for users
Switching the logged in account
Creating an xinit script to automate starting the X-window system

1. Downloading the Arch Linux live ISO

2. Formatting a USB

NOTE: From a Linux distro -- Using the terminal

Step 1: Enumerate the physical devices and find the USB (/dev/sdX) using: `fdisk -l`; A description of the device(s) are shown.

Example: "Disk /dev/sdb: 7.33 GiB, 7864320000 bytes, 15360000 sectors" -- My USB device: /dev/sdb

Step 2: Set all data to 0x00 on the device (/dev/sdb in my example) using: `dd`; This will remove all partition tables, labels and data on the device.

Example: "`dd /dev/sdb if=/dev/zero status=progress`"

Commands used:

1. `fdisk -l`
2. `dd of=/dev/sdb if=/dev/zero status=progress`

3. Installing the image file (ISO) to the USB

NOTE: From a Linux distro -- Using the terminal

Step 1: Copy the Arch Linux ISO image to the USB using: `dd`

Example: "`dd /dev/sdb if=~/.Downloads/ArchLinuxLive.iso of=/dev/sdb status=progress`"

Commands used:

1. `dd of=/dev/sdb if=~/.Downloads/ArchLinuxLive.iso of=/dev/sdb status=progress`

4. Booting the live media (USB drive; Arch Linux live ISO)

NOTE: Reference your motherboard's manual for accessing it's firmware (BIOS or UEFI)

Step 1: Ensure the external USB ports / peripherals are accessible (security feature)

Step 2: Ensure CSM is enabled and secure-boot is disabled

Step 3: Save changes and power down the machine

Step 4: Insert the USB into a usable USB-port

Step 5: Power on the machine and forcibly select the USB to be booted to

Step 6: Select "Arch Linux Live" (automatically selected and activated after 5 seconds)

Commands used:

1. NONE

5. Securely wiping the hddisk

NOTE: From the Arch Linux live USB: We can fully access all hardware safely as a 'slave' device

Step 1: Enumerate the physical devices and find the harddrive (/dev/sdX) using: `fdisk -l`; A description of the device(s) are shown.

Example: "Disk /dev/sda: 931.53 GiB, 1000204886016 bytes, 1953525168 sectors" -- My harddrive device: /dev/sda

Step 2: Set all data to 0x00 on the device (/dev/sda in my example) using: `dd`; This will remove all partition tables, labels and data on the device.

Example: "`dd /dev/sda if=/dev/zero status=progress`"

Step 3: Manually synchronize everything to the disk using: `sync`

Step 4: Verify changes to the devices using: `lsblk`

Commands used:

1. `fdisk -l`

2. `dd of=/dev/sda if=/dev/zero status=progress`

3. `sync`

4. `lsblk`

6. Selecting a disk label type and creating partitions

Step 1: Create a new EFI partition type under an gpt partition label using: `cgdisk`

Example: "`cgdisk /dev/sda`"[/B]

Step 1.1: Select label type: gpt

Step 1.2: Select "New"

Step 1.3: Backspace the previous size and enter "512M" and press "enter"

Step 1.5: Select "Type"

Step 1.6: Select "EFI System"

Step 2: Create a new Linux root partition (where the Linux OS system, shared libraries and others are kept)

Step 2.1: Select "New"

Step 2.2: Backspace the previous size and enter "50G" (preferably anything over 20G) and press "enter"

Step 2.3: Select "Type"

Step 2.4: Select "Linux root (x86-64)"

Step 3: Create a new Linux home partition (where your user(s) personal files, settings, configurations, etc. are kept)

Step 3.1: Select "New"

Step 3.2: Fill the entire disk except for 2G (example: "400G" is shown; backspace and replace with: "398G") and press "enter"

Step 3.3: Select "Type"

Step 3.4: Select "Linux home"

Step 4: Create a new Linux swap partition (used for: "Hibernation" and misc. out-of-memory scenarios)

Step 4.1: Select "New"

Step 4.2: Verify the "2G" size and press "enter"

Step 4.3: Select "Type"

Step 4.4: Select "Linux swap"

Commands used:

1. `cfdisk /dev/sda`

7. Formatting the partition's filesystems (FAT32, ext4, swap)

Step 1: Verify the changes to the disk using: `lsblk`

Step 2: Format the EFI partition with a FAT32 file-system using: `mkfs.fat`

Example: `"mkfs.fat -F32 /dev/sda1"`

Step 3: Format the root partition with a EXT4 file-system using: `mkfs.ext4`

Example: `"mkfs.ext4 /dev/sda2"`

Step 4: Format the home partition with a EXT4 file-system using: `mkfs.ext4`

Example: `"mkfs.ext4 /dev/sda3"`

Step 5: Create a swap partition (physical disk for RAM usage) using: `mkswap`

Example: `"mkswap /dev/sda4"`

Step 6: Enable the usage of the partition for swapping out memory using: `swapon`

Example: `"swapon /dev/sda4"`

Commands used:

1. `lsblk`

2. `mkfs.fat -F32 /dev/sda1`

3. `mkfs.ext4 /dev/sda2`

4. `mkfs.ext4 /dev/sda3`

5. `mkswap /dev/sda4`

6. `swapon /dev/sda4`

8. Mounting the filesystems

Step 1: Mount the root partition at the mount point of `"/mnt"` using: `mount`

Example: `"mount /dev/sda2 /mnt"`

Step 2: Create a folder named "home" under the `"/mnt"` directory using: `mkdir`

Example: `"mkdir /mnt/home"`

Step 3: Mount the home partition at the mount point of the newly created `"/mnt/home"` directory using: `mount`

Example: `"mount /dev/sda3 /mnt/home"`

Step 4: Verify the changes using: lsblk, lsblk -f and df -h

Example: "lsblk && lsblk -f && df -h" (the && operator chain-links commands)

Commands used:

1. mount /dev/sda2 /mnt
2. mkdir /mnt/home
3. lsblk
4. lsblk -f
5. df -h

9. Learning a bit about self-help with electronic manuals (the man pages)

Step 1: Clear all text from the console (to improve readability) using: clear

Example: "clear"

Step 2: Learn more about the "clear" command using: man (press 'q' to quit)

Example: "man clear"

Step 3: Learn more about all previous commands using: man command-name

Example: "man mount" or "man mkdir" or "man lsblk" or "man df" or "man mkfs" or "man cfdisk" or "man dd" or "man fdisk" or "man sync"

Commands used:

1. clear
2. man clear
3. man ... insert any command to read more on it

10. Connecting to the internet (to install programs and the most recent version of the Linux kernel, drivers, etc)

NOTE: Invoking "ip link" or "iw dev" should reveal viable wifi ethernet ("link/ether") interfaces

Option 1: netctl

Option 1.1: Manual file configuration (using: cp, nano and netctl)

Step 1: For the sole purpose of simplification: Copy an example file from: "
/etc/netctl/examples/wireless-wpa" to "/etc/netctl/MyWifiSSID" using: cp

Example: "cp /etc/netctl/examples/wireless-wpa /etc/netctl/Home_Wifi-Router"

Step 2: Modify this newly created file using: nano

Example: "nano /etc/netctl/Home_Wifi-Router"

Step 2.1: Configure the interface to reflect the desired device to be used for wifi connections

Example: "Interface: wlp3s0"

Step 2.2: Configure the wifi key (password) if applicable

Example: "Key: SecretPassword12345"

Step 2.3: Configure the wifi's ESSID (broadcasted name)

Example: "ESSID: SecretWifiRouter-2G"

Step 2.4: Uncomment the "hidden" switch if the router does not broadcast it's name; otherwise: It assumes the name is broadcasted

Example: "hidden" (my wifi-router does not broadcast it's SSID (wifi-name); it is hidden)

Step 2.5: Configure the wifi connection's priority (useful if you have a laptop and/or multiple access-points (APs); routers)

Example: "#priority: 10" (priority is not factored in; as it is commented out as denoted by the "#" character)

Step 2.6: Save the changes to the netctl wifi-connection configuration file and exit

Step 3: Start the Home_Wifi-Router wifi-connection by using: netctl

Example: "netctl start Home_Wifi-Router"

Option 2: Automated file configuration (generated with: wifi-menu)

Option 2.1: wifi-menu

Step 1: Connect to the internet using: wifi-menu

Example: "wifi-menu"

Step 1.1: Select the SSID (the network's name as advertised by the router) using the directional arrow keys (pressing "enter" to select it)

Step 1.2: Enter the password for the network connection (if prompted)

Step 1.3: Start the wifi-connection by using: netctl (it should automatically connect and generate the netctl configuration file and stored at: "/etc/netctl")

NOTE: You will need to install the following packages using pacman to use "netctl" and "wifi-menu" outside of the Arch Linux Live OS: "netctl", "dialog" and "wpa_supplicant"

Option 3: iw

Step 1: Scan for available wifi-networks using: iw

Example: "sudo iw dev wlp3s0 scan"

Example #2: "sudo iw dev wlp3s0 scan | grep SSID" (if you wish to quickly filter out only the SSID names)

Step 2: Connect to the wifi-router using: iw

Example: "sudo iw dev wlp3s0 connect -w Home_Wifi-Router" ("wlp3s0" is my interface (wifi-card's) name)

NOTE: This will set up a static IP address and routing address (ARP (address resolution protocol))

Option 4: ip

Step 1: Set the link interface as "up" using: ip

Example: "ip link set wlp3s0 up"

Step 2: Add an IP address to the interface using: ip

Example: "ip address add 192.168.1.99/24 broadcast + dev wlp3s0"

Step 3: Add a route to the interface using: ip

Example: "ip route add 192.168.1.99 via 192.168.1.1 dev wlp3s0"

Commands used:

1. ip link
2. iw dev
3. cp /etc/netctl/examples/wireless-wpa /etc/netctl/Home_Wifi-Router
4. nano /etc/netctl/Home_Wifi-Router
5. netctl start Home_Wifi-Router
6. wifi-menu

7. `sudo iw dev wlp3s0 scan`
8. `sudo iw dev wlp3s0 scan | grep SSID`
9. `sudo iw dev wlp3s0 connect -w Home_Wifi-Router`
10. `sudo ip link set wlp3s0 up`
11. `sudo ip address add 192.168.1.99/24 broadcast + dev wlp3s0`
12. `sudo ip route add 192.168.1.99 via 192.168.1.1 dev wlp3s0`

11. Installing the base operating system (Arch Linux "base," "base-devel," "linux" and "linux-firmware")

NOTE: "pacstrap" is a special Arch Linux live program that is not found within the non-live version of Arch Linux

Step 1: Install all packages from the "base," "base-devel," "linux," "linux-firmware," package-groups to the "/mnt" directory using: pacstrap

Example: `"pacstrap -i /mnt base base-devel linux linux-firmware"`

Commands used:

1. `pacstrap -i /mnt base base-devel linux linux-firmware`

12. Creating the fstab file

NOTE: "genfstab" is a special Arch Linux live program that is not found within the non-live version of Arch Linux

Step 1: Generate an fstab file at the path of: `"/mnt/etc/fstab"` using: genfstab

Step 2: Verify the file was created by using: cat

Example: `"cat /mnt/etc/fstab"`

Commands used:

1. `genfstab -U -p /mnt >> /mnt/etc/fstab`
2. `cat /mnt/etc/fstab`

13. Defragment the filesystems

Step 1: Defragment all partitions on the device using: e4defrag

Example: `"e4defrag -v /dev/sda*"` (the asterisk (`*`) acts as a wildcard; the observable behavior then includes: `e4defrag /dev/sda1`, `e4defrag /dev/sda2`, `e4defrag /dev/sda3`, `e4defrag /dev/sda...`)

Step 2: Ensure all changes are written out to the disk using: sync

Example: `"sync"`

Commands used:

1. `e4defrag -v /dev/sda`
2. `sync`

14. Chrooting to the system (root partition)

Step 1: chroot into the directory of: `/mnt` using: arch-chroot

Example: `"arch-chroot /mnt /bin/bash"`

Commands used:

1. `arch-chroot /mnt /bin/bash`

15. Setting and generating a locale (default language used system-wide)

NOTE: These options relate to CLI text editors that are used to edit the `/etc/locale.gen` configuration file

Step 1: Configure the locale settings by modifying the `/etc/locale.gen` file using: nano

Example: `"nano /etc/locale.gen"`

Step 1.1: Search for your language using the hotkey: CTRL + W

Example: `"CTRL + W"` then type `"#en_US"`

Step 1.2: Remove the comment: `'#'` using the directional arrow-keys and the backspace key (the line should now read: `"en_US"` from the previous `"#en_US"` text)

Step 1.3: Save changes to the file using the hotkey: `"CTRL + X"` then press `"y"` for `"yes"`

NOTE: `"locale-gen"` is a special Arch Linux live program that is not found within the non-live version of Arch Linux

Step 1: Generate all locale files using: `locale-gen`

Example: `"locale-gen"`

Commands used:

1. `nano /etc/locale.gen`
2. `locale-gen`

16. Setting a time-zone

NOTE: Finding listed time-zones that are available

Option 1: Enumerate the `/usr/share/zoneinfo` directory using: `"ls"`

Example: `"ls /usr/share/zoneinfo"`

Option 2: List all time-zones using: `"timedatectl"`

Example: `"timedatectl list-timezones"`

Option 3: List all time-zones using: `"timedatectl"` and filter a desired category with `"grep"`

Example: `"timedatectl list-timezones | grep America"` (the `"|"` operator applies the command to the formerly used command)

Step 1: Create a symbolic link from: `/usr/share/zoneinfo/XXX` to `/etc/localtime` using: `ln`

Example: `"ln -sf /usr/share/zoneinfo/America/Chicago /etc/localtime"`

Commands used:

1. `ls /usr/share/zoneinfo`
2. `timedatectl list-timezones`
3. `timedatectl list-timezones | grep America`
4. `ln -sf /usr/share/zoneinfo/America/Chicago /etc/localtime`

17. Setting the clock

Option 1: Set the system-clock (operating system level) from the hardware clock (motherboard level)

Step 1: Read the `"--hctosys"` flag for `"hwclock"` from the man-page using: `man hwclock`

Step 1.1: Set the system-clock from the hardware clock using: `hwclock -s`

Example: `"hwclock -s"`

Option 2: Set the hardware clock (motherboard level) from the system-clock (operating system level)

Step 1: Read the "--systohc" flag for "hwclock" from the man-page using: `man hwclock`

Step 1.1: Set the hardware clock from the system-clock using: `hwclock -w`

Example: `"hwclock -w"`

Commands used:

1. `man hwclock`
2. `hwclock -s`
3. `hwclock -w`

18. Creating a hostname (machine name for local networks)

NOTE: The hostname shows up on the terminal as: `[Username@Hostname ~]$`

Step 1: Write the desired hostname to: `"/etc/hostname"` using: `echo`

Example: `"echo MyHostName > /etc/hostname"` (the ">" command writes the data to the harddrive ("hostname" is the file; "MyHostName" is written to that file))

Commands used:

1. `echo MyHostName > /etc/hostname`

19. Creating a hosts file

Step 1: Read the "hostname" page from the man-page using: `man`

Example: `"man hostname"`

Step 2: Associate your hostname with `"127.0.1.1 localhost.localdomain"` by using: `echo`

Example: `"echo "127.0.1.1 localhost.localdomain MyHostName" > /etc/hosts"`

Step 3: Amuse yourself with learning by using: `hostname` and it's assorted flags

Example: `"hostname -a && hostname -d && hostname -f && hostname -s && hostname -y && hostname -i"`

Commands used:

1. `man hostname`
2. `echo "127.0.1.1 localhost.localdomain MyHostName" > /etc/hosts`
3. `hostname -a && hostname -d && hostname -f && hostname -s && hostname -y && hostname -i`

20. Creating a password for the "root" user

Step 1: Read the "passwd" page from the man-page using: `man`

Example: `"man passwd"`

Step 2: Set the password for the user "root" using: `passwd`

Example: `"passwd"`

Step 2.1: Type a password for the user "root", then press enter

Step 2.2: Confirm the password for the user "root" by entering it a second time, then press enter

Commands used:

1. `man passwd`

2. passwd

21. Installing packages (programs)

NOTE: You can install all packages with a single: "pacman -S"; package names are space separated

Step 1: Read the "pacman" page from the man-page using: man

Example: "man pacman"

NOTE: Installing the system-related packages

Step 2: Install the following packages: "grub efibootmgr", using: pacman

Example: "pacman -S grub efibootmgr"

NOTE: Installing the X-window related packages

Step 3: Install the following packages: "xorg xorg-xinit xorg-server", using: pacman

Example: "pacman -S xorg xorg-xinit xorg-server"

NOTE: Installing the other X-window related packages (desktop environment and GUI-based) login greeter)

Step 4: Install the following packages: "lxde lxdm", using: pacman

Example: "pacman -S lxde lxdm"

NOTE: Installing the audio related packages

Step 5: Install the following packages: "pulseaudio pulseaudio-alsa", using: pacman

Example: "pacman -S pulseaudio pulseaudio-alsa"

NOTE: Installing the network related packages

Step 6: Install the following packages: "networkmanager network-manager-applet", using: pacman

Example: "pacman -S networkmanager network-manager-applet"

NOTE: Installing GUI notification (eye-candy) related packages

Step 7: Install the following packages: "xfce4-notifyd", using: pacman

Example: "pacman -S xfce4-notifyd"

NOTE: Installing a web-browser, a media player, a CLI-based text editor and a GUI-based text editor set of packages

Step 8: Install the following packages: "firefox vlc nano leafpad", using: pacman

Example: "pacman -S firefox vlc nano leafpad"

NOTE: Installing a power-manager, a screen-shot, a GUI-based disk management and a general-use unnecessary clutter removal package

Step 9: Install the following packages: "xfce4-power-manager gnome-screenshot gparted bleachbit", using: pacman

Example: "pacman -S xfce4-power-manager gnome-screenshot gparted bleachbit"

NOTE: Installing a common chat-program (Discord) and multi-client (Pidgin) package

Step 10: Install the following packages: "discord pidgin", using: pacman

Example: "pacman -S discord pidgin"

Commands used:

1. man pacman

2. pacman -S nano grub efibootmgr networkmanager network-manager-applet pulseaudio

pulseaudio-alsa xorg xorg-xinit xorg-server lxde lxdm xfce4-notifyd xfce4-power-manager vlc discord
pidgin firefox leafpad gnome-screenshot gparted bleachbit

22. Enabling systemd units

Step 1: Read the "systemctl" page from the man-page using: man

Example: "man systemctl"

Step 2: Read the "sudo" page from the man-page using: man

Example: "man sudo"

Step 3: Enable the "NetworkManager" system unit using: systemctl

Example: "systemctl enable NetworkManager"

Step 4: Enable the "lxdm" system unit using: systemctl

Example: "systemctl enable lxdm"

Commands used:

1. man systemctl
2. man sudo
3. sudo systemctl enable NetworkManager
4. sudo systemctl enable lxdm

23. Installing the GRUB bootloader

NOTE: This is for a UEFI based motherboard installation (x86-64 or "amd64" architecture is implied;
Arch Linux no longer supports the x86 arch)

Step 1: Create a folder named "/boot/efi" under the "/" directory using: mkdir

Example: "mkdir /boot/efi"

Step 2: Mount the EFI partition at the mount point of the newly created "/boot/efi" directory using:
mount (invoke lsblk if you forget what the partition name is)

Example: "mount /dev/sda1 /boot/efi"

Step 3: Verify changes to the devices using: lsblk

Example: "lsblk -f"

Step 4: Read the "grub-install" page from the man-page using: man

Example: "man grub-install"

Step 5: Install the GRUB-bootloader, using: grub-install

Example: "grub-install --target=x86_64-efi --bootloader-id=GRUB --efi-directory=/boot/efi --recheck"

Commands used:

1. mkdir /boot/efi
2. mount /dev/sda1 /boot/efi
3. lsblk -f
4. man grub-install
5. grub-install --target=x86_64-efi --bootloader-id=GRUB --efi-directory=/boot/efi --recheck

24. Updating the GRUB configuration file

Step 1: Read the "grub-mkconfig" page from the man-page using: man
Example: "man grub-mkconfig"

Step 2: Generate a new GRUB configuration file named "grub.cfg" under the "/boot/grub" directory using: grub-mkconfig
Example: "grub-mkconfig -o /boot/grub/grub.cfg"

Commands used:

1. man grub-mkconfig
2. grub-mkconfig -o /boot/grub/grub.cfg

25. Installing an EFI stub for the motherboard

Step 1: Create a directory named "/boot/efi/EFI/" using: mkdir
Example: "mkdir /boot/efi/EFI/BOOT"

Step 2: Read the "cp" page from the man-page using: man
Example: "man cp"

Step 3: Copy the GRUB bootloader ("grubx64.efi") from the "/boot/efi/EFI/GRUB" directory to the "/boot/efi/EFI/" directory, using: cp
Example: "cp /boot/efi/EFI/GRUB/grubx64.efi /boot/efi/EFI/BOOT/BOOTx64.EFI"

Commands used:

1. mkdir /boot/efi/EFI/BOOT
2. man cp
3. cp /boot/efi/EFI/GRUB/grubx64.efi /boot/efi/EFI/BOOT/BOOTx64.EFI

26. Installing an EFI start-up script (read by the motherboard after power-up)

Step 1: Create a file named "startup.nsh" under the "/boot/efi" directory using: nano
Example: "nano /boot/efi/startup.nsh"

Step 2: Write the line of text for the NSH script as follows: "bcfg boot add 1
fs0:\EFI\GRUB\grubx64.efi "Your custom label here""

Step 2.1: Save changes to the file using the hotkey: "CTRL + X" then press "y" for "yes"

Step 2.2: Write changes to the harddrive (create the file) by pressing enter (confirm the filename first)

Commands used:

1. nano /boot/efi/startup.nsh

27. Finishing up before rebooting

Step 1: Exit out of the "chroot" using: exit
Example: "exit"

Step 2: Force a flush (writes all buffered, cached or unwritten data in memory) of all dirty data to the harddrive using: sync
Example: "sync"

Step 3: Defragment all partitions on the device using: e4defrag
Example: "e4defrag -v /dev/sda*"

Step 4: Read the "umount" page from the man-page using: man

Example: "man umount"

Step 5: Unmount all mounted partitions using: umount

Example: "umount -R /mnt"

Step 6: Verify changes to the devices using: lsblk

Example: "lsblk"

Step 7: Read the "e2fsck" page from the man-page using: man

Example: "man e2fsck"

Step 8: Optimize the root and home partitions using: e2fsck

Example: "e2fsck -Dfv /dev/sda2 && e2fsck -Dfv /dev/sda3"

Step 9: Read the "shutdown" page from the man-page using: man

Example: "man shutdown"

Step 10: Shutdown quickly by using: shutdown

Example: "shutdown now"

Step 11: Remove the USB after the machine powers off and then proceed to power on the machine

Commands used:

1. exit
2. sync
3. e4defrag -v /dev/sda*
4. man umount
5. umount -R /mnt
6. lsblk
7. man e2fsck
8. e2fsck -Dfv /dev/sda2 && e2fsck -Dfv /dev/sda3
9. man shutdown
10. shutdown

28. Adding new users and passwords

Step 1: Log into the "root" user by typing "root" for the "login" prompt and press enter

Step 1.1: Type in the password you created earlier and press enter

Step 2: Read the "useradd" page from the man-page using: man

Example: "man useradd"

Step 3: Add a new user login using: useradd

Example: "useradd -m -g users -G wheel -s /bin/bash MyUniqueUsername"

Step 4: Set the password for the newly created user "MyUniqueUsername" using: passwd

Example: "passwd MyUniqueUsername"

Commands used:

1. man useradd
2. useradd -m -g users -G wheel -s /bin/bash MyUniqueUsername
3. passwd MyUniqueUsername

29. Enabling sudo privileges for users

Step 1: Open the special "sudoers" file with the command: "EDITOR=nano visudo"

Step 2: Search for "# %wheel ALL=(ALL) ALL" and uncomment it (remove the "#" character)[/B]"

Step 3: Save the changes to the file using the hotkeys and exit using the hotkeys[/B]

Commands used:

1. NONE

30. Switching the logged in account

Step 1: Exit out of the "root" login using: exit

Example: "exit"

Step 2: Log in to your newly created account by typing it and pressing enter

Example: "MyUniqueUsername"

Step 2.1: Type in your password and press enter

Example: "MyUniquePassword"

Commands used:

1. exit

31. Creating an xinit script to automate starting the X-window system

Step 1: Create a new file named ".xinitrc" under the "~/" directory using: nano

Example: "nano ~/.xinitrc"

Step 2: Write the line of text for the xinit script as follows: "exec startlxde"

Step 2.1: Save changes to the file using the hotkey: "CTRL + X" then press "y" for "yes"

Step 2.2: Write changes to the harddrive (create the file) by pressing enter (confirm the filename first)

Step 3: Reboot by using: "reboot"

Example: "reboot"

Commands used:

1. nano ~/.xinitrc

2. reboot

guardian - November 13, 2021, 12:27 am

I was actually going to write a tutorial on how to install Arch Linux today, as I am about to make a fresh installation. Great to see that one has been posted not even two days ago. Will find something else to make a tutorial about then. :) Haven't read through it, but looks good, keep up the HQ content man!

Irritable - November 13, 2021, 11:38 pm

(Nov 13, 2021 01:27 AM)guardian Wrote: [CITE](#) I was actually going to write a tutorial on how to install Arch Linux today, as I am about to make a fresh installation. Great to see that one has been posted not even two days ago. Will find something else to make a tutorial about then. :) Haven't read through it, but looks good, keep up the HQ content man!

Thank you.

I wrote this quite a few years ago.

Another poster pointed out that the ArchWiki has a 'guided installer' written in Python finally (~2020).

The URL: <https://wiki.archlinux.org/title/Archinstall>

guardian - November 14, 2021, 3:31 am

(Nov 14, 2021 12:38 AM)Irritable Wrote: [\[CITE\]](#) Thank you.

I wrote this quite a few years ago.

Another poster pointed out that the ArchWiki has a 'guided installer' written in Python finally (~2020).

The URL: <https://wiki.archlinux.org/title/Archinstall>

Didn't even know they had that.

Got my fresh Arch install done yesterday, looks super sick. Playing around with themes rn.

Betty Blue - November 14, 2021, 6:14 am

(Nov 14, 2021 04:31 AM)guardian Wrote: [\[CITE\]](#) Didn't even know they had that.

Got my fresh Arch install done yesterday, looks super sick. Playing around with themes rn. I forget the name but if you're using a WM, people around /r/unixporn have an automatic theme switcher for WM's. I don't use it, but it looks really nice.

guardian - November 14, 2021, 7:03 am

(Nov 14, 2021 07:14 AM)Deadhead Wrote: [\[CITE\]](#) I forget the name but if you're using a WM, people around /r/unixporn have an automatic theme switcher for WM's. I don't use it, but it looks really nice.

Thanks mate! You mean that the theme switches over time? Or on command?

Cannot Compute - November 14, 2021, 7:10 am

(Nov 14, 2021 08:03 AM)guardian Wrote: [\[CITE\]](#) Thanks mate! You mean that the theme switches over time? Or on command?

There's pywal which generates color schemes based on your background. May be something you're looking for.

guardian - November 14, 2021, 7:18 am

(Nov 14, 2021 08:10 AM)Cannot Compute Wrote: [\[CITE\]](#) There's pywal which generates color schemes based on your background. May be something you're looking for.

Sounds cool! Thanks man, will look into it.

Betty Blue - November 14, 2021, 10:35 am

(Nov 14, 2021 08:03 AM)guardian Wrote: [\[CITE\]](#) Thanks mate! You mean that the theme switches over time? Or on command? On command, widget you can scroll through graphically, really however you want it probably.

Irritable - November 19, 2021, 8:48 pm

(Nov 14, 2021 04:31 AM)guardian Wrote: [\[CITE\]](#) Didn't even know they had that.

Got my fresh Arch install done yesterday, looks super sick. Playing around with themes rn.

Arch Linux is a very mean, lean and lightweight operating system that's usually kept close to the bleeding edge.

Please note: WMs and DEs are not restricted to specific distros or Linux kernels.

However! Arch Linux is fairly naked, which means instead of removing clutter and cruft: You only add what you want and remove the miniscule amount that it ships with at stock (OEM, if you will).

guardian - November 19, 2021, 10:14 pm

(Nov 19, 2021 09:48 PM)Irritable Wrote: [\[CITE\]](#) Arch Linux is a very mean, lean and lightweight operating system that's usually kept close to the bleeding edge.

Yeah man, that's why I like it haha

SIM-353284 - November 21, 2021, 1:17 am

Great tutorial really learnt alot thanks.

Irritable - November 21, 2021, 5:50 am

(Nov 21, 2021 02:17 AM)SIM Wrote: [\[CITE\]](#) Great tutorial really learnt alot thanks.

You're welcome!

This was just a guide written before the Python-based installation guide was created.

This isn't a tutorial or anything informative.