

# Arch Linux from Scratch

Diese Kurzanleitung beschreibt die Installation von **Arch Linux** in einer virtuellen Maschine unter VirtualBox mit deutschem Tastaturlayout und deutscher Systemsprache. Sie kann nahezu 1:1 auf "echte" Hardware adaptiert werden.

Grundsätzlich empfehle ich **Arch Linux** nach der Installationsanleitung im englischen oder deutschen **Arch Wiki** zu installieren. Hier sind alle Schritte sehr detailliert erklärt und sehr viele Querverweise sauber verlinkt.

Beginners Guide (english): [https://wiki.archlinux.org/index.php/Beginners%27\\_guide](https://wiki.archlinux.org/index.php/Beginners%27_guide)

Beginners Guide (german): [https://wiki.archlinux.de/title/Anleitung\\_f%C3%BCr\\_Einsteiger](https://wiki.archlinux.de/title/Anleitung_f%C3%BCr_Einsteiger)

Experienced Guide (english): [https://wiki.archlinux.org/index.php/Installation\\_guide](https://wiki.archlinux.org/index.php/Installation_guide)

Arch Linux Grundinstallation und mehr (german):

[http://www.schreibdichte.de/resource/dokumente/archlinux\\_grundinstallation\\_und\\_mehr/](http://www.schreibdichte.de/resource/dokumente/archlinux_grundinstallation_und_mehr/)

Diese Anleitung ist nur ein kurzer Abriss einer Installation eines deutschen Systems in Virtualbox.

## Warum Arch Linux?

---

Mit Arch Linux gibt es nur sehr wenig im System, von dem du nichts weißt. Du musst alles selbst konfigurieren, indem Du Konfigurationsdateien erstellst oder anpasst. Dieser Prozess ist nicht allzu schwer, da diese Konfigurationsdateien genau für diesen Zweck gedacht sind. Desweiteren kann man sich auf ein exzellentes Wiki verlassen!

Die Arch Linux Philosophie versucht nicht den Benutzer vor der Komplexität zu schützen. Stattdessen fokussiert sie sich darauf, die direkte Konfiguration so einfach wie möglich zu machen. Natürlich kann man auch sehr viel leichter Dinge funktionsunfähig machen.

Arch Linux aktualisiert sich sehr aggressiv aus den Upstream-Quellen. Das hat den Vor- und Nachteil, dass man sich immer am blutigen Ende des Schwertes bewegt.

Mir persönlich gefällt, dass die Verantwortlichkeit für die Stabilität der Software eher von den Entwicklern als von den Paketierern abhängig ist.

Das absolute Killer-Feature was Arch Linux von vielen Distributionen abhebt ist das Paketierungssystem. Pacman, ABS, AUR, makepkg und das PKGBUILD Format sind einfach genial.

Normalerweise muss man sich um das Paketieren nicht kümmern, alles installiert sich sauber und Abhängigkeiten werden korrekt gehandhabt, besonders wenn man nur aus dem offiziellen Repositories installiert.

Wenn einem aber etwas Bestimmtes an einem Paket nicht gefällt oder man eine andere Version benötigt, hat man alle Werkzeuge in der Hand um ein eigenes Paket zu erzeugen ohne die Verwaltung durch Pacman zu gefährden. (Pacman ist das Gegenstück zu dpkg/apt-get in Debian).

Meinen persönlichen "Aha-Effekt" hatte ich bei der Anbindung meines iPhone 5S an mein Arch System. Arch

Linux ist das erste Linux System, mit dem ich alle Features dieses Smartphones uneingeschränkt unter Linux nutzen kann. Dies ist der Aktualität der entsprechenden Pakete geschuldet.

## Voraussetzungen

---

### ISO downloaden

Das offizielle Arch Linux ISO gibt es unter: <https://www.archlinux.org/download/> . Mit welcher Version man installiert ist prinzipiell egal, da **Arch** ein Rolling Release ist, aber es schadet nicht, die neueste Version zu nehmen. Bitte nach dem Download des ISO-Files unbedingt die SHA1- oder MD5-Prüfsumme verifizieren.!

### VirtualBox VM erstellen

Gibt man bei der Bezeichnung der virtuellen Maschine "**Arch**" ein, wird automatisch **Arch Linux 64Bit** gewählt, genau richtig für uns.

In der Testumgebung nutze ich 512MB RAM und eine 8GB große, virtuelle Festplatte. Nicht vergessen, das ISO als virtuelles CD-Laufwerk zu konfigurieren.

### Hardware Uhr überprüfen

Ein kleiner und vielleicht unbedeutender Schritt, aber man spart sich ggf. später ein paar Unklarheiten und Boot-Vorgänge. Prüfe im BIOS, ob die Hardware-Uhr im PC UTC oder lokale Zeit anzeigt. Zeigt sie (in deutschsprachigen Ländern) eine Zeit, die (je nach Sommerzeit) der aktuellen Uhrzeit um ein bzw. zwei Stunden hinterherhinkt, dann ist es UTC. Diesen Sachverhalt einfach merken! Bei **VirtualBox** ist die Standardeinstellung "Hardware Uhr in UTC".

## Booten vom Installationsmedium (CD)

---

### 32- oder 64Bit Installation?

Im Bootmenü der CD muss entschieden werden ob man die 32-Bit-Version (i686) oder 64-Bit-Variante (x86\_64) installieren möchte. Wenn man nicht weiß, ob der Rechner bzw. der Prozessor 64-Bit-tauglich ist, kann man unter „**Hardware Information**“ (HDT) unter „**Processor**“, „**Main Processor**“ die Information „**x86\_64 : No oder yes**“ sehen. Bei "**yes**" ist der Rechner 64-Bit-tauglich. Zurücknavigieren kann man mit den Cursor-Tasten.

### Booten + Tastatur

---

# VM (Arch x86\_64) bis zum Root-Prompt anbooten.

## Deutsches Tastaturlayout setzen

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

## Partitionierung + Swap

---

### Festplatte partitionieren

Hier gibt es natürlich verschiedene Ansätze, bei denen berücksichtigt werden muss, ob man mit UEFI booten möchte, ob man eine MS-DOS oder GPT-Partitionstabelle benutzt. In unserem Beispiel werden wir eine MS-DOS-Partitionstabelle anlegen.

Hier im Beispiel wird angenommen, dass die erste Partition auf der ersten Festplatte für die Root-Partition und die zweite Partition für die Swap-Partition benutzt wird. Die Bezeichnung der Dateisysteme mit Labels ist sinnvoll.

```
/dev/sda1 für Root (/) mit 7 Gigabyte mit setzen des Bootflags  
/dev/sda2 für Swap mit 1 GigaByte
```

Ggf. bitte individuell anpassen.

```
fdisk /dev/sda
```

```
n -> p -> 1  
First sector: 2048  
Last sector: +7G  
n -> p -> 2  
First sector: übernehmen  
Last sector : übernehmen  
t -> 2 > 82  
a -> 1  
w
```

### Ext4 Dateisystem auf Partition 1 mit dem Label "arch" anlegen

```
mkfs.ext4 -L arch /dev/sda1
```

### Swap Partition auf Partition 2 mit dem Label "swap" anlegen

```
mkswap -L swap /dev/sda2
```

# Einbinden der Partitionen

```
mount /dev/sda1 /mnt
```

Wenn man verschiedene Partitionen für z.B. /boot /root usw. verwendet haben, so muss man unter /mnt die entsprechenden Verzeichnisse anlegen und dann die Partitionen dort hin mounten, z.B.

```
mkdir /mnt/boot  
mkdir /mnt/home  
mount /dev/sda3 /mnt/boot  
mount /dev/sda4 /mnt/home
```

In unserem Beispiel aber unrelevant, alles ist unter /mnt.

# Swap aktivieren

per Label:

```
swapon -L swap
```

oder per Device:

```
swapon /dev/sda2
```

Bitte jeweils nur 1 Option wählen, nicht beide!

# Internetverbindung herstellen

---

Erst mal mit PING testen, ob die Verbindung ggf. schon steht

```
ping -c3 www.pcfreak.de
```

Falls noch keine Verbindung zum Internet hergestellt werden kann, mit dem Befehl

```
ip link
```

die Netzwerkkarten anzeigen lassen, z.B.

```
1: lo: <LOOPBACK ...  
2: enp4s0: <BROADCAST ...
```

und dann z.B. per DHCP eine Adresse für das entsprechende Interface beziehen:

```
dhcpcd enp4s0
```

Bekommt man an diesem Punkt kein Internet zu Stande, sollte man das Arch-Wiki konsultieren und alles probieren, bis es klappt. Hat bei mir bisher aber immer geklappt, selbst WLAN ist hier kein Problem!

# Basissystem auf Festplatte installieren

---

Vor der Installation müssen wir die Datei

```
/etc/pacman.d/mirrorlist
```

anpassen, also vorher ein Backup erstellen:

```
cp /etc/pacman.d/mirrorlist /etc/pacman.d/mirrorlist.bak
```

Die Datei sollte so editiert werden, dass man die Mirror-Server des entspr. Landes benutzt. Wir wollen also deutsche Server aktivieren. Wer sich das mühsame Editieren mit nano ersparen will, kann auch diesen Befehl absetzen:

```
grep -E -A 1 ".*Germany.*$" /etc/pacman.d/mirrorlist.bak | sed '/--/d' > /etc/pacman.d/mirrorlist
```

Nun werden mit Befehl `pacstrap` die Pakete **base**, **base-devel** und **wpa\_supplicant** in unseren Mountpoint installiert. Diese bilden das Basissystem.

```
pacstrap /mnt base base-devel wpa_supplicant
```

Dazu werden ca. **100 MB** aus dem Internet geladen.

## FSTAB erzeugen

Für unser neues System benötigen wir für das spätere Mounten der Partitionen natürlich eine FSTAB-Datei. Diese kann man sehr einfach mit dem Befehl

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

erzeugen. Will man eine FSTAB mit Labels ist der Befehl

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

der Richtige. Ich persönlich ziehe UDID vor, das kann man aber später problemlos manuell anpassen, da **genfstab** die UDID bereits als Kommentar in die **FSTAB** einfügt.

## Wechsel in die Betriebssystemumgebung unter /mnt

```
arch-chroot /mnt/
```

## Grundkonfiguration der neuen Installation

---

Im Folgenden wird die neue Installation konfiguriert. **Systemd**, das bevorzugte Initsystem bei Arch Linux,

greift dabei auf einzelne Klartextdateien zurück.

## Rechnername festlegen

```
echo arch > /etc/hostname
```

## Spracheinstellung /etc/locale.conf

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

```
echo LC_COLLATE=C >> /etc/locale.conf
```

Welche Auswahlmöglichkeiten es hier für LANG gibt, kann man z.B. der Datei

```
| /etc/locale.gen
```

entnehmen

## Tastaturbelegung /etc/vconsole.conf

```
echo KEYMAP=de-latin1-nodeadkeys > /etc/vconsole.conf
```

```
#a font with Euro symbol
```

```
echo FONT=lat9w-16
```

Welche Keymaps sind möglich?

```
| find /usr/share/kbd/keymaps/ -name "*.map.gz" -exec basename {} \;
```

## Zeitzone festlegen durch Link /etc/localtime

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

Die verschiedenen Zonen kann man unter

```
| /usr/share/zoneinfo/
```

finden.

## /etc/hosts

Die Datei **/etc/hosts** muss normalerweise nicht verändert werden, da das Paket nss-myhostname die Auflösung des Hostnames übernimmt.

```
cat /etc/hosts
```

```
#<ip-address> <hostname.domain.org> <hostname>
```

```
127.0.0.1 localhost.localdomain localhost
```

```
::1 localhost.localdomain localhost
```

## /etc/resolv.conf

Diese Datei muß nur verändert werden, wenn das Netzwerk mit statischen IP-Adressen konfiguriert wird.

## /etc/locale.gen

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

Die Kommentare # am Anfang folgender Zeilen entfernen:

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

dann generieren mit:

```
locale-gen
```

## /etc/modules-load.d/mymodules.conf

Sollen Module geladen werden, die noch unbedingt nötig sind, bitte hier je Zeile pro Modul eintragen.

## /etc/pacman.conf

Diese Datei enthält die Informationen über die Installationsrepositorien und konfiguriert den Paketmanager **pacman**. Hier wird i. d. R. nichts geändert. Für den Fall, dass man die 64bit-Variante gewählt hat, und man weiß, dass man 32bit-Bibliotheken braucht, so ist das [multilib]-Repository zu aktivieren. Entferne dazu die Kommentarzeichen in den Zeilen:

```
[multilib]
SigLevel = PackageRequired TrustedOnly
Include = /etc/pacman.d/mirrorlist
```

Ich aktiviere die 32Bit Bibliotheken normalerweise schon, da einige Anwendungen diese benötigen werden.

Um die Pacman Repository Datenbanken neu zu laden, anschließend folgenden Befehl eintippen

```
pacman -Syy
```

## etc/pacman.d/mirrorlist

Bitte bei einigen Zeilen von Server in deiner Nähe, die # am Anfang der Zeile vor dem Server = entfernen. (Das dürften meist die Einträge im Abschnitt ## Germany sein).

Haben wir schon erledigt!

## Linux Kernel erzeugen

```
mkinitcpio -p linux
```

## Setzen des Root-Passworts.

Man vergewissere sich, dass man es nicht vergisst!

```
passwd
```

```
Enter new UNIX password: geheim
```

```
Retype new UNIX password: geheim
```

## Installation des GRUB Bootloaders

### Paket Grub installieren

```
pacman -S grub
```

### Grub Konfiguration erzeugen

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

### Grub auf Festplatte installieren

```
grub-install /dev/sda
```

## Ins neue System booten

Zum Schluß das installierte Arch-Linux verlassen und wieder in das Installationsmedium zurückkehren:

```
exit
```

```
umount /dev/sda1
```

```
swapoff -a
```

und neu starten.

```
reboot
```

und als "root" anmelden!

## Konfiguration des neuen Systems

---

# Netzwerk erneut herstellen

```
ip link
```

```
1: lo: <LOOPBACK ...  
2: enp4s0: <BROADCAST ...
```

```
dhcpcd enp4s0
```

## Testen

```
ping -c3 www.pcfreak.de
```

# Benutzer erstellen

Die täglichen Arbeiten sollten nicht mit dem root-Account gemacht werden. Es ist mehr als nur schlechte Angewohnheit; es ist gefährlich. Root ist für administrative Aufgaben. Deshalb wird nun ein normaler Benutzer hinzugefügt. Bitte beachten sie, dass Benutzernamen nur Kleinbuchstaben und Sonderzeichen enthalten dürfen:

In diesem Beispiel heißt der Benutzer pcfreak:

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

```
passwd pcfreak
```

Geben Sie ein neues UNIX-Passwort ein: geheim

Geben Sie das neue UNIX-Passwort erneut ein: geheim

passwd: Passwort erfolgreich geändert

# Sudo installieren und einrichten

## Sudo installieren

```
pacman -S sudo
```

## Sudo konfigurieren

```
EDITOR=nano visudo
```

dann die Zeile

```
%wheel ALL=(ALL) ALL
```

aktivieren (Kommentarzeichen entfernen).

## Benutzer zur Gruppe wheel hinzufügen

```
gpasswd -a pcfreak wheel
```

Nun melden wir uns ab (logout) und als der neue Benutzer (pcfreak) wieder an. Ab jetzt führen wir alle Befehle die ROOT-Rechte benötigen unter diesem Benutzeraccount mit SUDO aus!

## Einen vernünftigen Editor installieren (optional)

```
sudo pacman -S vim
```

und vim gleich ein wenig konfigurieren:

```
cp /usr/share/vim/vimfiles/archlinux.vim ~/.vimrc  
echo "syn on" >> ~/.vimrc
```

## Ein paar wichtige Dienste nachinstallieren

```
sudo pacman -S acpid ntp dbus avahi cups cronie
```

### Dienste aktivieren

```
sudo systemctl enable <Dienstname>
```

In unserem Fall:

```
sudo systemctl enable cronie  
sudo systemctl enable acpid  
sudo systemctl enable ntpd  
sudo systemctl enable avahi-daemon  
sudo systemctl enable cups
```

## Automatische Zeiteinstellung

Wer die Zeit automatisch korrigieren lassen will, kann das mit ntp machen.

### Installieren des Pakets ntp

```
sudo pacman -S ntp
```

Ändern der Konfigurationsdatei z.B. für einen deutschen Zeitserver von dem die genaue Uhrzeit kommt.

```
sudo vim /etc/ntp.conf
```

```
server 0.de.pool.ntp.org  
server 1.de.pool.ntp.org  
server 2.de.pool.ntp.org  
server 3.de.pool.ntp.org
```

Für die erste Zeitkorrektur gibt man ein:

```
sudo ntpd -gq
```

...kurz warten

## Hardware-Uhr setzen

```
sudo hwclock -w
```

## OPTIONAL LTS-Kernel installieren

```
sudo pacman -S linux-lts
```

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

```
sudo mkinitcpio -p linux
```

```
sudo mkinitcpio -p linux-lts
```

## Zusammenfassung

Wir haben nun eine Arch Linux Grundinstallation ohne grafische Oberfläche. Das System benötigt im Moment nur ca. 1.4 GByte!!!

## Default DHCP Daemon/Client anpassen

Ich mag dhcpcd persönlich nicht, ich nutze lieber dhclient, deshalb **dhcpcd deinstallieren** und **dhclient installieren**.

```
sudo pacman -S dhclient
```

```
sudo pacman -R dhcpcd
```

## net-tools installieren

Einige Befehle rund ums Netzwerk z.b. **ifconfig** sind noch nicht installiert. Das Paket **net-tools** bringt sie mit:

```
sudo pacman -S net-tools
```

## Installation und Konfiguration von X

---

### Xorg Basispakete installieren

```
sudo pacman -S xorg-server xorg-xinit xorg-utils xorg-server-utils
```

ca. 10 Mbyte

### Grafiktreiber installieren

Grafikkarte ermitteln

```
lspci | grep VGA
```

InnoTek Systemberatung GmbH VirtualBox Graphics Adapter

## Mögliche Treiber anzeigen

```
pacman -Ss xf86-video | less
```

## Für VirtualBox virtualbox-guest-utils

```
sudo pacman -S virtualbox-guest-utils
```

```
sudo pacman -S linux-headers
```

```
sudo pacman -S linux-lts-headers
```

```
sudo pacman -S virtualbox-guest-dkms
```

```
sudo dkms install vboxguest/4.3.14
```

```
sudo systemctl enable dkms.service
```

```
sudo modprobe -a vboxguest vboxsf vboxvideo
```

```
sudo vim /etc/modules-load.d/virtualbox.conf
```

```
#This file will load the VirtualBox Modules at startup
#You should also install and enable virtualbox-guest-dkms
vboxguest
vboxsf
vboxvideo
```

Alternativ kann man auch **alle** Grafiktreiber (außer VirtualBox) durch Installation des Pakets **xorg-drivers** bereitstellen:

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

Xorg sucht sich dann später selbst den richtigen Treiber.

## Touchpad Treiber

```
sudo pacman -S xf86-input-synaptics
```

## Deutsche Tastaturbelegung für Xorg einstellen

Hinweis: Seit der Einführung von systemd kann die Tastaturbelegung statt über Änderung der **20-keyboard.conf** auch mit dem Befehl

```
localectl set-x11-keymap <layout> <model> <variant>
<options>
```

eingestellt werden. Wir machen es über die **20-keyboard.conf** :

```
sudo vim /etc/X11/xorg.conf.d/20-keyboard.conf
```

und folgendes hinzufügen:

```
Section "InputClass"
  Identifier "keyboard"
  MatchIsKeyboard "yes"
  Option "XkbLayout" "de"
  Option "XkbModel" "pc105"
  Option "XkbVariant" "de_nodeadkeys"
EndSection
```

**Alle möglichen XKbLayouts anzeigen mit:**

```
| localectl list-x11-keymap-layouts | less
```

**Alle möglichen XkbModels anzeigen mit**

```
| localectl list-x11-keymap-models | less
```

**Alle möglichen XkbVariants anzeigen mit:**

```
| localectl list-x11-keymap-variants | less
```

## Schöne Schriften

```
sudo pacman -S ttf-dejavu
```

## Desktop/Fenstermanager installieren

In unserem Fall wählen wir Gnome (Gnome 3)

```
sudo pacman -S gnome gnome-extra
```

```
| ca. 500 MB
```

## .xinitrc kopieren und anpassen

als **pcfreak** ausführen (normaler User)

```
cp /etc/skel/.xinitrc ~/
```

dann die Datei **~/xinitrc** editieren

und darin

```
| exec gnome-session
```

aktivieren.

Mit

```
startx
```

sollte nun Gnome starten (Geduld)! Danach wieder abmelden!

## Grafischen Loginmanager (GDM) aktivieren

```
sudo pacman -S gdm
```

(bereits installiert)

```
sudo systemctl enable gdm.service
```

## GDM: Standard Tastaturlayout wählen

Sollte das default Keyboardlayout abgeändert werden - oder von "en" auf deutsch (sowie andere Sprachen) umgestellt werden, muss die `~/.dirc` wie folgt konfiguriert werden:

```
[Desktop]
Language=de_DE.UTF-8
Layout=de nodeadkeys
```

## Installation und Konfiguration von ALSA

```
sudo pacman -S alsa-utils
```

Benutzer **pcfreak** zur Gruppe **audio** hinzufügen.

```
sudo gpasswd -a pcfreak audio
```

## gnome tweak tool

```
sudo pacman -S gnome-tweak-tool
```

## network manager

```
sudo pacman -S networkmanager network-manager-applet
```

```
sudo systemctl enable NetworkManager.service
```

## network manager plugins for vpn

```
sudo pacman -S networkmanager-openconnect networkmanager-openvpn networkmanager-pptp networkmanager-vpnc
```

Einstellen der Netzwerkverbindungen und VPN am besten über

```
nm-connection-editor
```

geht aber auch sauber aus dem Applet durch drücken auf "+"

## Renaming der Netzwerkinterfaces unter Systemd

NICHT verwenden: eth0, wlan0 usw. diese Namen sind dem Kernel vorenthalten!

Wir editieren die Datei `/etc/udev/rules.d/10-net-naming.rules`

```
# This file will rename network interfaces
# Do not use ethX or wlanX
# MAC addresses must be lower case!
SUBSYSTEM=="net", ACTION=="add", ATTR{address}=="00:24:d6:12:
5c:a4", NAME="net0"
```

## SSH

[https://wiki.archlinux.org/index.php/Secure\\_Shell#Installing\\_OpenSSH](https://wiki.archlinux.org/index.php/Secure_Shell#Installing_OpenSSH)

Empfehlenswert ist zunächst **haveged** zu installieren, dieses Paket sorgt für mehr Entropie:

```
sudo pacman -S haveged
sudo systemctl start haveged
sudo systemctl enable haveged
```

Danach OpenSSH

```
sudo pacman -S openssh
systemctl start sshd
systemctl enable sshd
```

ala entspr. Benutzer dann:

```
ssh-keygen
```

\*\*Als Alternative zur Aktivierung des Dienstes "openssh" bietet der SSH Daemon die Möglichkeit in per "Socket Aktivierung" zu starten. Wählt man diese Methode, hört "systemd" auf dem SSH Socket und startet den Daemon-Prozess nur für eingehende Verbindungen bei Bedarf. Die Einrichtung ist einfach.

```
sudo systemctl stop sshd
sudo systemctl disable sshd
sudo systemctl start sshd.socket
sudo systemctl enable sshd.socket
```

## Für Laptops

```
sudo modprobe acpi-cpufreq
```

(wenn klappt einbauen in /etc/modules-load.d/moduleladen.conf)

```
sudo pacman -S acpid
```

```
sudo systemctl enable acpid
```

```
sudo systemctl start acpid
```

## AUR enablen

---

Normalerweise müsste man yaourt, das Tool zur Installation aus dem AUR (Arch user Repository) manuell bauen usw. Einfacher geht es, zur Datei **/etc/pacman.conf** folgenden Eintrag hinzuzufügen:

```
[archlinuxfr]
```

```
SigLevel = Optional TrustAll
```

```
Server = http://repo.archlinux.fr/$arch
```

danach

```
sudo pacman -Syy
```

```
sudo pacman -Syu
```

Nun kann man *\*yaourt* installieren durch:

```
sudo pacman -Syu yaourt
```

Rsync sollte man noch installieren:

```
sudo pacman -S rsync
```

## FERTIG

---

### pacman Cheat Sheet

---

```
pacman -Sy # synchronisiere die Repository Datenbank wenn notwendig
```

```
pacman -Syy # erzwing die Synchronisation der Repository Datenbank
```

```
pacman -Ss xyz # durchsuche das Repository nach dem Paket xyz
```

```
pacman -S xyz # installiere das Paket xyz
```

```
pacman -Sy xyz # synchronisiere das Repository und installier Paket xyz
```

```
pacman -Syy xyz # "wirkliche" Synchronisierung des Repos und Installation von Paket xyz
```

```
pacman -R xyz # entferne das Paket xyz aber behalte die Abhängigkeiten installiert
```

```
pacman -Rs xyz # entferne das Paket xyz und all seine Abhängigkeiten (wenn sie nicht mehr benötigt werden)
```

```
pacman -Rsc xyz # entferne Paket xyz, all seine Abhängigkeiten und Pakete die vom ihm abhängen
```

```
pacman -Ql xyz # zeige alle Dateien die von Paket xyz installiert wurden
```

```
pacman -Qo /path # finde das Paket welches die Dateien im Pfad /path installiert hat
```

Manchmal ist das gewünschte Paket nicht (oder noch nicht) in den Standard-Repositories verfügbar, aber irgendjemand hat bereits ein "AUR-Paket" erzeugt. In diesem Fall kann man sich sehr schnell sein eigenes Paket aus dem entsprechenden PKGBUILD schnüren (oder mit AUR Paketmanagern wie yaourt direkt installieren).

## Paket vom PKGBUILD bauen und installieren

```
wget https://aur.archlinux.org/.../PKGBUILD
```

```
makepkg
```

```
sudo pacman -U path-to-generated.tar.xz
```

Sollte es hier zu **conflicting files** kommen:

### WARNUNG!

Bevor man diesen Schritt durchführt immer erst unter <http://www.archlinux.org/news> nachsehen ob es nicht bereits eine Lösung für das Problem gibt. Manchmal kann es sein, dass durch die folgende Lösung das System beschädigt wird!

Manchmal schlägt das Update mittel **-Su** mit folgendem Fehler fehl:

```
error: failed to commit transaction (conflicting files)
package: /path/to/file exists in filesystem
```

Um dies zu umschiffen, einfach die Installation erzwingen:

```
pacman -S --force package
```