



Arch Linux | Installation

Arch Linux Installation über die Textkonsole

Eine ausführliche Anleitung für Einsteiger und Fortgeschrittene

Stand: **08. Februar 2023**



Anleitung als PDF hier herunterladen:

[Arch-Linux.Installation.pdf](#)

Übersicht

- a. Einleitung
- b. Voraussetzungen und Vorgaben
- c. Vorbereiten eines Live-ISO Bootsticks
- d. Arch Linux Live-Stick booten und erste Vorbereitungen
- e. Partitionierung der Festplatte
- f. Dateisysteme anlegen
- g. Dateisysteme einhängen
- h. Installation des Basissystems | fstab | arch-chroot
- i. Benutzer einrichten

- [j. Sprach-Einstellungen](#)
 - [k. Der Paketmanager](#)
 - [l. nützliche Programme, Tools und Dienste](#)
 - [m. Installation von X, Grafiktreiber und Xfce-Desktop](#)
 - [n. Dienste aktivieren](#)
 - [o. Grub Bootloader](#)
 - [p. Neustart](#)
 - [q. Ersteinrichtung](#)
-

a. Einleitung

Die Installation von Arch Linux ist kein Hexenwerk. Diese Distribution besitzt keinen graphischen Installer und die Basisausstattung von Arch Linux ist nur einige 100 MB groß. Als sehr hilfreich erweist sich die Seite:

<https://www.archlinux.de/>

Mit Hilfe der Video-Reihe [Total Arch Linux](#) von unicks.eu und der ausführlichen deutschen Anleitung zur Arch Linux Installation https://wiki.archlinux.de/title/Anleitung_für_Einsteiger ist diese angepaßte Schritt-für-Schritt Anleitung entstanden.

Eine sehr ausführliche, englische Video-Anleitung zur Einrichtung von Arch Linux mit verschiedenen Desktop-Umgebungen findet man hier:

[Arch Linux: Full Installation Guide - A complete tutorial/walkthrough in one video!](#)

Es gibt im Netz zwar eine Reihe an ausführlichen Anleitungen, um Arch Linux zu installieren, doch oftmals sind sie leider nicht aus "einem Guß". So ist dann dieser Anfänger-Guide zustande gekommen, den ich auch im eigenen Interesse aktuell zu halten versuche. Verbesserungsvorschläge nehme ich natürlich gerne entgegen.

b. Voraussetzungen und Vorgaben

Für einen aktuellen 64bit Desktop PC mit UEFI habe ich mir folgende Hard- und Software-Vorgaben gesetzt:

- Verwendung einer aktuellen (!) 64bit Arch-Linux Live-ISO von einem USB-Stick
- Installation im UEFI-Modus auf einem **Desktop PC**
- Die Partitionen auf der Festplatte sind/werden mit GPT erstellt.
- Die Verwendung einer 2 TByte M.2 SSD, auf der Arch Linux als alleiniges Betriebssystem installiert werden soll.
- Der Arbeitsspeicher sollte mindestens 2 GByte, besser aber 8 GByte oder mehr besitzen.
- Secure Boot im UEFI Bios **de**-aktiviert
- UEFI Boot aktiviert, der BIOS Modus vom Mainboard ist also **de**-aktiviert!
- Der GRUB Boot-Loader wird per UEFI eingerichtet.
- Verwendung des Partitionswerkzeugs **gdisk**
- Der Rechner wird über ein LAN-Kabel (automatisch) mit dem Netzwerk verbunden
- keine Verschlüsselung

c. Vorbereiten eines Live-ISO Bootsticks

Eine aktuelles Arch Linux als ISO Datei zieht man sich hier:

<https://www.archlinux.de/download/>

Das ISO-Abbild wird geprüft mit, wobei * für die aktuelle Datumsversion steht.

```
md5sum archlinux-*-x86_64.iso
```

Das heruntergeladene ISO-Image wird über die Kommandozeile auf einen ausreichend großen USB-Stick kopiert. Eine genauere Anleitung dazu findet man unter: [Linux Bootstick erstellen](#)

Der Aufruf **lsblk** im Terminal zeigt den zu ermittelnden (!) Laufwerksbuchstaben x des USB-Sticks /dev/sdx und wird zuerst mit **umount** ausgehängt.

```
sudo umount /dev/sdx
```

Mit dem **dd** Befehl wird dann das Arch Live ISO auf den Stick kopiert:

```
sudo dd if=/path/to/archlinux-*-x86_64.iso of=/dev/sdx bs=4M status=progress oflag=ds
```

Unter Linux kann man auch das Festplattenverwaltungswerkzeug Gnome-Disk-Utility ("Laufwerke") verwenden, welches in vielen Distributionen bereits vorhanden ist bzw. sich leicht über den Paketmanager installieren läßt. In der oberen Leiste findet sich der Eintrag "Laufwerksabbild wiederherstellen ...", über den sich die ISO Datei leicht auf den USB-Stick aufspielen läßt.

d. Arch Linux Live-Stick booten und erste Vorbereitungen

Der soeben erstellte Arch Live Stick bootet mit einem Rechnerneustart in ein Boot-Auswahlmenü und zeigt als ersten Eintrag "Arch Linux install medium (x86_64, UEFI CD)". Mit dieser Auswahl wird der Live Stick in den UEFI-Modus gestartet. Am Ende des Bootvorganges wird folgende Zeile angezeigt, die besagt, dass man nun als root User angemeldet ist:

```
root@archiso ~#
```

Als nächstes stelle ich die Sprache bzw. das Tastaturlayout auf Deutsch um, mit korrekter Ausgabe der Umlaute. Verwende `z`, um `y` zu erhalten:

```
loadkeys de
loadkeys de-latin1
```

Bei einem großen Monitor (z.B. 34") wie ich ihn verwende, ist die Standard-Schriftausgabe mit 8 Pixeln doch recht klein und nicht allzugut lesbar. Das läßt sich zumindest für diese Session aber leicht ändern, indem ich mit folgendem Aufruf eine hinreichend große und gut lesbare Schrift auf dem Bildschirm zaubere:

```
setfont ter-132n
```

Gegebenenfalls noch die NUM-Taste auf dem Ziffernblock aktivieren, um die numerischen Tasten bedienen zu können.

Mit dem Befehl

```
clear
```

lösche ich gelegentlich den aktuellen Bildschirminhalt zwecks besserer Übersicht.

e. Partitionierung der Festplatte

Meine für diese Anleitung verwendete 2 TByte große M.2 Samsung SSD 970 EVO Plus wird gelöscht und soll anschließend in eine 512 MByte Boot- sowie in eine 20 GByte Swap-Partition aufgeteilt werden, die von der Größe her in etwa der des Arbeitsspeichers entspricht. Die dritte Partition wird für das Betriebssystem bereitgestellt, auf der Archlinux samt Homeverzeichnis installiert werden soll. Die einzelnen Partitionen sind dabei in `nvme0n1p1 .. p2 .. p3` unterteilt.

Mit `lsblk` verschafft man sich einen Überblick über vorhandene Partitionen und angeschlossene Festplatten. Hier muß man sehr darauf achten, die richtige Festplatte zu "erwischen", ansonsten droht Datenverlust! Nicht benötigte, externe Festplatten sollte man sowieso vorher abklemmen.

Zum Partionieren eines UEFI Systems benutze ich nun das mitgelieferte Partitionierungswerkzeug **gdisk**. Der Aufruf erfolgt in meinem Beispiel mit

```
gdisk /dev/nvme0n1
```

Mit Eingabe von "o" lösche ich alle evtl. vorhandenen Partitionen und mit Eingabe von "n" wird eine neue Partition angelegt. Die möglichen einzelnen Arbeitsschritte sehen in tabellarischer Form folgendermaßen aus:

BOOT-Partition mit <code>/dev/nvme0n1p1</code> - Kernel, Bootloader	Eingabe 1	Eingabe 2
neue Partition anlegen mit	n	ENTER

BOOT-Partition mit /dev/nvme0n1p1 - Kernel, Bootloader	Eingabe 1	Eingabe 2
Partition number		ENTER
First sector		ENTER
Last sector	+512M	ENTER
HEX code	ef00	ENTER

SWAP-Partition mit /dev/nvme0n1p2 - Auslagerungsspeicher	Eingabe 1	Eingabe 2
neue Partition anlegen mit	n	ENTER
Partition number		ENTER
First sector		ENTER
Last sector	+20G	ENTER
HEX code	8200	ENTER

ROOT-Partition mit /dev/nvme0n1p3 - Arch Linux Betriebssystem	Eingabe 1	Eingabe 2
neue Partition anlegen mit	n	ENTER
Partition number		ENTER
First sector		ENTER
Last sector		ENTER
HEX code	8300	ENTER

Wird bei "Last sector" kein Wert eingetragen, so wird der Rest des Festplattenspeichers an die aktuelle Partition `/dev/nvme0n1p3` vergeben. Die Eingabe von "p" zeigt nochmals einen Überblick. Mit Eingabe von "w" und der abschließenden Bestätigung mit "Y" werden die Änderungen in die Partitionstabelle geschrieben. Am Schluß erfolgt die Meldung: "OK, writing new GUID partition table (GPT) to `/dev/sdb`. The operation has completed successfully." lsblk zeigt einen abschließenden Überblick.

f. Dateisysteme anlegen

Mit **mkfs** (make file system) werden die Dateisysteme angelegt und formatiert.

Boot-Partition

```
mkfs.fat -F32 -n BOOT /dev/nvme0n1p1
```

Swap-Partition

```
mkswap -L SWAP /dev/nvme0n1p2
```

Root-Partition

```
mkfs.ext4 -L ROOT /dev/nvme0n1p3
```

g. Dateisysteme einhängen

Zuerst wird die System-Partition (ROOT) am Einhängepunkt `/mnt` eingehängt. Das Device (hier: Root-Partition) `/dev/nvme0n1p3` wird eingehängt und unter `/mnt` per Label `-L` verfügbar gemacht:

```
mount -L ROOT /mnt
```

Die Boot-Partition wird in einem /mnt Unterverzeichnis von ROOT verfügbar gemacht. Zuvor muß jedoch das boot-Verzeichnis angelegt werden:

```
mkdir /mnt/boot  
  
mount -L BOOT /mnt/boot/
```

Das Aktivieren bzw. Einschalten der Swap-Partition geht mit dem Aufruf:

```
swapon -L SWAP
```

Die Festplatte ist nun für die Arch Linux Installation vorbereitet. Möchte man noch eine weitere interne Festplattenpartition (z.B. sda1) dauerhaft in die fstab einhängen, so muß zuerst das gewünschte Verzeichnis erstellt und dann an passender Stelle eingehängt werden. Zuvor verschafft man sich einen Überblick per lsblk.

```
mkdir /mnt/interne_Festplatte  
  
mount /dev/sda1 /mnt/interne_Festplatte
```

h. Installation des Basissystems | fstab erzeugen | arch-chroot

Auf dem Live-System aktualisiert Reflector nach der Verbindung mit dem Internet die Spiegelliste für die zu installierenden Pakete in der Datei "/etc/pacman.d/mirrorlist". Diese Datei wird später per Pacstrap auf das neue System kopiert. Es lohnt sich also, diese Mirrorliste gleich zu Beginn nach eigenen Bedürfnissen anzupassen:

```
sudo nano /etc/pacman.d/mirrorlist
```

wobei STRG + K unerwünschte Zeilen löscht bzw. ausgeschnitten und ggf. an passender Stelle wieder eingefügt werden können. Speichern mit STRG + O und beenden mit STRG + X.

Als nächstes wird das Arch Linux Basissystem durch Aufruf von **pacstrap** installiert:

```
pacstrap /mnt base base-devel linux linux-firmware dhcpd nano dosfstools gptfdisk
```

Die Gruppe "base" lädt Pakete aus dem Repository für ein minimales System, "base-devel" lädt zusätzlich Extra-Tools, die es ermöglichen, weitere Software aus dem AUR zu installieren.

Die Pakete intel-ucode bzw. amd-ucode sorgen für Fehlerbereinigungen und Verbesserungen für die CPU, je nachdem, ob ein Intel- oder AMD-Prozessor verbaut ist:

```
pacstrap /mnt intel-ucode
```

oder

```
pacstrap /mnt amd-ucode
```

Die Konfigurationsdatei **fstab** (file system table) liegt im Verzeichnis `/etc/fstab` und enthält eine Liste aller zu mountender Dateisysteme. Alle angelegten Partitionen sind bereits mit einem Label L (siehe: f. Dateisysteme anlegen) versehen:

```
genfstab -Lp /mnt > /mnt/etc/fstab
```

Die Datei `fstab` wird mit allen eingebundenen, bereits gemounteten (!) Laufwerken erzeugt. Das Programm **genfstab** nimmt also nur die Geräte in die Datei `fstab` auf, die bereits gemountet sind. Auf diese Weise können auch nachträglich Festplatten oder andere Laufwerke an das System angebunden und anschließend in die Datei `fstab` aufgenommen werden.

Mit dem **cat** Aufruf läßt sich `fstab` nochmals anzeigen:

```
cat /mnt/etc/fstab
```

Mit diesem Aufruf wechselt man vom Live ISO in das neue System von /mnt

```
arch-chroot /mnt
```

i. Benutzer einrichten

Der **Rechnername** wird festgelegt (hier: kitty) mit

```
echo kitty > /etc/hostname
```

Setzen des Root-Passwortes mit

```
passwd
```

Einen Benutzer mit **useradd** hinzufügen. Der Benutzername darf nur Kleinbuchstaben und Sonderzeichen enthalten. Der Benutzername ist hier wieder mit "kitty" gewählt.

```
useradd -m -g users -s /bin/bash kitty
```

Passwortvergabe für den Benutzer "kitty" mit zweifacher Passwordeingabe:

```
passwd kitty
```

Benutzer kitty zur Gruppe wheel hinzufügen:

```
gpasswd -a kitty wheel
```

Damit der Benutzer Rootrechte erhält, muß eine Konfiguration verändert werden:

```
EDITOR=nano visudo
```

Bei folgenden Zeilen müssen die Kommentarzeichen # und das Leerzeichen entfernt werden:

```
# %wheel ALL=(ALL:ALL) ALL
```

Nach dem Abspeichern haben damit alle User der Gruppe wheel Administratorrechte, wenn sie sudo verwenden.

j. Sprach-Einstellungen

Die Systemsprache wird auf Deutsch eingestellt.

```
echo LANG=de_DE.UTF-8 > /etc/locale.conf
```

Die Konfigurationsdatei **locale.gen** mit dem nano Editor öffnen:

```
nano /etc/locale.gen
```

Das Kommentarzeichen "#" am Anfang folgender Zeilen entfernen und dann abspeichern mit STRG+O und ENTER und beenden mit STRG+X:

```
.#de_DE.UTF-8 UTF-8  
.#de_DE ISO-8859-1  
.#de_DE@euro ISO-8859-15  
.#en_US.UTF-8 UTF-8
```

Abschließend generieren mit:

```
locale-gen
```

Die Tastaturbelegung und Schriftart in **vconsole.conf** festlegen:

```
echo KEYMAP=de-latin1 > /etc/vconsole.conf  
echo FONT=lat9w-16 >> /etc/vconsole.conf
```

Die Zeitzone festlegen:

```
ln -sf /usr/share/zoneinfo/Europe/Berlin /etc/localtime
```

k. Der Paketmanager

Über die "pacman.conf" Datei läßt sich das Aussehen und Verhalten des Paketmanagers Pacman ändern. Mit Nano wird diese Datei geöffnet:

```
sudo nano /etc/pacman.conf
```

In den "Misc options" werden durch das Entfernen des #-Zeichen bei "Color" bestimmte Zeichen farbig dargestellt. Durch Einfügen in einer leeren Zeile **ILoveCandy** wird dem Fortschrittsbalken ein richtiger Pacman hinzugefügt!

Falls man für das 64bit Betriebssystem 32bit Bibliotheken benötigt, muss bei "multilib" das Multilib-Repository aktiviert werden:

```
[multilib-testing]  
Include = /etc/pacman.d/mirrorlist  
[multilib]  
Include = /etc/pacman.d/mirrorlist
```

Mit CRTL+O und ENTER abspeichern und mit CRTL+X beenden.

Anschließend muß die Pacman Repository Datenbanken neu geladen werden:

```
pacman -Sy
```

I. Nützliche Programme, Tools und Dienste

```
pacman -S acpid avahi cups mtools udisks2 git
pacman -S ntfs-3g unrar unzip gvfs
pacman -S ttf-dejavu ttf-liberation noto-fonts leafpad
pacman -S alsa-tools alsa-utils pulseaudio-alsa pavucontrol
pacman -S bash-completion networkmanager network-manager-applet
pacman -S doublecmd chromium thunderbird thunderbird-i18n-de
pacman -S libreoffice-fresh libreoffice-fresh-de hunspell-de gimp
pacman -S mpv gnome-disk-utility k3b asunder clementine
```

Acpid steht für die Energieverwaltung, avahi für die Netzwerkerkennung und cups für den Drucker. Das UDisks- und das gvfs-Paket enthalten Tools und Bibliotheken für den Zugriff auf von Festplatten und Speichergeräten und dessen Bearbeitung. Das ntfs-3g-Paket ermöglicht den Zugriff auf ntfs-formatierte Festplatten unter Windows-Betriebssystemen.

m. Installation von X, Grafiktreiber und der Xfce-Desktop

X ermöglicht die Benutzung einer graphischen Oberfläche unter einem Linux-System. Die Basispakete von Xorg werden installiert mit

```
pacman -S xorg-server xorg-xinit
```

Nun wird der Treiber für die Grafikkarte hinzugefügt. Mit

```
lspci | grep VGA
```

zeigt die vorhandene Grafikkarte. In meinem Rechner steckt eine NVidia GTX 1070. PC-Spieler mit einer NVidia-Karte wählen vorzugsweise den proprietären NVidia Grafiktreiber:

```
pacman -S nvidia-utils nvidia
```

Wem der Open Source Treiber reicht, gibt für seine Grafikkarte den passenden Befehl ein:

Hersteller	Befehlsaufruf des Open Source Pakets
Intel	<code>pacman -S xf86-video-intel</code>
AMD	<code>pacman -S xf86-video-amdgpu</code>
Nvidia	<code>pacman -S xf86-video-nouveau</code>

Ist man sich jedoch weiterhin unsicher, welches Treiber-Paket zu seiner Grafikkarte paßt, so lassen sich auch die gesamten Videotreiber installieren:

```
pacman -S xorg-drivers
```

Xfce gehört zu den stabilsten und sparsamsten Bedienungsoberflächen unter Linux. Die Basis Xfce-Installation mit einigen Extras:

```
pacman -S xfce4 xfce4-goodies arc-gtk-theme
```

```
pacman -S lightdm lightdm-gtk-greeter lightdm-gtk-greeter-settings
```

Mittels der Benutzeroberfläche `lightdm-gtk-greeter-settings` ist es möglich, später das Hintergrundbild auf dem Begrüßungsbildschirm anzupassen. Siehe auch

[Konfiguration | Xfce4-Desktop](#)

n. Benötigte Dienste aktivieren

Um diese Dienste automatisch beim Booten zu starten, werden sie mit folgendem Aufruf aktiviert:

```
systemctl enable acpid avahi-daemon cups.service
systemctl enable lightdm.service NetworkManager
systemctl enable fstrim.timer systemd-timesyncd.service
```

Der acpid Dienst sorgt für die Energieverwaltung, avahi für die Netzwerkerkennung und cups ist ein Drucksystem für Linux, worüber sich zahlreiche Drucker nutzen lassen. Der Dienst lightdm.service bzw. lightdm richtet den Login-Manager für den Xfce-Desktop ein. Bitte auf Großschreibung **NetworkM**anager achten. Beim ersten Start des Systems startet der Display-Manager und nach dem Login läuft der Network-Manager, der automatisch eine IP-Adresse mit einer Internetverbindung erstellt. Wird das System, wie in unserem Falle, auf einer SSD betrieben, die TRIM unterstützt, sollte der fstrim.timer aktiviert werden. Für die automatische Korrektur der Zeit mit der Atomuhr über das Internet sorgt timesyncd.

o. GRUB Bootloader per UEFI

Die Installation des GRUB Bootloaders erfolgt mit

```
pacman -S grub
pacman -S efibootmgr
pacman -S dosfstools
```

Fortsetzung der GRUB Installation für 64bit UEFI:

```
grub-install --target=x86_64-efi --efi-directory=/boot --bootloader-id=grub
```

Abschließend muß noch die Grub Konfiguration generiert werden.

```
grub-mkconfig -o /boot/grub/grub.cfg
```

p. Neustart

Nachdem alles Benötigte installiert und konfiguriert ist, kann das installierte Arch Linux bzw. die

Chroot Umgebung verlassen werden mit dem Aufruf:

```
exit  
umount -R /mnt
```

Der Rechner-Neustart erfolgt mit:

```
reboot
```

q. Ersteinrichtung

siehe auch: [Xfce Desktop einrichten unter Konfiguration | Xfce4-Desktop](#)

Nach dem Neustart in den Arch-Linux Desktop müssen evtl. noch wichtige Anpassungen vorgenommen werden. So kann es durchaus sein, daß trotz aller Einstellungen auf Deutsch das englische Tastaturlayout aktiviert ist und auch bleibt. Das Editieren der entsprechenden Konfigurationsdateien haben in meinem Falle nichts geholfen. Aber es gibt einen Weg über ein Script, daß in den Xfce Autostart gelegt wird. Eine genaue Anleitung dazu gibt es im obigen Link zur Rubrik "Konfiguration | Xfce4-Desktop".

Man kann im Thunar-Dateimanager versteckte Dateien und Ordner (un-)sichtbar machen mit

```
CRTL+h
```

Aktualisieren Sie anschließend Ihr System mit:

```
pacman -Syu
```


