

Pacman

Installation

Pacman ist bereits Bestandteil jeder Arch Linux-Installation.

Die wichtigsten Pacman-Befehle

Für neue Arch Linux-AnwenderInnen seien zunächst die wichtigsten Pacman-Befehle angeführt.

Befehl	Beschreibung
<code>pacman -Syy</code>	Update
<code>pacman -Syu</code>	Upgrade
<code>pacman -S <Paket></code>	Installiert ein Paket
<code>pacman -R <Paket></code>	Deinstalliert ein Paket
<code>pacman -U <Paket-Dateiname></code>	Installation eines Paketes aus einer lokalen Datei (z.B. für ein Downgrade oder zur Installation selbst gebauter Pakete)

Diese Befehle sind der Grundstock, um ein Arch-System einrichten und auf aktuellem Stand halten zu können.

Die Pacman-Syntax

Der Aufruf der Hauptoptionen wird mit einem Minuszeichen eingeleitet und beginnt immer mit einem Grossbuchstaben.

- **S** – Synchronisation der lokalen Paketdatenbank mit den Repositorien, welche in der Datei `/etc/pacman.conf` aktiviert sind (*Sync*)
- **Q** – Bezieht sich immer auf die lokale Paketdatenbank des Systems (*Query*)
- **R** – Dient zum Entfernen von installierten Paketen (*Remove*)

- **D** – Bearbeitung der lokalen Paketdatenbank (*Database*)
- **U** – Installiert Pakete aus einem lokalen Verzeichnis (*Upgrade*)
- **F** – Sucht das Paket, welches eine Datei beinhaltet (*File*)

Die vorangestellte Hauptoption kann mit weiteren Optionen in Kleinbuchstaben erweitert und spezifiziert werden, beispielsweise:

- **s** – Bewirkt eine Suche. Bei `-Ss` wird ein Paketname in der Paketdatenbank gesucht, Bei `-Rs` werden die jeweils abhängigen Pakete gesucht und mit entfernt
- **y** – Bewirkt eine Aktualisierung der Paketdatenbank (falls es tatsächlich Änderungen gibt)
- **yy** – Erzwingt die Aktualisierung der Paketdatenbank auch dann, wenn keine Updates vorhanden sind

Synchronisation und Installation von Paketen

Befehl	Beschreibung
<code>pacman -Syu</code>	Führt eine komplette System-Aktualisierung aus
<code>pacman -S <Paket1 Paket2></code>	Eines oder mehrere Pakete installieren oder aktualisieren
<code>pacman -Sy</code>	Lokale Datenbank aktualisieren
<code>pacman -Su</code>	Alle installierten Pakete aktualisieren
<code>pacman -Syy</code>	Lokale Datenbank komplett neu aufbauen und aktualisieren
<code>pacman -Syuu</code>	Alle installierten Pakete downgraden (von Testing nach Core/Extra)
<code>pacman -S testing/<Paket></code>	Paket aus einem spezifischen Repo (hier: testing) installieren
<code>pacman -Sw <Paket></code>	Paket herunterladen, ohne es zu installieren

Informationen zu installierbaren Paketen

Befehl	Beschreibung
<code>pacman -Ss <Paket></code>	Sucht nach installierbaren Paketen. Es reicht ein Teil des Paketnamens oder der Paketbeschreibung
<code>pacman -Sg</code>	Sucht nach installierbaren Paketgruppen
<code>pacman -Sg <Paketgruppe></code>	Zeigt den Inhalt einer Paketgruppe

pacman -Si <Paket>	Informationen zu (noch) nicht installierten Paketen anzeigen
--------------------	--

Verwaltung lokaler Pakete

Befehl	Beschreibung
pacman -U <Paket-Dateiname>	Ein lokales Paket installieren (nicht aus einem Repo)
pacman -D --asexplicit <Paket>	Status eines installierten Paketes auf „Ausdrücklich installiert“ setzen
pacman -D --asdeps <Paket>	Status eines installierten Paketes auf „Installiert als Abhängigkeit“ setzen
pacman -Scc	Leert das lokale Paketarchiv von Pacman vollständig (<i>/var/cache/pacman/pkg</i>)
pacman -Sc	Löscht veraltete Pakete aus <i>/var/cache/pacman/pkg</i> sowie ungenutzte Repositorien aus <i>/var/lib/pacman/sync</i> . Vorsicht , behalten werden nur aktuell installierte Versionen – Vorversionen für ein Paket-Downgrade sind dann nicht mehr vorhanden.

Pakete entfernen

Befehl	Beschreibung
pacman -R <Paketname>	Deinstallation eines oder mehrerer Pakete (aus AUR oder Repos)
pacman -Rs <Paketname>	Wie oben, zusätzlich werden alle abhängige Pakete gesucht und mit entfernt, falls diese nicht von einer anderen Anwendung gebraucht werden
pacman -Rsc <Paketname>	Wie oben. Abhängigkeiten werden kaskadierend entfernt
pacman -Rscn <Paketname>	Wie oben. Die Konfigurationsdateien der Anwendung werden mit entfernt
pacman -Rdd <Paket>	Deinstallation eines Paketes ohne Prüfung bestehender Paketabhängigkeiten. Vorsicht , hierdurch kann die Konsistenz und Funktionstüchtigkeit der Systeminstallation beeinträchtigt werden!
pacman -Rss <Paket>	Paket mit allen benötigten Abhängigkeiten und deren Abhängigkeiten entfernen

Abfragen der lokalen Paketdatenbank

Befehl	Beschreibung
pacman -Q	Zeigt alle installierten Pakete inklusive Versionsnummer auf dem System an
pacman -Qi <Paket>	Informationen zu bereits installiertem Paket anzeigen
pacman -Qs <Suchmuster>	Installierte Pakete nach Name oder einem Begriff in der Beschreibung durchsuchen.
pacman -Qdt	Verwaiste Pakete anzeigen, die als Abhängigkeiten installiert wurden, aber nicht mehr von anderen Paketen benötigt werden
pacman -Qet	Pakete anzeigen, die ausdrücklich installiert wurden, aber nicht von anderen als Abhängigkeit benötigt werden
pacman -Qi <Paket>	Zeigt alle installierten Dateien des Pakets im System
pacman -Qm	Pakete anzeigen, die sich in keinem aktivierten Repo laut /etc/pacman.conf befinden
pacman -Qo <Pfad zur Datei>	Zeigt das Paket an, welches die gesuchte Datei enthält
pacman -Sy && pacman -Qu	Aktualisiert die lokale Paketdatenbank und zeigt verfügbare Updates an
pacman -Qk grep warning	Überprüft alle Pakete auf fehlende Dateien und schränkt die Ausgabe auf Problempakete ein

Paketdateien suchen

Befehl	Beschreibung
pacman -Fy	Lokale Datenbank aktualisieren (wie -Sy)
pacman -Fyy	Lokale Datenbank neu aufbauen und aktualisieren (wie -Syy)
pacman -F <Datei>	Paket suchen das die Datei enthält
pacman -Fx <Regex>	Wie -F, aber Suche mit regulären Ausdrücken (hilfreich, falls der vollständige Dateiname unbekannt ist)
pacman -Fi <Paket>	Alle Dateien des Paketes anzeigen

Es können auch mehrere Befehlsaufrufe kombiniert werden; hier z.B. zur vorhergehenden Prüfung und anschließenden Deinstallation aller verwaisten Pakete:

```
pacman -Qdtq
pacman -Rsn $(pacman -Qdtq)
```

Paccache

Paccache ist ein Werkzeug zur Bereinigung des Pacman-Cache `/var/cache/pacman/pkg`. Archivierte Pakete können hiermit differenzierter als mit `pacman -Scc` bereinigt werden.

Um den Befehl nutzen zu können muss zuvor das Paket `pacman-contrib` installiert werden.

Zur Bereinigung des Paket-Cache kann man beispielsweise so vorgehen:

```
du -sh /var/cache/pacman/pkg # Belegung des Cache-Speicherplatzes prüfen
paccache -h                  # Befehlsübersicht
paccache -dk2                # Testdurchlauf; Wieviel Platz kann eingespart werden?
paccache -vrk2               # Entfernt Pakete aus dem Cache, behält die jüngsten 2 Versionen
paccache -ruk0               # Alle Pakete aus dem Cache entfernen, die nicht (mehr) installiert sind
```

Zur automatischen Bereinigung gibt es einen Systemd Timer. Bei aktiviertem Timer wird der Cache wöchentlich mit `paccache -r` bereinigt.

```
systemctl enable paccache.timer
```

Checkupdates

Mit *Checkupdates* lässt sich prüfen, ob Aktualisierungen für installierte Pakete verfügbar sind, ohne eine komplette System-Aktualisierung (`pacman -Syu`) durchzuführen.

Um den Befehl nutzen zu können muss zuvor das Paket `pacman-contrib` installiert werden.

```
checkupdates
```

Da die Prüfung auf Updates mit *Checkupdates* ohne Root-Rechte durchgeführt werden kann, eignet sich der Befehl sehr gut zur Verwendung in Conky und Scripts.

Pacman-Datenbank

Pacman speichert alle Paketinformationen in Form einer Vielzahl einzelner Dateien als logisch zusammengehörenden Datenbestand im Verzeichnis `/var/lib/pacman`.

Installierbare Pakete:

Die Paketinformationen der in der Pacman-Konfigurationsdatei `/etc/pacman.conf` aktivierten

Repositorien (s.u.) werden im Verzeichnis `/var/lib/pacman/sync` gespeichert. Für diese Repos werden von Pacman Dateien mit folgenden Dateiendungen angelegt:

- `<Repo>.sig` – PGP-Sicherheitssignatur; diese ist nur für inoffizielle Repos erforderlich und vorhanden (*Textdatei*)
- `<Repo>.files` – Enthält die Textdateien `desc` und `files` mit den entsprechenden Informationen für jedes einzelne im Repo verfügbare Paket (*Gzip-Datei*)
- `<Repo>.db` – Enthält die Datei `desc` mit entsprechenden Informationen für jedes einzelne im Repo verfügbare Paket (*Gzip-Datei*)

Pacman-Befehle, die mit `-S` oder `-F` eingeleitet werden, beziehen sich auf diesen Bereich der Datenbank; er kann mit `-Sy` aktualisiert und mit `-Syy` neu generiert werden.

Installierte Pakete:

Die Paketinformationen der im System vorhandenen Pakete befinden sich im Unterverzeichnis `/var/lib/pacman/local`. Für alle installierten Pakete werden dort gesonderte Verzeichnisse mit Namen und Version des Paketes angelegt; folgende Dateien sind darin enthalten:

- `desc` – Paketinformationen, welche mit dem Befehl `pacman -Qi <Paketname>` abgerufen werden können (*Textdatei*)
- `files` – Pfadangaben aller mit dem Paket installierten Dateien, die mit dem Befehl `pacman -FI <Paketname>` abgerufen werden können (*Textdatei*)
- `mtree` – Zeit- und Größenangaben sowie Prüfsummen (Hashwerte) aller mit dem Paket installierten Dateien (*Gzip-Datei*)

Pacman-Befehle, die mit `-Q` oder `-D` eingeleitet werden, beziehen sich auf diesen Bereich der Datenbank. Bei Installationen und Deinstallationen erfolgt eine fortlaufende Aktualisierung. Dieser systembezogene Teil der Datenbank kann nicht neu generiert werden, da er Ausgangsreferenz der lokalen Paketverwaltung ist.

Konfiguration

Die Einstellungen für Pacman sind in der Datei `/etc/pacman.conf` gespeichert und können dort angepasst werden.

Allgemeine Einstellungen

```
#RootDir    = /
#DBPath     = /var/lib/pacman/
#CacheDir   = /var/cache/pacman/pkg/
#LogFile    = /var/log/pacman.log
#GPGDir     = /etc/pacman.d/gnupg/
HoldPkg     = pacman glibc
#XferCommand = /usr/bin/curl -C - -f %u > %o
#XferCommand = /usr/bin/wget --passive-ftp -c -O %o %u
#CleanMethod = KeepInstalled
Architecture = auto
```

```
# Pacman won't upgrade packages listed in IgnorePkg and members of IgnoreGroup
#IgnorePkg =
#IgnoreGroup =
```

```
#NoUpgrade =
#NoExtract =
```

- Pakete, die durch `HoldPkg` markiert sind, müssen vor dem Entfernen nochmals bestätigt werden.
- Pakete, die mit `IgnorePkg` markiert sind, werden vom Update völlig ausgenommen. Dies betrifft auch alle damit verbundenen Abhängigkeiten.
- Dateien, die durch `NoUpgrade` markiert sind, werden beim Update nicht überschrieben. Pacman legt stattdessen eine neue Datei im Format `datei.pacnew` an.
- Dateien, die durch `NoExtract` markiert sind, werden bei der Installation oder einem Update nicht installiert.

Seit der Version 4.1 unterstützt Pacman auch farbige Ausgaben. Dazu muss die Option `#Color` auskommentiert werden.

Repositorien

Offizielle Arch Linux-Pakete sind einem von sechs unterschiedlichen Repositorien zugeordnet:

- `[core]` enthält grundlegende Programme, die zum Betrieb von Arch Linux unbedingt erforderlich sind.
- `[extra]` enthält eine Vielzahl zusätzlicher, optionaler Anwendungen in der jeweils letzten stabilen Version. Hier liegen etwa auch die Pakete von KDE und GNOME.
- `[testing]` enthält neue Versionen, die noch nicht hinreichend getestet wurden – mit anderen Worten: Dinge, die etwas kaputt machen könnten.
- `[multilib]` enthält 32-Bit-Anwendungen, die auf x86_64-Systemen installiert werden sollen.
- `[multilib-testing]` enthält noch nicht hinreichend getestete 32-Bit-Anwendungen, die auf x86_64-Systemen installiert werden sollen.

Repos festlegen

Die Konfiguration der `/etc/pacman.conf` ist weitgehend selbsterklärend und erfolgt durch Ein- oder Auskommentierung bestehender Einträge sowie durch Hinzufügen zusätzlicher Einträge.

Inoffizielle Repos verwenden

Weitere Repos kann man beliebig anhängen, indem man Repo-Name und Server spezifiziert. Eine Liste inoffizieller Repos findet man im [engl. Wiki](#).

Spiegelserver

Die Spiegelserver, von welchen Pacman die Pakete zur Installation und für Updates herunterladen soll, werden in der Datei `/etc/pacman.d/mirrorlist` gespeichert; ein Ausschnitt als Beispiel:

```
## Germany
Server = http://mirror.23media.de/archlinux/$repo/os/$arch
Server = http://mirror.gnomus.de/$repo/os/$arch
#Server = https://mirror.bethselamin.de/$repo/os/$arch
#Server = http://ftp.fau.de/archlinux/$repo/os/$arch
```

Die Wahl der Architektur und der Zugriff auf die Repos ist mit den Variablen `$repo` und `$arch` gekennzeichnet.

Pacman greift vorrangig auf den ersten einkommentierten Server-Eintrag zu. Sollte der Server nicht erreichbar sein oder ein Paket dort nicht vorgefunden werden, wird auf den nächsten Server zugegriffen. Für eine optimale Zugänglichkeit vom eigenen Standort aus ist die Wahl heimischer Spiegelserver ratsam.

Bei Veränderungen der weltweiten Spiegelserver werden von Pacman bei System-Updates automatisch neue Mirrorlisten als `/etc/pacman.d/mirrorlist.pacnew` hinterlegt. Bei Bedarf kann man die darin enthaltenen deutschen Server auslesen:

```
awk '/Germany/{i=1;next}/^##/{i=0}i{print}' /etc/pacman.d/mirrorlist.pacnew
```

Tipp: Es reicht aus, nur wenige Spiegelserver in der Mirrorliste zu aktivieren und ggf. weitere, aber auskommentierte Server darunterstehend zu belassen. So kann der favorisierte Server – falls er sich nicht bewähren sollte – leicht durch einen anderen aus der auskommentierten „Vorratsliste“ ersetzt werden.

Optimale Spiegelserver

Die Spiegelserver unterscheiden sich in ihrer Aktualität und Geschwindigkeit. Relevante Messwerte sind:

- **Completion:** Die Anzahl erfolgreicher Verbindungen bei Tests des Spiegelserver in Prozent; bei weniger als 100% könnte der Server unzuverlässig sein
- **Delay:** Durchschnittliche Verzögerungsdauer der letzten Synchronisation, diese sollte unter einer Stunde liegen
- **Duration:** Durchschnittliche Verbindungs- und Abbrufzeit in Sekundenbruchteilen; ein hoher Wert kann auf eine Überlastung des Servers hindeuten
- **Mirror Score:** Eine grobe Berechnung des Rankings, das sich aus den vorgenannten Werten ergibt

Eine vollständige Übersicht an Messergebnissen findet man auf der Seite [Mirror-Status](#).

Da die dort gelisteten Messergebnisse nicht dem eigenen Standort entsprechen, ist es sinnvoll,

selbst Messungen durchzuführen. Das Paket `pacman-contrib` stellt hierfür das Tool *Rankmirrors* zur Verfügung. Mit folgendem Befehl können die Antwortzeiten der in der Mirrorliste einkommentierten Server geprüft werden:

```
rankmirrors -n 0 -t /etc/pacman.d/mirrorlist
```

Falls man differenzierte Vergleichsergebnisse wie z.B. einen „Mirror Score“ wünscht, kann man das Python-Script `Reflector` verwenden. Eine Liste von 10 deutschen Spiegelservers mit der schnellsten Downloadrate erhält man z.B. mit diesem Befehl:

```
reflector -c Germany -p http -p https --sort rate -n 10
```

Fügt man am Ende des Befehles die Option `--info` hinzu, werden zusätzlich die auf der Seite „Mirror-Status“ verfügbaren Details angezeigt.

Mirrorliste aktualisieren

Solange die favorisierten Spiegelservers gut und verlässlich funktionieren, sind Aktualisierungen der Mirrorliste nicht erforderlich. Falls man die Datei `/etc/pacman.d/mirrorlist` jedoch auf Basis möglichst optimaler Messwerte aktualisieren möchte, bieten sich die vorgenannten Verfahren an.

- **Rankmirrors:** Man kann die Antwortzeiten der deutschen Spiegelservers aus der Datei `mirrorlist.pacnew` messen, aus den Ergebnissen z.B. die 15 schnellsten Server extrahieren und von diesen wiederum die 4 besten Spiegelservers aktivieren – diese gründliche Messung dauert einen Moment:

```
awk '/Germany/{i=1;next}/^##/{i=0}{print}' /etc/pacman.d/mirrorlist.pacnew | sed 's/^##/' > /tmp/mirror && ra
```

- **Reflector:** Man kann die deutschen Spiegelservers online abrufen, z.B. eine Kurzmessung der Downloadrate durchführen, 15 Server mit dem schnellsten Datendurchsatz extrahieren und von diesen die 4 besten Spiegelservers aktivieren:

```
reflector -c Germany -p http -p https --sort rate -n 15 | sed '11,14!s/^S/#S/'
```

Falls man eine Sortierung auf Basis des bei `Mirror-Status` abgerufenen „Mirror Score“ vornehmen möchte, kann man die Anweisung `--sort rate` durch `--sort score` ersetzen.

Hinweis: Beide oben angeführten Befehle geben die Resultate im Terminal aus. Um die vorhandene Mirrorliste zu ersetzen, kann als Ausgabeumleitung `> /etc/pacman.d/mirrorlist` angefügt werden.

Mirrorliste erweitern

Eine weitere Möglichkeit besteht darin, einen favorisierten Spiegelservers in der Pacman-Konfigurationsdatei `/etc/pacman.conf` vorzuschalten, zum Beispiel:

[core]

Server = [http://ftp-stud.hs-esslingen.de/pub/Mirrors/archlinux/\\$repo/os/\\$arch](http://ftp-stud.hs-esslingen.de/pub/Mirrors/archlinux/$repo/os/$arch)

Include = /etc/pacman.d/mirrorlist

Hinweis: Wird ein Spiegelserver in `/etc/pacman.conf` eingetragen, sollte unbedingt der gleiche Eintrag bei allen offiziellen Repos erfolgen (`[core]`, `[extra]`, `[multilib]`). Ansonsten besteht die Gefahr, dass aufgrund unterschiedlicher Aktualität der Server versucht wird, nicht kompatible Paketversionen zusammenzuführen.

Keine partiellen Upgrades

Arch Linux wird als *Rolling-Release* fortlaufend aktualisiert. Sobald neue Versionen von Programmbibliotheken [Wikipedia.png \[1\]](#) in den Repositorien erscheinen, werden vom Arch-Entwicklungsteam alle Pakete, die sich auf diese Bibliothek beziehen, neu (auf)gebaut. Auf diese Weise wird sicher gestellt, dass alle Pakete aus den offiziellen Repos mit der veränderten Schnittstelle der neuen Version funktionieren.

Sollte man fälschlicherweise nur ein einzelnes Paket aktualisieren, werden mit diesem Paket bei Bedarf neue Versionen von Programmbibliotheken im System integriert. Andere bereits installierte und von einem Upgrade ausgenommene Pakete blieben jedoch weiterhin von älteren Versionen der Bibliotheken abhängig und kämen mit den veränderten Schnittstellen neuerer Versionen nicht zurecht. Aus diesem Grunde werden partielle Upgrades nicht unterstützt.

Bevor ein neues Paket installiert wird, sollte immer zunächst mit `pacman -Syu` eine Aktualisierung des kompletten Systems durchgeführt werden. Vor dem gleichen Hintergrund sollte man auch bei Ausnahmen unter `IgnorePkg` und `IgnoreGroup` sehr umsichtig verfahren (s. [Konfiguration](#)).

Pakete, die nicht aus den offiziellen Repositorien installiert worden sind (z.B. aus dem [AUR](#)) und sich auf Bibliotheken aus den Repos beziehen, müssen im Falle einer Änderung von `so`-Namen (z.B. `libfoo.so.1` → `libfoo.so.1.2`) auf Anwenderseite neu gebaut werden.

Hinweis: Probleme, die auf Versionsunterschiede von Programmbibliotheken zurückzuführen sind, sollte man nicht durch „*Symlinking*“ verschlimmbessern. In der Regel lassen sich solche Probleme mit einem `'pacman -Syu'` beheben.

Graphische Oberflächen

Pacman wurde als reines CLI-Programm konzipiert, und greift grundsätzlich nur auf installierbare Pakete aus den offiziellen Repositorien zurück, nicht aber auf Skripte (PKGBUILDs) aus dem [AUR](#). Rund herum entstanden jedoch im Laufe der Jahre zahlreiche Wrapper, GUIs, oder Programme, die

Dateien aus den offiziellen Repositorien und dem AUR gleichzeitig aktualisieren können. Diese sind im Artikel [Graphische Paketmanager](#) ausführlich beschrieben. Die dort gesetzten Warn-Hinweise sollten unbedingt beachtet werden.

Hilfe

Keine Verbindung zu einem Mirror

Aufgrund einer langsamen Internetverbindung oder nicht optimaler Pacman-Konfiguration kann es vorkommen, dass der Verbindungsaufbau zu einem Spiegelserver zu lange dauert und es so zu einem [Wikipedia.png Timeout](#) kommt. Um dieses Problem zu beheben, kann man – wie im Abschnitt [Optimale Spiegelserver](#) beschrieben – den schnellsten Server suchen und verwenden. Sollte auch das nicht zu gewünschtem Erfolg führen, kann versucht werden, einen alternativen Downloader einzusetzen. Dazu muss man die Datei `/etc/pacman.conf` bearbeiten und einen der beiden Einträge durch Entfernen der Raute `#` auskommentieren:

```
#XferCommand = /usr/bin/{ {Paket|curl} } -C - -f %u > %o
#XferCommand = /usr/bin/{ {Paket|wget} } --passive-ftp -c -O %o %u
```

Bandbreite beschränken

Um die Download-Bandbreite zu beschränken, kann in in der Konfigurationsdatei `/etc/pacman.conf` innerhalb der Zeile ...

```
XferCommand = /usr/bin/wget --passive-ftp -c -O %o %u
```

... dem `wget`-Befehl die entsprechende Option hinzugefügt werden:

```
XferCommand = /usr/bin/wget --passive-ftp --limit-rate=40k -c -O %o %u
```

Pacman aus Versehen deinstalliert

Wie kann Pacman wiederhergestellt werden, wenn es aus Versehen deinstalliert wurde?

- Manuelle Variante

```
cd /
bsdtar -x -f /var/cache/pacman/pkg/pacman-*.pkg.tar.zst
```

Das Sternchen(*) entspricht dem Paket mit der höchsten Versionsnummer, welches hoffentlich noch im o.g. Verzeichnis vorhanden ist. Danach Pacman mittels des „neuen“ Pacman wieder installieren, damit der Datenbankeintrag für Pacman selbst wieder angelegt wird.

```
pacman -S pacman
```

- Per Installationsmedium

Vom Installationsmedium booten und die deutsche Tastaturbelegung `loadkeys de` wählen. Anschließend Partitionen einhängen: Die Root-Partition nach `/mnt` einhängen (Verzeichnis `/etc` sollte darin enthalten sein) Für weitere Partitionen unterhalb `/mnt` Ordner anlegen und entsprechend einhängen (bspw. `/mnt/usr`). Eine Neuinstallation von Pacman kann nun mit diesem Befehl erfolgen:

```
pacstrap /mnt pacman
```

Revision #2

Created 5 July 2023 08:30:09 by Hermann

Updated 27 July 2023 14:25:45 by Hermann