

# Arch – Vom Terminal-Prompt zum Desktop

---

Stand: 09.05.2024

Quellen: <https://www.freiesoftwareog.org>, <https://wiki.archlinux.org/>

---

„Ach übrigens, ich verwende Arch“

Wer diese Broschüre durcharbeitet, kann das in Zukunft auch jedem der's nicht hören will sagen...

## I heard he is using Arch Linux



Allgemeines.....	2
Installation und Konfiguration des Grundsystems.....	2
Deutsches Tastaturlayout.....	2
Testen, ob UEFI aktiviert ist.....	2
Partitionierung der Festplatte.....	2
Bei einem UEFI System.....	3
Bei einem „legacy“ System.....	3
Vorbereiten der Softwarequellen.....	3
Grundsystem installieren.....	4
Konfiguration des Netzwerks.....	4
root-Passwort setzen.....	4
GRUB installieren.....	5
Bei einem UEFI System.....	5
Bei einem „legacy“ System.....	5
Drucker einrichten.....	5
Benutzer anlegen und berechtigen.....	5
Anpassen der Openbox-Menüstruktur.....	6
Anpassen der Autostart-Konfiguration.....	6
Erweiterung des Grundsystems um eine GUI.....	7
X und Fenstermanager.....	7
Login-Manager.....	8
Erweiterung des Grundsystems um weitere Werkzeuge und Anwendungen.....	9
Weitere, optionale Werkzeuge und Konfigurationen.....	10
Benutzerverzeichnis-Ordner erstellen (optional).....	10
Statistische Informationen an das Arch-Projekt zurückmelden.....	10
Anpassen der lightdm-Konfiguration / Autologin (optional).....	10
jgmenu installieren und einbinden (optional).....	11
Einbindung des Menüs in tint2 (optional).....	11
Weitere Anpassungen der Openbox-Konfiguration (optional).....	12
Anpassen der systemweiten Zeitzone (optional).....	12
Codecs, Plugins und Multimedia-Programme (optional).....	13
Zusätzliche Schriftarten und Rechtschreibprüfungen (optional).....	13
Weitere optionale Schritte.....	14
Installation eines LTS-Kernels.....	14
Installation von Microcode für den Prozessor.....	14
Firewall einrichten.....	15
Grafischer Paketmanager.....	15
Yaourt.....	15
pamac.....	16
Paketmanager aufpeppen.....	16
Den schnellsten pacman-Spiegelservers einstellen.....	16
Farbe für pacman.....	16
Mehr Geschwindigkeit für pacman.....	17
Automatisches Bereinigen des Paket-Caches.....	17
Datensicherung und Backup.....	18
GRUB-Verzögerung deaktivieren.....	18
Weitere Informations- und Ideenquellen.....	19
Die Communities.....	19
Internetquellen.....	19
Schlussbetrachtung und eigene Notizen.....	19

## Allgemeines

Arch Linux ist eine unabhängig entwickelte, x86\_64-optimierte Linux-Distribution, die sich an erfahrene Linux-Nutzer richtet. Sie verwendet den selbst entwickelten Paketmanager „pacman“, um Aktualisierungen für die neuesten Softwareanwendungen mit vollständiger Verfolgung der Abhängigkeiten bereitzustellen. Arch arbeitet mit einem Rolling-Release-System und kann von einem CD-Image oder über einen FTP-Server installiert werden. Die Standardinstallation bietet eine solide Grundlage, die es dem Benutzer ermöglicht, eine benutzerdefinierte Installation zu erstellen. Darüber hinaus bietet das Arch Build System (ABS) die Möglichkeit, auf einfache Weise neue Pakete zu erstellen, die Konfiguration von Bestandspaketen zu ändern und diese Pakete über das Arch Linux User Repository mit anderen Benutzern zu teilen.

Für diesen Kurz-/Schnelldurchlauf einer typischen Arch-Linux Installation setze ich voraus, dass es bereits einen bootfähigen Datenträger mit der aktuellsten Arch-Version gibt (DVD, USB-Stick, ...).

Bei Drucklegung war dies die Version „2024.05.01“.

## Installation und Konfiguration des Grundsystems

Nachdem mit dem Boot-Datenträger gestartet wurde, geht es an die Vorbereitungen für die Installation.

### Deutsches Tastaturlayout

Damit man von Anfang an mit einem deutschen Tastaturlayout arbeiten kann, gibt man folgendes ein:

```
loadkeys de-latin1
```

Hierbei bitte beachten, dass zum Zeitpunkt dieser Befehlseingabe „z“ und „y“ noch vertauscht sind. Auch der Bindestrich liegt noch auf der Taste „ß“. Danach sollten Umlaute usw. aber verfügbar sein.

### Testen, ob UEFI aktiviert ist

Ebenfalls im Terminal mit:

```
ls /sys/firmware/efi/efivars
```

existiert dieses Verzeichnis, handelt es sich um ein UEFI-System. Diese Information wird relevant, wenn es gleich um die Partitionierung des Systems geht.

### Partitionierung der Festplatte

Zunächst verschafft man sich einen Überblick über die installierten Datenträger:

```
fdisk -l
```

## Bei einem UEFI System

```
fdisk /dev/sda
```

Bei einem UEFI-System muss die erste Partition eine ESP/EFI-Partition sein:

```
n → disk number: 1 → default block size → partition size:  
+512M
```

```
t → L für Liste → entsprechende EFI-Nummer eingeben
```

Das Dateisystem für diese spezielle Partition wird mit

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

erzeugt.

Die weiteren Partitionen (root, home und swap) werden entsprechend der „legacy“ Methode angelegt.

## Bei einem „legacy“ System

Hier entfällt die Erzeugung einer EFI-Partition und es werden nur die drei gewünschten Partitionen für root, home und swap angelegt:

```
n → disk number: 1 → default block size → partition size +10G
```

```
n → disk number: 2 → default block size → partition size +8G
```

```
n → disk number: 3 → default block size → partition size +2G
```

```
w
```

das Dateisystem für diese wird in beiden Fällen dann mit

```
mkfs.ext4 /dev/sda1
```

```
mkfs.ext4 /dev/sda2
```

```
mkswap /dev/sda3
```

erzeugt.

## Vorbereiten der Softwarequellen

Um die Paket-/Softwarequellen einzurichten benötigt es die folgenden Befehle:

```
pacman Syy
```

```
pacman -S reflector
```

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

```
reflector -c „DE“ -f 12 -l 10 -n 12 -save  
/etc/pacman.d/mirrorlist
```

## Grundsystem installieren

Um ein minimales Grundsystem zu erhalten werden nun folgende Befehle abgesetzt:

```
mount /dev/sda1 /mnt
pacstrap /mnt base linux linux-firmware nano xterm
```

Grundkonfiguration des installierten Arch-Systems:

```
genfstab -U /mnt >> /mnt/etc/fstab
arch-chroot /mnt
timedatectl list-timezones
timedatctl set-timezone Europe/Paris
```

Dann stellen wir noch die Lokalisierung korrekt ein. In der Datei

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

wird die Zeile mit „de\_DE“ entkommentiert und danach gespeichert.

Dann werden die Zeichentabellen usw., generiert:

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

## Konfiguration des Netzwerks

Zunächst wird der lokale Hostname eingetragen:

```
echo meinarch > /etc/hostname
```

Dann die Hosts-Datei erzeugt und mit dem Editor bearbeitet:

```
touch /etc/hosts
```

```
nano /etc/hosts
```

Die Einträge lauten:

```
127.0.0.1    localhost
::1         localhost
127.0.0.1    meinarch
```

## root-Passwort setzen

```
passwd
```

## GRUB installieren

Jetzt wird noch der Bootloader benötigt.

### Bei einem UEFI System

```
pacman -S grub efibootmgr
mkdir /boot/efi
mount /dev/sda1 boot/efi
```

```
grub-install --target=x86_64-efi --bootloader-id=GRUB
--efi-directory=/boot/efi
grub-mkconfig -o /boot/grub/grub.cfg
```

### Bei einem „legacy“ System

```
pacman -S grub
grub-install /dev/sda
grub-mkconfig -o /boot/grub/grub.cfg
```

## Drucker einrichten

Das Einrichten von Druckern unter GNU/Linux ist dank CUPS sehr einfach. Es kümmert sich um viele Probleme und stellt alle notwendigen Pakete für die gängigsten Drucker zur Verfügung. Um einen Drucker in Arch Linux einzurichten, kann man die folgenden Befehle verwenden. Der erste Befehl ist die Installation des CUPS-Pakets und dann das anschließende Aktivieren/Starten des Dienstes:

```
sudo pacman -S cups
sudo systemctl enable cups.service
sudo systemctl start cups.service
```

Und damit man die Einrichtung nicht immer über `http://localhost:631` machen muss, kann man noch das Paket `system-config-printer` installieren:

```
sudo pacman -S system-config-printer
```

## Benutzer anlegen und berechtigen

Da das System momentan nur den `root`-User kennt, und man mit diesem natürlich keinesfalls regulär arbeiten sollte, wird nun (mindestens) ein weiterer „normaler“ Benutzer angelegt und mit entsprechenden Berechtigungen ausgestattet:

```
pacman -S sudo
useradd -m edi
passwd edi
```

Diesen neuen User in benötigte Gruppen hinzufügen:

```
usermod -aG wheel, audio, video, storage edi
```

Die sudoers-Datei mit nano (statt visudo) aufrufen:

```
EDITOR=nano visudo
```

und darin die folgende Zeile entkommentieren.

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

## Anpassen der Openbox-Menüstruktur

Da das Openbox-Menü statisch erzeugt ist, kann man es entweder von Hand anpassen (und zwar jedesmal, wenn man etwas (de-)installiert), oder man bedient sich eines bereitgestellten Python-Skriptes namens „menumaker“, welches man unter <https://sourceforge.net/projects/menumaker/files/> herunterladen kann. Danach ruft man es wie folgt auf, um sich eine auf das System zugeschnittene Menüstruktur aufbauen zu lassen:

```
$ mmaker openbox -f -t xterm
```

## Anpassen der Autostart-Konfiguration

Es sollten noch einige Anpassungen in der Autostart-Datei des Fenstermanagers gemacht werden, damit man vernünftig arbeiten kann.

Diese liegt in: `~/.config/openbox/autostart`

```
# polkit für sudo Befehle
/usr/lib/mate-polkit/polkit-mate-authentication-agent-1 &
# Bildschirmauflösung
xrandr -s 1920x1080 &
# panel
tint2 &
# Hintergrundbild
nitrogen --restore &
# Systemtray Lautstärkeregler
volumeicon &
```

## Erweiterung des Grundsystems um eine GUI

Das installierte Arch-Grundsystem beinhaltet die standardmäßigen GNU/Linux Werkzeuge. Ansonsten allerdings nicht wirklich viel, womit sich vernünftige (Büro-) Arbeiten erledigen lassen würden.

Alles, was man sonst noch an Anwendungen und Werkzeugen benötigt, die einem das Leben leichter machen, muss man jetzt nachinstallieren.

Beispielsweise fehlt es an einer Grafischen Benutzeroberfläche (GUI) und an Standard-Anwendungen, welche man von „ganzheitlichen“ Distributionen gewohnt ist (Office-Paket, Mailclient, Browser, Multimedia, ...).

## X und Fenstermanager

Für ein grafisches GNU/Linux Erlebnis installieren wir den schlanken und bewährten Fenstermanager „Openbox“ nebst einigen Hilfsprogrammen.

```
pacman -S xorg-server openbox obconf obmenu obkey tint2
  lxappearance gnome-backgrounds volumeicon networkmanager
  mate-polkit
```

Paket	Zweck
xorg-server	Der X-Window-Server, der es gestattet, eine grafische Umgebung zu haben
openbox	Der leichtgewichtige Fenstermanager
obconf	Ein Hilfswerkzeug zur Konfiguration von Openbox
obmenu	Ein Menü-Editor für Openbox
obkey	Grafisches Frontend zur Konfiguration der Tastenkürzel von Openbox
tint2	Eine „Taskleiste“
lxappearance	Ein Themen-Switcher
gnome-backgrounds	Verschiedene Hintergrundbilder
volumeicon	Ein leichtgewichtiges Applet für die Lautstärkekontrolle
networkmanager	Ein Applet zur Netzwerkverwaltung
mate-polkit	Agent zur Kontrolle systemweiter Privilegien

Den oben installierten Netzwerk-Manager muss man noch aktivieren:

```
systemctl enable NetworkManager.service
```

Nach Installation des Fenstermanagers muss man noch einige Nachbereitungen vornehmen, um später die Konfiguration der Openbox für den Benutzer anpassen zu können:

```
mkdir -p /home/edi/.config/openbox
cp /etc/xdg/openbox/rc.xml /home/edi/.config/openbox/
cp /etc/xdg/openbox/menu.xml /home/edi/.config/openbox/
```



Schliesslich wird in der Datei *xinitrc* noch die Openbox als auszuführende Sitzung festgelegt:

```
nano /home/edi/.xinitrc
```

hier die Zeile

```
exec openbox-session
```

einfügen.

## Login-Manager

Um dem Benutzer eine grafische Benutzeroberfläche zur Anmeldung und Bedienung zu geben, installieren wir den „Greeter“ lightdm:

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

Paket	Zweck
lightdm	Ein schlanker Displaymanager (auch Loginmanager oder Anmeldefenster genannt)
lightdm-gtk-greeter	Grafisches Konfigurationswerkzeug für lightdm

Danach muss lightdm noch aktiviert werden:

```
systemctl enable lightdm.service
```

Wenn das abgeschlossen ist, kann man das System einmal neu starten, um zu sehen, ob soweit alles in Ordnung ist, bevor man weitere Pakete installiert:

```
exit  
umount /mnt  
shutdown now -r
```

## Erweiterung des Grundsystems um weitere Werkzeuge und Anwendungen

In diesem Schritt geht es um die Installation einiger weiterer Programme um die grafische Oberfläche „aufzupeppen“ und natürlich Anwendungen für die tägliche Arbeit:

```
pacman -S pcmanfm nitrogen nano neofetch termite  
lxterminal firefox thunderbird libreoffice-fresh conky  
pluma sudo xarchiver unzip unrar tar p7zip p7zip-plugins  
rsync rofi screen
```

Paket	Zweck
pcmanfm	Einfacher aber schneller Dateimanager
nitrogen	Grafisches Hintergrundbildprogramm, das in zwei Modi verwendet werden kann: Browser und Abruf
nano	Einfacher Editor für's Terminal
neofetch	Terminal-Programm zur Anzeige von Systeminformationen
termite	Terminal-Emulator
lxterminal	Terminal-Emulator
firefox	Der beliebte freie Browser
thunderbird	Komfortabler und umfangreicher Mailclient
libreoffice	Vollständiges Office-Paket
conky	Flexibles Tool zur Anzeige von Systeminformationen
pluma	Grafischer Editor
sudo	Ermöglicht berechtigten Benutzern, das Programm im Namen und mit den Rechten eines anderen Benutzers, auch root auszuführen
xarchiver, unzip, unrar, tar, p7zip, p7zip-plugins	(Ent-)packprogramme für alle Bedürfnisse
rsync	Ein Programm, um Dateien zwischen lokalen oder über das Netzwerk erreichbaren Pfaden abzugleichen
rofi	Ein Fensterwechsler und Ausführndialog
screen	Fenstermanager zur Verwendung mit textbasierten Eingabefenstern (Textkonsole)

Für Automatisierungs-Freaks können auch noch folgende Pakete installiert werden:

```
sudo pacman -S yad xdotool
```

## Weitere, optionale Werkzeuge und Konfigurationen

In diesem Abschnitt geht es darum, das System nochmals zu individualisieren und den eigenen Wünschen, Vorstellungen und Bedürfnissen anzupassen.

## Benutzerverzeichnis-Ordner erstellen (optional)

Nach der Installation von Arch Linux gibt es die von anderen Distributionen vertrauten Benutzerverzeichnisse wie Dokumente, Bilder, Videos usw. nicht. Man könnte diese manuell erstellen, aber es gibt einen einfacheren Weg.

Man installiert einfach xdg-user-dirs:

```
sudo pacman -S xdg-user-dirs
```

Dann führt man es aus:

```
xdg-user-dirs-update
```

Nun sind alle gewohnten Benutzerverzeichnisse erstellt.

## Statistische Informationen an das Arch-Projekt zurückmelden

Mit

```
sudo pacman -S pkgstats
```

installiert man ein Paket, welches eine Liste aller installierten Pakete, der Architektur und des Spiegels, der verwendet wird, an das Arch Linux Projekt sendet. Diese Informationen sind anonym und können nicht zur Identifizierung des Benutzers verwendet werden, aber sie helfen den Arch-Entwicklern, ihre Bemühungen zu priorisieren (Quellcode).

## Anpassen der lightdm-Konfiguration / Autologin (optional)

Auch diese Anpassungen erfolgen über eine Konfigurationsdatei.

Diese liegt in `/etc/lightdm/` und muss somit als root bearbeitet werden:

```
sudo nano /etc/lightdm/lightdm.conf
```

Dort unkommentiert man (oder fügt hinzu):

```
[SeatDefaults]
autologin-guest=false
autologin-user=username
autologin-user-timeout=0
```

Dann fügt man den gewünschten Benutzer zur Autologin-Gruppe hinzu:

```
sudo groupadd -r autologin
sudo gpasswd -a username autologin
```

## jgmenu installieren und einbinden (optional)

jgmenu ist ein alternatives Menüsystem für den Openbox Fenstermanager.

Die Installation erfolgt mit:

```
sudo pacman -S jgmenu
```

Es gibt verschiedene Möglichkeiten, das Menü ins System einzubinden um es dann entsprechend bequem aufrufen zu können.

Die händische Ausführung wäre einfach nur der Aufruf im Terminal:

```
jgmenu_run
```

Natürlich kann man den Aufruf auch an eine Tastenkombination binden. Dies erfolgt in der Konfigurationsdatei `~/.config/openbox/rc.xml` der Openbox in der Sektion `<keyboard></keyboard>`:

```
<keybind key="C-Escape">
  <action name="Execute">
    <command>jgmenu_run</command>
  </action>
</keybind>
```

In diesem Beispiel wäre das die Tastenkombination „Strg/Control + Escape“. Die Bindung des Aufrufes an eine Mausaktion erfolgt auf gleichem Wege, allerdings in der Sektion `<mouse><context name="Root"></context></mouse>`:

```
<mousebind button="Right" action="Press">
  <action name="Execute">
    <command>jgmenu_run</command>
  </action>
</mousebind>
```

Was den Aufruf des Menüs auf die rechte Maustaste legen würde.

## Einbindung des Menüs in tint2 (optional)

In der Konfigurationsdatei `jgmenurc`, setzt man die Konfigurationsvariable:

```
tint2_look = 1
```

um `jgmenu` wie `tint2` aussehen zu lassen (z.B. Farbe, Schriftart, margin, padding und Position).

Andere Konfigurationsvariablen in `jgmenurc` überschreiben jegliche `tint2`-Einstellungen, somit ist es geboten, alle anderen Zeilen auskommentiert zu lassen.

Das Button-Plugin wurde für `jgmenu` entwickelt und ist daher die bevorzugte Methode. Um das Schaltflächen-Plugin einzufügen braucht es die folgenden Schritte:

Im Bereich `panel_items` der Datei `tint2rc` ist der Buchstabe „P“ hinzuzufügen:

```
panel_items = PTSBC
```

Und die folgenden Zeilen in der Datei `tint2rc` hinzufügen:

```
button = new
button_icon = /pfad/zu/gewuenshtes_icon.png
button_lclick_command = jgmenu_run
```

Man kann auch ein „exec plugin“ (E) oder „launcher plugin“ (L) verwenden. Hinweise zur Implementierung finden sich in der Manpage von tint2.

Hinweis: Das tint2 „launcher plugin“ ist signifikant langsamer als „button“ und „exec plugins“.

Um die Startbenachrichtigung (drehende Sanduhr nach dem Klicken auf die Schaltfläche oder den Launcher) zu vermeiden, stellt man Folgendes in der Datei `~/.config/tint2/tint2rc` ein:

```
startup_notifications = 0
```

## Weitere Anpassungen der Openbox-Konfiguration (optional)

Möchte man in der Infoleiste der Openbox einen kleinen Kalender haben, muss man sich das Paket „yad“ installieren (siehe Kapitel „Weitere Tools und Anwendungen“).

Dann die Konfigurationsdatei des Benutzers bearbeiten:

```
nano ~/.config/openbox/rc.xml
```

und folgende Zeile ergänzen:

```
yad --calendar --mouse --undecorated --timeout="5"  
    --skip-taskbar
```

## Anpassen der systemweiten Zeitzone (optional)

Um die systemweite Zeitzone zu setzen, lässt man sich die verfügbaren Zeitzonen anzeigen:

```
tzselect
```

Nun hat man einen Zeitzonen-Code für die Datei „.profile“.

In diese Datei wird der Code eingefügt (Datei muss eventuell im home-Verzeichnis des Benutzers angelegt werden)

```
TZ='Europe/Berlin'; export TZ
```

## Codecs, Plugins und Multimedia-Programme (optional)

Um für Unterhaltung in Form von Audio und Video zu sorgen, müssen noch diverse Codecs installiert werden:

```
sudo pacman -S a52dec faac faad2 flac jasper lame libdca  
libdv libmad libmpeg2 libtheora libvorbis libxv wavpack  
x264 xvidcore gstreamer0.10-plugins
```

Oder man installiert den beliebten VLC-Player. Damit werden benötigte Codecs automatisch mitinstalliert:

```
sudo pacman -S vlc
```

Ein weiterer adäquater Musikplayer wäre beispielsweise amarok:

```
sudo pacman -S amarok
```

## Zusätzliche Schriftarten und Rechtschreibprüfungen (optional)

Für eine Auswahl an Schriftarten installiert man folgende Pakete:

```
sudo pacman -S mythes-en ttf-liberation ttf-bitstream-vera  
ttf-droid ttf-dejavu ttf-ubuntu-font-family  
ttf-anonymous-pro enchant languagetool hunspell-de_DE  
aspell-de libmythes
```

## Weitere optionale Schritte

### Installation eines LTS-Kernels

Warum sollte man einen LTS-Kernel in Arch Linux installieren, wenn es doch eigentlich die neueste, „cutting edge“ Version sein soll?

Die Installation eines LTS-Kernels bedeutet, dass man einen stabileren Kernel mit besserer Unterstützung für ältere Hardware hat. Außerdem werden die LTS-Kernel für mindestens 2 Jahre mit Fehlerbehebungen und Leistungsverbesserungen unterstützt. Zwei relativ gute Gründe dafür.

Gegenanzeige wäre Hardware, welche brandneu ist und nur mit dem letzten Kernel funktioniert.

Bevor man einen LTS-Kernel installiert, sollte man mit

```
uname -r
```

oder dem detaillierteren

```
uname -a
```

prüfen, was für einen Kernel man gerade verwendet.  
Die Ausgabe könnte folgendermaßen aussehen:

```
5.15.0-101-generic
```

Die angegebenen Zahlen bedeuten folgendes:

**5** – Kernel-Version, **15** – Hauptrevision, **0** – Unterrevision, **101** – Bugfix

Installiert wird der LTS-Kernel dann mit:

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

## Installation von Microcode für den Prozessor

Der Mikrocode ist ein wesentlicher Bestandteil der Funktionalität des Prozessors. Er muss also sicher und aktuell sein. Um den Mikrocode zu aktualisieren, muss man das Mikrocode-Paket installieren.

Je nach Prozessor installiert man es mit

```
sudo pacman -S amd-ucode # für AMD-Prozessoren  
sudo pacman -S intel-ucode # für Intel-Prozessoren
```

## Firewall einrichten

Für den einfachen Einstieg in dieses zugegebenerweise komplexe Thema kann man die „UFW“ (Uncomplicated Firewall) empfehlen. Diese wird beispielsweise auch in Linux Mint mitgeliefert.

Die Installation erfolgt mit

```
sudo pacman -S ufw
```

Und die Aktivierung mit:

```
sudo ufw enable
```

Überprüfen des Status:

```
sudo ufw status verbose
```

Es sollte aktiv sein und die Standardeinstellungen sind für die meisten Benutzer ausreichend.

Für die Aktivierung der Firewall beim Systemstart:

```
sudo systemctl enable ufw.service
```

Und schon hat man einen Firewall-Schutz.

## Grafischer Paketmanager

Für diesen optionalen Schritt benötigt man zwei Werkzeuge: Yaourt und pamac

### Yaourt

Yaourt kann zum Suchen, Herunterladen und Installieren von Paketen sowohl aus der offiziellen Paketquelle als auch aus AUR (Arch User Repository) verwendet werden.

AUR enthält viele nützliche Anwendungen, die nicht in der Hauptpaketquelle enthalten sind und mit Hilfe von Yaourt aus dem AUR installiert werden können. Yaourt ist vergleichbar mit dem Synaptic Package Manager und kann mit den folgenden Schritten installiert werden:

Zunächst bearbeitet man die Datei `/etc/pacman.conf` und fügt folgende Zeilen am Ende ein:

```
[archlinuxfr]
SigLevel = Never
Server = http://repo.archlinux.fr/$arch
```

Danach kann Yaourt installiert (und mit AUR synchronisiert) werden:

```
sudo pacman -Syu yaourt
yaourt -Syy
```

### pamac

Pamac stellt eine grafische Oberfläche für pacman dar und funktioniert wie der Synaptic Package Manager.

Pamac dient als Oberfläche für die Installation oder Aktualisierung von Paketen und arbeitet gut mit dem Arch User Repository „AUR“ zusammen.

Die Installation ist recht einfach. Zu Beachten ist lediglich, dass vorher yaourt installiert sein muss.

```
yaourt -S pamac-aur
```

Beim ersten Aufruf der Oberfläche sollte man die AUR-Pakete in den Einstellungen noch aktivieren.

## Paketmanager aufpeppen...

### Den schnellsten pacman-Spiegelserver einstellen

pacman wird beim Herunterladen von Paketen manchmal ein wenig langsam. Wenn man ihn richtig konfiguriert, und die richtigen Spiegelserver einbezieht, kann man schnellere Downloadgeschwindigkeiten erreichen.



Die Konfigurationen werden in der Datei `/etc/pacman.d/mirrorlist` gespeichert. Es gibt mehrere Programme, die diese Liste automatisch mit den schnelleren Spiegelservers für den eigenen Standort aktualisieren.

Eines davon ist *reflector*, ein Python-Skript. Es ist einfach zu installieren:

```
sudo pacman -S reflector
```

Natürlich erstellt man eine Sicherungskopie der aktuellen Liste, bevor man es ausführt:

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

Im Terminal führt man dann den folgenden Befehl aus (Land evtl. anpassen):

```
sudo reflector --country 'Germany' --latest 5 --age 2  
--fastest 5 --protocol https --sort rate --save  
/etc/pacman.d/mirrorlist
```

## Farbe für pacman

Standardmäßig zeigt pacman alles in Schwarz und Weiß an. Wir werden das ändern und die Ausgabe von pacman ein wenig farbig gestalten.

Dazu öffnet man die Konfigurationsdatei von pacman:

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

Und entfernt darin das Kommentarzeichen vor der Zeile „color“, um die Ausgabe farbig zu machen. Wer es noch etwas abenteuerlicher möchte, kann auch *IloveCandy* in der Konfigurationsdatei direkt nach der Option „color“ hinzufügen.

## Mehr Geschwindigkeit für pacman

Hat man eine gute Internetverbindung, kann man auch das Kommentarzeichen vor der Zeile „ParallelDownloads“ entfernen, so dass pacman mehrere Pakete gleichzeitig herunterlädt (standardmäßig: 5). Das spart Zeit, wenn das System aktualisiert wird, und wir wollen nicht lügen, mit Arch Linux macht man das sehr oft.

## Automatisches Bereinigen des Paket-Caches

Wie im vorherigen Kapitel erwähnt, lädt man unter Arch sehr häufig Aktualisierungen herunter und stellt schon bald fest, dass das Paket-Cache-Verzeichnis unakzeptabel groß wird.

Überprüfen kann man die Größe des Paket-Caches mit:

```
du -sh /var/cache/pacman/pkg/
```

Das kann man von Zeit zu Zeit manuell machen, aber es ist besser, so etwas zu automatisieren. Das ist mit dem paccache-Skript möglich, welches ihn wöchentlich bereinigt, indem es alte Pakete entfernt und die aktuellste Version jedes Pakets aufbewahrt, für den Fall, dass man einige Pakete downgraden muss.

Installiert wird das Skript mit:

```
sudo pacman -S pacman-contrib
```

Danach aktiviert man den paccache-Timer:

```
sudo systemctl enable paccache.timer
```

Nun wird paccache jede Woche das Paket-Cache-Verzeichnis überprüfen und gegebenenfalls bereinigen.

## Datensicherung und Backup

Vielleicht hätte es das erste sein sollen, was man nach der Installation von Arch Linux macht... Es ist immer gut, ein Backup des Systems und vor Allem seiner Dateien zu haben.

Es gibt viele grafische Programme für GNU/Linux, um ein Backup zu erstellen, aber da wir mit dieser Broschüre Arch Linux-Benutzer geworden sind, wird hier natürlich rsync empfohlen...

Es ist ein Kommandozeilen-Tool, aber es ist super leistungsstark.

Installiert wird es mit:

```
sudo pacman -S rsync
```

Der folgende Befehl sichert alles, bis auf einige nicht benötigte Dateien:

```
sudo rsync -aAXv --delete --dry-run --exclude=/dev/*
--exclude=/proc/* --exclude=/sys/* --exclude=/tmp/*
--exclude=/run/* --exclude=/mnt/* --exclude=/media/*
--exclude="swapfile" --exclude="lost+found"
--exclude=".cache" --exclude="Downloads"
--exclude=".VirtualBoxVMs" --exclude=".ecryptfs"
/ /mnt/backup_ziel/
```

## GRUB-Verzögerung deaktivieren

Um den Bootvorgang zu beschleunigen, kann man den GRUB-Bildschirm, der das GRUB-Menü mit einem 5-Sekunden-Countdown anzeigt, deaktivieren und den Bootvorgang sofort starten. Sollte man das GRUB-Menü einmal benötigen, kann man es aufrufen, indem man während des Bootens die Esc-Taste gedrückt hält. Um diese Funktion zu aktivieren, öffnet man die GRUB-Konfiguration:

```
sudo nano /etc/default/grub
```

Dort wird die Einstellung

```
GRUB_TIMEOUT_STYLE=hidden
```

hinzugefügt.

Die GRUB-Konfiguration wird folgendermaßen aktualisiert:

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

Ab jetzt wird das System 5 Sekunden schneller booten.

## Weitere Informations- und Ideenquellen

### Die Communities

Natürlich stehen die Offenburger Communities gern bereit, wenn es um Fragen oder Anwendungsprobleme geht.  
Gerne bei unserem regelmäßigen Treffen oder im Offenen Computerraum (TiP).

### Internetquellen

[https://de.wikipedia.org/wiki/Arch\\_Linux](https://de.wikipedia.org/wiki/Arch_Linux)  
<https://wiki.archlinux.org/>  
<https://forum.archlinux.de/>

## Schlussbetrachtung und eigene Notizen

Die Zeiten, in denen ein echter Nerd seinen Kernel noch selbst kompilieren musste, sind lange vorbei.

Allerdings muss man sagen, wenn man GNU/Linux und all die wichtigen technischen Hintergründe wirklich verstehen möchte, ist die Installation „von der Pike auf“ eine unschätzbare Möglichkeit sich sein eigenes System von Anfang an selbst zusammenzustellen und enorm viel dabei zu lernen.

Und wer glaubt, dass die Installation von Arch kompliziert, nervenaufreibend und umständlich ist, der sollte sich mal LFS anschauen... ;-)

