



GNUSlashLinux

Das vollständige Systemhandbuch & Referenzwerk

Ein modulares, keyboard-driven Betriebssystem auf Arch-Basis
mit `niri` WM, `Noctalia` Shell, `Helix`, `Yazi` und performanten CLI-Tools.

Alien-Tec

<https://alien-tec.com>

Inhaltsverzeichnis

1	Systemarchitektur & Philosophie	3
1.1	Die Unix-Philosophie	3
1.2	Die Arch Linux Basis	3
2	Installation & Erste Schritte	4
2.1	Der Live-Modus & Start des Installers	4
2.2	Erste Konfiguration nach dem Booten	4
3	niri WM & Noctalia Shell Referenz	5
3.1	Vollständige Tastenkombinationen (Keybindings aus <code>keybinds.kdl</code>)	5
3.1.1	System-, Shell- & App-Launcher	5
3.1.2	Fenster-, Spalten- & Layout-Navigation	5
3.1.3	Sizing & Formatierung von Spalten / Fenstern	6
3.1.4	Workspaces & Multimonitor-Steuerung	6
3.1.5	Screenshots, Audio, Display & Session-Management	6
4	Kitty Terminal & Yazi File Manager	8
4.1	Kitty Terminal Referenz	8
4.2	Yazi TUI File Manager Referenz	8
5	Helix Text Editor (hx)	9
5.1	Grundlegende Modi & Navigation	9
5.2	Bearbeitung, Suchen & Mehrfach-Cursor	9
5.3	Das Space-Menü & LSP-Integration	9
6	Moderne CLI-Befehlsreferenz	11
6.1	Dateisystem & Navigation	11
6.2	Paketmanagement & Systemwartung	11
7	Hauseigene Skripte & TUI-Helfer	12
7.1	Übersicht der Systemskripte	12
7.1.1	<code>git-sync</code> – Interaktiver Git-Manager	12
7.1.2	<code>hx-open</code> – Workspace- & Konfigurations-Launcher	12
7.1.3	<code>media-fire</code> – TUI Medien-Konverter	12
7.1.4	<code>pass-manager</code> – Password Store Migration & Backup	12
7.1.5	<code>pkexec-wl</code> – Wayland Root-App Launcher	13
7.1.6	<code>smart-fire</code> – Interactive Find & Replace	13
7.1.7	<code>toggle-language</code> – System-Sprachkonfigurator	13
7.1.8	<code>toggle-shell</code> – Standard-Shell Konfigurator	13
8	Backup & Disaster-Recovery	14

8.1	Erstellung eines System-Snapshots	14
8.2	Wiederherstellung mit Ventoy	14
9	Vollständige Paketübersicht (System-Manifest)	15
9.1	Systemkern, Boot & Hardware	15
9.2	Benutzeroberfläche & Desktop (Wayland)	15
9.3	Shell & CLI-Werkzeuge (Kommandozeile)	16
9.4	Netzwerk, Bluetooth & Audio	16
9.5	Medien, Dokumente & Anwendungen	16
9.6	Systemwartung, Live-ISO & Backup-Werkzeuge	17
9.7	Entwicklung, Editoren & Paketquellen	17
9.8	Dateimanager, Dienstprogramme & Archivierung	18
10	Fehlerbehebung & Troubleshooting (FAQ)	19
11	Wichtige Links & Ressourcen	20

Kapitel 1

Systemarchitektur & Philosophie

GNUSlashLinux ist keine gewöhnliche Linux-Distribution. Es wurde konzipiert für Power-User, Softwareentwickler und Systemadministratoren, die maximale Kontrolle, kompromisslose Performance und eine effiziente, tastaturgesteuerte Arbeitsumgebung (Keyboard-Driven Workflow) verlangen.

Die Unix-Philosophie

Traditionelle Desktop-Umgebungen (wie GNOME oder KDE Plasma) setzen oft auf ressourcenhungrige Hintergrunddienste, integrierte Monolithen und schwergewichtige Applikations-Suiten. GNUSlashLinux bricht radikal mit diesem Ansatz:

- **Modularität:** Jede Komponente (Fenstermanager, Statusleiste, Terminal, Dateimanager, Editor) ist ein eigenständiges, hochoptimiertes Werkzeug, das über standardisierte Protokolle (Wayland, IPC, CLI-Pipes) kommuniziert.
- **Asynchrones CLI-Toolset:** Veraltete POSIX-Utilities wurden konsequent durch moderne, in Rust geschriebene Alternativen ersetzt (`eza`, `ripgrep`, `zoxide`, `dust`, `duf`), die Multithreading und Caching von Haus aus unterstützen.
- **Wayland-First mit niri:** Statt veralteter X11-Strukturen kommt ein moderner, horizontal scrollender Tiling-Fenstermanager zum Einsatz, der Fenstersitzungen perfekt strukturiert, ohne dass man sich um starre Grid-Raster kümmern muss.

Die Arch Linux Basis

Durch die Verwendung von Arch Linux als Unterbau profitiert GNUSlashLinux von:

- **Rolling-Release-Modell:** Keine Versionssprünge oder Neuinstallationen mehr. Das System ist immer auf dem exakten Stand der Upstream-Entwicklung.
- **Dem AUR (Arch User Repository):** Zugriff auf das weltweit größte Community-Repräsentatorium für Linux-Software.
- **Kompilierungsfreiheit:** Pakete werden exakt für die lokale Architektur optimiert.

Kapitel 2

Installation & Erste Schritte

GNUSlashLinux wird als Live-ISO ausgeliefert, die sowohl zum Testen als auch direkt für die dauerhafte Installation auf physischen Systemen oder VMs verwendet werden kann.

Der Live-Modus & Start des Installers

Nach dem erfolgreichen Bootvorgang landet der Anwender direkt in der Standardumgebung, gesteuert durch `niri` und die `Noctalia Shell`.

- | | |
|------------------------------|--|
| 1. Terminal öffnen | Drücke die Tastenkombination <code>Mod</code> + <code>Enter</code> , um das Kitty Terminal zu starten. |
| 2. Installer aufrufen | Gib den Befehl <code>minstall-launcher</code> ein und bestätige mit <code>Enter</code> . |
| 3. Partitionierung | Wähle im grafischen Assistenten das Ziel-Laufwerk aus (Unterstützung für ext4, btrfs mit Subvolumes und LUKS-Verschlüsselung). |
| 4. Benutzerkonto | Lege den Hauptbenutzer, das Passwort sowie den Rechnernamen (Hostname) fest. |
| 5. Abschluss | Nach dem Durchlaufen der Paketinstallation fordert der Installer zum Neustart auf. |

Notfall-Fallback: Sollte der grafische Installationsassistent aufgrund von Display-Server-Problemen im Live-System klemmen, lässt sich der Installer mit Root-Rechten und direktem Wayland-Kontext erzwingen über:
`pkexec-wl minstall`

Erste Konfiguration nach dem Booten

Nach dem ersten Systemstart empfiehlt sich die Prüfung der Systemaktualität bequem über das im System hinterlegte Alias:

```
update
```

(Dieses Alias ersetzt den Befehl `sudo pacman -Syu` vollautomatisch). Anschließend können über die Noctalia-Einstellungen (`Mod` + `,`) Monitoreinstellungen, Audio-Routing und Design-Themes angepasst werden.

Kapitel 3

niri WM & Noctalia Shell Referenz

Der Fenstermanager **niri** organisiert Fenster in horizontal aneinandergereihten Spalten (*Columns*). Das bedeutet, dass neue Fenster sich logisch in den horizontalen Stream einfügen.

Vollständige Tastenkombinationen (Keybindings aus `keybinds.kdl`)

System-, Shell- & App-Launcher

Mod + Space	App-Launcher umschalten (<code>noctalia msg panel-toggle launcher</code>)
Mod + S	Noctalia Control Center umschalten (<code>noctalia msg panel-toggle control-center</code>)
Mod + ,	Noctalia Einstellungen öffnen (<code>noctalia msg settings-toggle</code>)
Mod + Alt + H	Hotkey-Overlay (Tastenkürzel-Übersicht) anzeigen
Mod + Enter	Kitty Terminal starten
Mod + Shift + Enter	Helium Browser öffnen
Mod + STRG + Enter	DoubleCommander Dateimanager öffnen
Mod + E	Helix Editor im Terminal öffnen
Mod + Y	Yazi Dateimanager im Terminal öffnen
Mod + M	Midnight Commander im Terminal starten
Mod + B	btcp++ System-Monitor starten

Fenster-, Spalten- & Layout-Navigation

Mod + D	Übersicht / Overview umschalten (<code>toggle-overview</code>)
Mod + Shift + Q	Aktives Fenster schließen
Mod + ← / → (bzw. H / L)	Fokus zur linken / rechten Spalte wechseln
Mod + ↑ / ↓ (bzw. K / J)	Fokus innerhalb der Spalte nach oben / unten bewegen
Mod + STRG + ← / → (H / L)	Aktive Spalte nach links / rechts verschieben
Mod + STRG + ↑ / ↓ (K / J)	Aktives Fenster innerhalb der Spalte nach oben / unten verschieben
Mod + Home / End	Fokus zur ersten / letzten Spalte bewegen
Mod + STRG + Home / End	Aktive Spalte an erste / letzte Position verschieben
Mod + [/]	Fenster in Spalte aufnehmen / verdrängen (<code>consume-or-expel-window</code>)

Mod + W	Spalten-Tabbed-Anzeige	umschalten
	(toggle-column-tabbed-display)	
Mod + V	Fenster Floating-Modus umschalten	
Mod + Shift + V	Fokus zwischen Floating und Tiling wechseln	

Sizing & Formatierung von Spalten / Fenstern

Mod + F	Aktive Spalte maximieren (maximize-column)
Mod + Shift + F	Vollbildmodus (Fullscreen) umschalten
Mod + STRG + F	Spalte auf verfügbare Breite erweitern
Mod + C	Aktive Spalte zentrieren
Mod + STRG + C	Alle sichtbaren Spalten zentrieren
Mod + R / Mod + Shift + R	Vordefinierte Spaltenbreiten durchschalten / zurück
Mod + STRG + Shift + R	Vordefinierte Fensterhöhen durchschalten
Mod + STRG + R	Fensterhöhe zurücksetzen
Mod + - / Mod + =	Spaltenbreite um 10% verkleinern / vergrößern
Mod + Shift + - / Mod + Shift + =	Fensterhöhe um 10% verkleinern / vergrößern

Workspaces & Multimonitor-Steuerung

Mod + PageDown / PageUp	Workspace nach unten / oben wechseln
(U / I)	
Mod + STRG + PageDown / PageUp (U/I)	Aktive Spalte auf Workspace unten / oben verschieben
Mod + Shift + PageDown / PageUp (U/I)	Workspace selbst nach unten / oben bewegen
Mod + 1 ... 9	Direkt zu Workspace 1 bis 9 wechseln
Mod + STRG + 1 ... 9	Aktive Spalte zu Workspace 1 bis 9 verschieben
Mod + Shift + ← / → / ↑ / ↓	Fokus auf Monitor links / rechts / oben / unten wechseln
Mod + Shift + STRG + Arrow	Spalte auf Monitor links / rechts / oben / unten verschieben
Mod + WheelScrollDown / Up	Workspace wechseln (Scroll-Cooldown: 150ms)
Mod + STRG + WheelScrollDown / Up	Spalte auf Workspace nach unten / oben verschieben
Mod + WheelScrollRight / Left	Spaltenfokus nach rechts / links bewegen
Mod + STRG + WheelScrollRight / Left	Spalte nach rechts / links verschieben

Screenshots, Audio, Display & Session-Management

Print / STRG + Print / Alt + Print	Screenshot aufnehmen (Auswahl / Bildschirm / Fenster)
Super + Alt + L	Bildschirm sperren (noctalia msg session lock)
Mod + Shift + X	Noctalia Session-Menü öffnen
Mod + Shift + E / STRG + Alt + Del	Sitzung beenden / Logout

Mod + STRG + P	Monitore ausschalten (power-off-monitors)
Mod + Shift + P	System herunterfahren (shutdown)
Mod + Escape	Tastatur-Shortcuts sperren / entsperren (toggle-keyboard-shortcuts-inhibit)
XF86AudioRaiseVolume / LowerVolume / Mute	Audio Lauter / Leiser / Stummschalten
XF86MonBrightnessUp / Down	Bildschirmhelligkeit erhöhen / verringern

Kapitel 4

Kitty Terminal & Yazi File Manager

Die Kombination aus Kitty als GPU-beschleunigtem Terminal und Yazi als blitzschnellem TUI-Dateimanager bildet das Herzstück der CLI-Arbeit.

Kitty Terminal Referenz

Kitty nutzt die GPU zum Rendern von Schriften und Grafiken, was extrem flüssiges Scrollen und die direkte Vorschau von Bildern im Terminal ermöglicht.

STRG + Shift + C	Markierten Text in Zwischenablage kopieren
STRG + Shift + V	Text aus Zwischenablage einfügen
STRG + Shift + N	Neues Kitty-Fenster im aktuellen Arbeitsverzeichnis öffnen
STRG + Shift + Enter	Kitty-Fenster im aktuellen Layout splitten
STRG + Shift + +	Schriftgröße im Terminal vergrößern
STRG + Shift + -	Schriftgröße im Terminal verkleinern
STRG + Shift + O	Schriftgröße auf Standardwert zurücksetzen
STRG + Shift + T	Neuen Tab im Terminal öffnen
STRG + Shift + W	Aktiven Tab / Fenster schließen
STRG + Tab	Zum nächsten Tab wechseln

Yazi TUI File Manager Referenz

Yazi ist in Rust geschrieben, asynchron aufgebaut und bietet integrierte Bild- und Dokumentenvorschauen.

h / l	Verzeichnishistorie zurück / In Ordner hineingehen
j / k	Cursor nach unten / oben bewegen
g g / G	Zum Anfang / Ende der Dateiliste springen
Enter	Datei mit Standard-Handler öffnen
Space	Datei für Batch-Operationen markieren
STRG + A	Alle Dateien im Ordner auswählen
y / x / p	Ausgewählte Dateien Kopieren / Ausschneiden / Einfügen
d	Datei in den Papierkorb verschieben
r	Datei oder Ordner umbenennen
f	Fuzzy-Suche im Ordner via fzf
/	Inkrementelle Textfilterung
z	Schnelles Springen über zoxide -Historie
q	Yazi beenden

Kapitel 5

Helix Text Editor (hx)

Helix ist ein moderner Modal-Editor ähnlich wie Neovim, setzt jedoch im Gegensatz zu diesem auf ein konsequentes **Selection** → **Action**-Prinzip und bringt von Haus aus integrierten LSP-Support (Language Server Protocol) ohne umständliche Plugin-Konfigurationen mit.

Grundlegende Modi & Navigation

<code>i</code>	In den Einfügemodus (Insert) vor dem Cursor wechseln
<code>a</code>	In den Einfügemodus nach dem Cursor wechseln
<code>v</code>	In den visuellen Auswahlmodus (Select) wechseln
<code>Esc</code>	Zurück in den Normalmodus / Auswahl aufheben
<code>h</code> / <code>j</code> / <code>k</code> / <code>l</code>	Cursor links / runter / hoch / rechts bewegen
<code>w</code> / <code>b</code> / <code>e</code>	Wort vorwärts / rückwärts / zum Wortende springen
<code>f</code> <char>	Auf das nächste Vorkommen von <char> springen
<code>t</code> <char>	Bis vor das nächste Vorkommen von <char> springen
<code>g g</code> / <code>G</code>	Zur ersten / letzten Zeile der Datei springen
<code>g h</code> / <code>l</code>	Zum Zeilenanfang / Zeilenende springen

Bearbeitung, Suchen & Mehrfach-Cursor

<code>d</code>	Aktuelle Selektion löschen (Yank in Register)
<code>c</code>	Selektion löschen und direkt in den Einfügemodus wechseln
<code>y</code> / <code>p</code>	Selektion kopieren (Yank) / Einfügen (Paste)
<code>u</code> / <code>U</code>	Letzte Änderung rückgängig machen / wiederholen
<code>/</code>	In der Datei nach einem Suchbegriff suchen
<code>n</code> / <code>N</code>	Nächstes / vorheriges Suchergebnis anspringen
<code>*</code>	Aktuelles Wort als Suchbegriff übernehmen
<code>s</code>	Selektion anhand eines regulären Ausdrucks aufteilen
<code>C</code>	Cursor auf die Zeile darunter duplizieren (Multi-Cursor)
<code>Alt</code> + <code>C</code>	Cursor auf die Zeile darüber duplizieren

Das Space-Menü & LSP-Integration

<code>Space</code> + <code>f</code>	Projektweiter Datei-Finder (nutzt <code>ripgrep</code>)
<code>Space</code> + <code>b</code>	Geöffnete Buffer (Dateien) anzeigen und wechseln

Space + **k**

LSP-Dokumentation (Hover-Info) für das Objekt unter dem Cursor

Space + **r**

Variable oder Funktion projektweit umbenennen (LSP Rename)

Space + **d**

Code-Fehler und Warnungen (Diagnostics) auflisten

Space + **g**

Git-Diff-Änderungen einsehen

g **d**

Zur Definition der Funktion / Klasse springen (LSP)

g **y**

Zum Typ der Variable springen

:w / **:q** / **:wq**

Datei Speichern / Schließen / Speichern & Schließen

Kapitel 6

Moderne CLI-Befehlsreferenz

GNUSlashLinux ersetzt antiquierte GNU-Befehle durch moderne Implementierungen. Diese Sektion dient als kompakte Befehlsreferenz für den täglichen Systemalltag.

Dateisystem & Navigation

<code>ls</code>	Standardliste mit Dateityp-Icons und Farbmarkierungen
<code>ll</code>	Erweiterte Langform (<code>eza</code>): Zeigt Berechtigungen, Owner, Git-Status
<code>lt</code>	Verzeichnisbaum (<code>eza --tree</code>) bis Ebene 2
<code>z</code> <ordner>	Blitzschneller Ordnerwechsel über Häufigkeitsindex (<code>zoxide</code>)
<code>zi</code>	Interaktives <code>zoxide</code> -Auswahlmenü via Fuzzy-Finder
<code>rg</code> <text>	Hochgeschwindigkeits-Volltextsuche in Dateien (<code>ripgrep</code>)
<code>df</code>	Festplattenübersicht als grafisches Dashboard (<code>duf</code>)
<code>du</code> / <code>dut</code>	Speicherfresser aufspüren als Balkendiagramm (<code>dust</code>)
<code>btop</code>	Interaktiver System-Monitor für CPU, RAM, GPU und Netzwerk

Paketmanagement & Systemwartung

Da GNUSlashLinux auf Arch basiert, kommt der Paketmanager `pacman` zusammen mit AUR-Helfern zum Einsatz. Um den Workflow zu vereinfachen, stellt das System vordefinierte Aliase bereit:

<code>update</code>	Vollständiges System-Update (<code>sudo pacman -Syu</code>)
<code>install</code> <pkg>	Paket installieren und Abhängigkeiten auflösen (<code>sudo pacman -S</code>)
<code>remove</code> <pkg>	Paket samt nicht mehr benötigter Abhängigkeiten löschen (<code>sudo pacman -Rns</code>)
<code>cleanup</code>	Paket-Cache bereinigen, um Festplattenspeicher freizugeben (<code>sudo pacman -Scc</code>)

Kapitel 7

Hauseigene Skripte & TUI-Helfer

GNUSlashLinux stellt eine Reihe vorgefertigter, interaktiver Skripte bereit, die komplexe System- und Administrationstasks über visuelle TUI-Oberflächen (`gum` und `fzf`) vereinfachen. Alle Skripte verfügen über eine automatische zweisprachige Spracherkennung (Deutsch / Englisch) und garantieren durch die Bereinigung der Terminal-History maximale Sicherheit.

Übersicht der Systemskripte

`git-sync` – Interaktiver Git-Manager

Ein visueller Workflow-Assistent für Git-Repositories.

- **Funktion:** Prüft den Git-Status, stellt eine interaktive Dateiauswahl zur Verfügung, fragt die Commit-Nachricht ab und führt bei vorhandenem Remote-Repository automatisch ein `git push` durch.
- **Highlights:** Nutzt `gum choose` mit Multi-Selektion, generiert automatische Zeitstempel-Commits als Fallback und bereinigt den Terminal-Buffer zuverlässig.

`hx-open` – Workspace- & Konfigurations-Launcher

Schnellstarter zum Bearbeiten von Systemkonfigurationen und eigenen Projekten im Helix-Editor.

- **Funktion:** Bietet Direktzugriff auf wichtige Systemdateien (`config.fish`, `.bashrc`, Niri WM Configs) oder durchsucht Projekte via `fzf` und öffnet die gewählte Datei direkt in Helix.
- **Highlights:** Nutzt Highspeed-Dateisuche mit `fd` (oder `find` als Fallback).

`media-fire` – TUI Medien-Konverter

Ein interaktiver All-in-One-Konverter und Downloader für Bild-, Audio- und Videodateien.

- **Funktion:** Ermöglicht das Herunterladen von Medien über URLs (z. B. via `yt-dlp`) wahlweise als Video oder reine MP3-Tonspur. Lokal steuert das Skript Formatkonversionen, Skalierungen und Metadaten-Bereinigungen (`EXIF`) für Bilder via ImageMagick sowie das verlustfreie Schneiden, Konvertieren und Extrahieren von Video- und Audiotracks via `ffmpeg`.
- **Highlights:** Nutzt `fd` und `fzf` für eine schnelle lokale Dateiauswahl und bietet eine dynamische, auf den jeweiligen MIME-Type angepasste TUI-Menüführung.

`pass-manager` – Password Store Migration & Backup

Ein vollautomatisches Backup- und Migrationswerkzeug für den Standard-GPG-Passwort-Manager `pass`.

- **Funktion:** Exportiert lokal alle verschlüsselten Passwörter temporär in eine Klartextdatei für den Umzug auf einen neuen Rechner. Auf dem Zielsystem unterstützt das Skript bei der Erstellung eines neuen GPG-Schlüssels, initialisiert `pass` neu, re-verschlüsselt alle Einträge und löscht die Importdatei sicher via `shred`.

`pkexec-wl` – Wayland Root-App Launcher

Sicherer Launcher zur Ausführung von GUI-Anwendungen mit Administratorrechten unter Wayland.

- **Funktion:** Listet verfügbare Desktop-Anwendungen via `gum filter` auf und führt sie nach Sudo/Polkit-Authentifizierung mit den passenden XWayland/Display-Umgebungsvariablen aus.

`smart-fire` – Interactive Find & Replace

Interaktives Werkzeug zum Suchen und Ersetzen von Text oder RegEx in Dateibäumen.

- **Funktion:** Verbindet `ripgrep/grep` mit `fzf`-Vorschauen. Erlaubt das Eingrenzen des Scopes (alle Treffer vs. gefilterte Auswahl) und führt gezielte `sed`-Ersetzungen nach Bestätigung durch.

`toggle-language` – System-Sprachkonfigurator

Verwaltung der System-Locale, Tastaturbelegung und Papiergrößen.

- **Funktion:** Schaltet das Gesamtsystem bequem zwischen Deutsch (`de_DE.UTF-8`, Tastatur `de`, Papier `A4`) und Englisch (`en_US.UTF-8`, Tastatur `us`, Papier `Letter`) um. Passt `localectl`, `vconsole.conf` sowie Live-System-Vorlagen an.

`toggle-shell` – Standard-Shell Konfigurator

Interaktives Umschalten der Standard-Login-Shell des Benutzers.

- **Funktion:** Ermöglicht den nahtlosen Wechsel der Standard-Shell zwischen `bash` und `fish` mittels `chsh` und prüft vorab den aktuellen Status.

Kapitel 8

Backup & Disaster-Recovery

Ein produktives System erfordert eine zuverlässige und unkomplizierte Backup-Strategie. GNUSlashLinux setzt hierbei auf vollständige System-Snapshots, die im laufenden Betrieb erstellt werden können.

Erstellung eines System-Snapshots

1. Öffne den Anwendungs-Launcher (**Mod** + **Space**) und starte das Tool **MX-Schnappschuss** (bzw. das entsprechende Backup-Utility).
2. Das Zielverzeichnis ist standardmäßig `/home/snapshot/` (alternativ kann auch eine externe Festplatte oder Partition gewählt werden).
3. Das Tool erzeugt im laufenden Betrieb eine vollständige `.iso`-Sicherungsdatei, die sämtliche installierten Pakete, Systemkonfigurationen (*Dotfiles*), Benutzerdaten und Einstellungen enthält.

Wiederherstellung mit Ventoy

1. Kopiere die erstellte Sicherungs-`.iso` per Drag & Drop auf einen mit **Ventoy** vorbereiteten USB-Rettungsstick.
2. Boote den Zielrechner von diesem USB-Stick.
3. Starte innerhalb der Live-Umgebung das Terminal und führe den Befehl `minstall-launcher` aus.
4. Das gesamte System wird innerhalb weniger Minuten exakt in dem Zustand wiederhergestellt, in dem der Snapshot erstellt wurde.

Kapitel 9

Vollständige Paketübersicht (System-Manifest)

Diese Übersicht dokumentiert die wichtigsten Pakete der vorliegenden GNUSlashLinux-Installation, aufgeteilt nach Systembereichen, Funktionalitäten und Aufgaben.

Systemkern, Boot & Hardware

Diese Komponente stellt die Basis des Betriebssystems, die Hardware-Erkennung und den Kernel zur Verfügung.

<code>base</code>	Essenzielle Grundpakete, die für jede GNUSlashLinux-Installation erforderlich sind.
<code>linux</code>	Der standardmäßige Linux-Kernel für die Ausführung des Betriebssystems.
<code>linux-firmware</code>	Treiber-Firmware-Paket für Netzwerkkarte, Grafikchips und weitere Hardware.
<code>intel-ucode / amd-ucode</code>	Mikrocode-Updates zur Behebung von CPU-Bugs auf Intel- bzw. AMD-Prozessoren.
<code>btrfs-progs / xfsprogs</code>	Werkzeuge zur Erstellung, Wartung und Reparatur von Btrfs- und XFS-Dateisystemen.
<code>efibootmgr</code>	Verwaltung von UEFI-Boot-Einträgen direkt aus dem System heraus.
<code>mkinitcpio-archiso</code>	Erstellung der Initial RAM Disk für Live-Systeme und Installationsmedien.
<code>lshw / hwinfo</code>	Umfassende Werkzeuge zur Hardware-Erkennung und Systemanalyse.

Benutzeroberfläche & Desktop (Wayland)

Programme, die für die grafische Darstellung, den Window-Manager und das visuelle Erscheinungsbild zuständig sind.

<code>niri</code>	Ein scrollbarer Wayland-Tiling-Window-Manager als Hauptfenstermanager.
<code>noctalia</code>	Eine erweiterte Desktop-Komponente bzw. Shell-Ergänzung für die grafische Benutzeroberfläche.
<code>sddm</code>	Moderner Display- und Anmeldebildschirm für grafische Sitzungen.

<code>kitty</code>	Ein extrem schneller, GPU-beschleunigter Terminal-Emulator mit umfassenden Anpassungsmöglichkeiten.
<code>wayland-utils</code> / <code>xwayland-satellite</code>	Hilfsprogramme und Kompatibilitätsschichten für Wayland- und X11-Anwendungen.
<code>nwg-look</code>	Konfigurationswerkzeug für GTK-Themes in Wayland-Umgebungen.
<code>papirus-icon-theme</code>	Umfangreiches, modernes Icon-Theme für den Desktop.

Shell & CLI-Werkzeuge (Kommandozeile)

Moderne und erweiterte Kommandozeilen-Tools, die den täglichen Workflow im Terminal erleichtern.

<code>fish</code> / <code>fisher</code>	Eine alternative, sehr benutzerfreundliche Shell inklusive Plugin-Manager.
<code>starship</code>	Minimalistischer, anpassbarer und performanter Shell-Prompt.
<code>bttop</code>	Moderner, ressourcenorientierter Konsolen-Monitor für CPU, RAM und Prozesse.
<code>gdu</code> / <code>duf</code>	Schnelle, interaktive Festplatten- und Speicherplatzanalysten.
<code>eza</code> / <code>bat</code> / <code>fd</code> / <code>ripgrep</code>	Modernisierte Alternativen zu klassischen Unix-Befehlen (<code>ls</code> , <code>cat</code> , <code>find</code> , <code>grep</code>).
<code>fzf</code> / <code>zoxide</code>	Fuzzy Finder für Dateien und blitzschnelle Verzeichnisnavigation.
<code>tmux</code>	Terminal-Multiplexer zur Verwaltung mehrerer Sitzungen in einem Fenster.

Netzwerk, Bluetooth & Audio

Verwaltung von Netzwerkschnittstellen, drahtloser Kommunikation und dem Audiosystem.

<code>networkmanager</code>	Umfassende Verwaltung von WLAN-, LAN- und VPN-Verbindungen.
<code>bluez</code> / <code>bluez-utils</code>	Bluetooth-Protokollstapel und Steuerungswerkzeuge für Funkverbindungen.
<code>ufw</code>	Einfache Firewall-Verwaltung (Uncomplicated Firewall) zur Systemsicherheit.
<code>pipewire-alsa</code> / <code>pipewire-jack</code>	Modernes, universelles Audio-Subsystem für Sound- und Medienwiedergabe.
<code>openssh</code>	Sichere Fernwartung und verschlüsselte Netzwerkkommunikation via SSH.

Medien, Dokumente & Anwendungen

Programme zur Wiedergabe von Medien, Betrachten von Dokumenten, fürs Web und den alltäglichen Gebrauch.

<code>mpv</code>	Leistungsstarker und modularer Medienplayer für Audio und Video.
<code>yt-dlp</code>	Kommandozeilen-Werkzeug zum Herunterladen von Videos von verschiedenen Plattformen.
<code>zathura</code>	Extrem modularer, tastaturgesteuerter Dokumenten- und PDF-Viewer.
<code>pavucontrol</code>	Grafische Lautstärkeregler-Steuerung für einzelne Audio-Streams.
<code>fontconfig</code> / <code>ttf-nerd-fonts-symbols</code>	Schriftarten-Verwaltung und universelle Symbol-Fonts für Entwickler-Shells.
<code>helium-browser-bin</code>	Ein alternativer Webbrowser auf Chromium-Basis.

Systemwartung, Live-ISO & Backup-Werkzeuge

Spezielle Werkzeuge für Systemüberprüfung, Installation, ISO-Erstellung und Live-Snapshots.

<code>mx-snapshot</code>	Mächtiges Werkzeug zur Erstellung eines Live-ISOs / System-Backups des laufenden Systems.
<code>gazelle-installer</code> / <code>data-mx</code>	Das Installationsprogramm (samt Daten) zur Einrichtung des Systems auf Festplatten.
<code>smartmontools</code>	Überwachung und Diagnose des Festplatten-Gesundheitsstatus (S.M.A.R.T.).
<code>rsync</code>	Klassisches und mächtiges Werkzeug für schnelle, inkrementelle Datei-Backups.
<code>reflector</code>	Automatische Sortierung und Aktualisierung der schnellsten Server-Mirrors.
<code>testdisk</code>	Datenrettungstools für beschädigte Partitionen und verlorene Dateisysteme.
<code>tlp</code> / <code>power-profiles-daemon</code>	Energie- und Akku-Optimierung speziell für mobile Geräte.

Entwicklung, Editoren & Paketquellen

Pakete für Software-Entwicklung, Kompilierung, Textbearbeitung und die Paketverwaltung.

<code>paru</code>	AUR-Helfer zum komfortablen Suchen und Installieren von Fremdpaketen.
<code>git</code>	Dezentrale Versionskontrolle für Quellcode und Konfigurationsdateien.
<code>gcc</code> / <code>make</code> / <code>patch</code>	Essenzielle Entwicklungswerkzeuge und Compiler zum Bauen von Software.
<code>python-virtualenv</code>	Erstellung isolierter Python-Umgebungen für Entwicklungsprojekte.
<code>helix</code>	Ein moderner, modal-basierter Terminal-Texteditor ("batteries-included").
<code>silicon</code>	Tool zum Erstellen von wunderschönen, grafischen Code-Schnipsel-Bildern.

Dateimanager, Dienstprogramme & Archivierung

Zusätzliche Dateiverwaltungswerkzeuge, Konsolen-Helfer und Komprimierungstools.

<code>yazi</code>	Ein rasanter, terminalbasierter Dateimanager in Rust mit asynchronem I/O und Bildvorschau.
<code>mc</code> (Midnight Commander)	Der klassische, textbasierte Zwei-Fenster-Dateimanager für die Konsole.
<code>doublecmd</code>	Ein schneller, mächtiger, leichtgewichtiger grafischer Dateimanager (Qt-basiert) für den Desktop.
<code>7zip</code> / <code>unzip</code> / <code>unrar</code>	Universelle Kommandozeilen-Tools zum Entpacken und Erstellen von Archiven.
<code>jq</code>	Ein leichtgewichtiger und flexibler JSON-Prozessor für die Kommandozeile.
<code>wl-clipboard</code>	Zwischenablage-Werkzeuge speziell für Wayland-Umgebungen (<code>wl-copy</code> , <code>wl-paste</code>).
<code>plocate</code>	Ein schneller, indexbasierter Dateisucher (<code>locate</code> -Ersatz).

Kapitel 10

Fehlerbehebung & Troubleshooting (FAQ)

Sollte es im täglichen Betrieb zu unerwarteten Problemen kommen, bietet diese Sektion schnelle Lösungsansätze.

Wayland / niri startet nicht	Prüfe, ob die proprietären oder Open-Source GPU-Treiber korrekt geladen sind. Logdateien lassen sich einsehen mit: <code>journalctl -b -e -u niri</code> .
Kein Ton / Audio stumm	Prüfe PipeWire und WirePlumber Status über: <code>systemctl --user status pipewire wireplumber</code> . Neustart des Audio-teilnehmers: <code>systemctl --user restart wireplumber</code> .
AUR-Paket schlägt fehl	Oft ist der lokale Build-Cache beschädigt. Bereinige ihn mit: <code>yay -Syu --cleancache</code> .
Netzwerkverbindung trennt	ge- Starte den NetworkManager neu mit: <code>sudo systemctl restart NetworkManager</code> .

Kapitel 11

Wichtige Links & Ressourcen

Offizielle Projektseiten, Quellcode-Repositorys und Herstellerinformationen:

- **Offizielle Website:** <https://gnuslashlinux.com>
- **SourceForge Projektseite:** <https://sourceforge.net/projects/gnuslashlinux/>
- **GitHub Repository:** <https://github.com/GNUSlashLinux>
- **Entwickler / Alien-Tec:** <https://alien-tec.com>