Linux – Auf ein Neues

Contents

[Aufbau von Linux 2](#_Toc170907183)

[Standardsoftware in Debian/Ubuntu, Fedora und Arch Linux 6](#_Toc170907184)

[Einsatzgebiete von Linux 8](#_Toc170907185)

[Weitere Einsatzbereiche von Linux 8](#_Toc170907186)

# Aufbau von Linux

Linux ist ein Open-Source-Betriebssystem, das aus verschiedenen Komponenten besteht, die zusammenarbeiten, um den Betrieb von Computern zu ermöglichen. Der Aufbau von Linux lässt sich in mehrere Schichten unterteilen, wobei jede Schicht spezifische Aufgaben übernimmt:

1. **Kernel und Systemstart:**
   * **Kernel:**
     + Der Kernel ist das Herzstück von Linux und bildet die Schnittstelle zwischen Hardware und Software. Er verwaltet die Systemressourcen, steuert die Hardware, verwaltet Speicher, führt Prozesse aus und bietet grundlegende Dienste wie Dateisystemzugriff und Netzwerkkommunikation. Der Linux-Kernel ist monolithisch, was bedeutet, dass er alle Kernfunktionen in einem einzigen, großen Kernel integriert. Der Kernel kann modular erweitert werden, wodurch zusätzliche Funktionalitäten als Module geladen werden können, ohne dass ein Neustart erforderlich ist.
   * **Bootloader:**
     + Der Bootloader ist das Programm, das beim Starten des Computers ausgeführt wird und den Kernel lädt. Er ermöglicht es, verschiedene Betriebssysteme auf einem Computer zu starten und bietet oft ein Menü zur Auswahl des gewünschten Systems.
     + Beispiele: GRUB (GRand Unified Bootloader), LILO (Linux Loader).
   * **Init-Systeme:**
     + Init-Systeme sind für den Start und die Verwaltung von Diensten und Prozessen während des Systemstarts verantwortlich. Sie stellen sicher, dass die notwendigen Dienste in der richtigen Reihenfolge gestartet werden.
     + Beispiele: systemd, SysVinit, Upstart.
2. **Systembibliotheken:**
   * Diese Bibliotheken enthalten grundlegende Funktionen und Dienste, die von Anwendungen genutzt werden. Die GNU C Library (glibc) ist ein prominentes Beispiel und stellt Standard-C-Funktionen bereit, die für die Entwicklung von Anwendungen unerlässlich sind. Systembibliotheken bieten eine Schnittstelle zu Kernel-Diensten und stellen Funktionen wie Datei-I/O, Speicherverwaltung und mathematische Operationen bereit.
3. **System- und Dienstprogramme:**
   * Diese Programme sind für die Verwaltung und Konfiguration des Systems zuständig. Sie ermöglichen die Automatisierung von Aufgaben, die Benutzerverwaltung, Netzwerkkonfiguration und Sicherheitsüberwachung.
   * Beispiele: systemd (Init-System und Dienstmanager), cron (Zeitgesteuerte Aufgaben), init (System- und Dienstverwaltung), sudo (Rechteverwaltung), ssh (Sicherer Fernzugriff), iptables (Firewall-Konfiguration).
4. **Shell:**
   * Die Shell ist die Kommandozeilenschnittstelle, die Benutzern und Skripten den Zugriff auf das System ermöglicht. Sie bietet eine Umgebung zur Ausführung von Befehlen, Verwaltung von Dateien und Ausführung von Skripten.
   * Beispiele: Bash (Bourne Again Shell), Zsh (Z Shell), Fish (Friendly Interactive Shell).
5. **Dateisysteme und Speicherverwaltung:**
   * **Dateisysteme:**
     + Linux unterstützt verschiedene Dateisysteme, die für die Organisation und Speicherung von Daten auf Datenträgern zuständig sind. Jedes Dateisystem hat seine eigenen Vor- und Nachteile in Bezug auf Geschwindigkeit, Zuverlässigkeit und Funktionalität.
     + Beispiele: ext4 (Standard-Dateisystem für viele Distributionen), XFS (Hochleistungs-Dateisystem), Btrfs (Btrfs: B-Tree File System, bietet Snapshotting und Checksumming).
   * **Netzwerkdateisysteme:**
     + Technologien zur Freigabe und Verwaltung von Dateisystemen über ein Netzwerk. Sie ermöglichen den Zugriff auf entfernte Dateisysteme, als ob sie lokal wären.
     + Beispiele: NFS (Network File System), Samba (für die Interoperabilität mit Windows-Netzwerken), AFP (Apple Filing Protocol).
   * **Storage-Management:**
     + Tools und Technologien zur Verwaltung von Speicherressourcen, einschließlich logischer Volumes und RAID-Konfigurationen.
     + Beispiele: LVM (Logical Volume Manager, ermöglicht flexible Festplattenverwaltung), RAID-Tools (mdadm, ermöglicht die Konfiguration von RAID-Arrays zur Datenredundanz und Leistungssteigerung).
6. **Paketverwaltung:**
   * Die Paketverwaltungssysteme ermöglichen die Installation, Aktualisierung und Deinstallation von Software. Sie verwalten Softwarepakete und deren Abhängigkeiten, um eine konsistente Systemumgebung zu gewährleisten.
   * Bekannte Paketmanager:
     + Debian/Ubuntu: apt (Advanced Package Tool), dpkg (Debian Package Manager)
     + Fedora/Red Hat: dnf (Dandified Yum), yum (Yellowdog Updater, Modified), rpm (Red Hat Package Manager)
     + Arch Linux: pacman (Package Manager)
7. **Graphische Benutzeroberfläche (GUI):**
   * Die GUI ermöglicht die Verwendung des Systems mit einer visuellen Oberfläche. Sie umfasst verschiedene Komponenten wie Fenstermanager, Panels und Anwendungen.
   * Hauptkomponenten:
     + **X Window System (X11):**
       - Ein altes, aber immer noch weit verbreitetes System für die Bereitstellung von GUI-Funktionalitäten. Es bietet eine client-server-Architektur, die Flexibilität bei der Verwendung von Remote- und lokalen Anwendungen ermöglicht.
     + **Wayland:**
       - Ein modernerer Ersatz für X11, der eine effizientere und sicherere Architektur bietet. Wayland zielt darauf ab, die Schwächen von X11 zu beheben und bietet eine einfachere und leistungsfähigere Rendering-Pipeline.
     + **Desktop-Umgebungen (DEs):**
       - Diese bieten vollständige Benutzeroberflächen mit Fenstermanagern, Panels und Anwendungen. Jede DE hat ihre eigene Philosophie und Benutzererfahrung.
       - Beispiele: GNOME (benutzerfreundlich und modern), KDE Plasma (anpassbar und leistungsstark), Xfce (leicht und ressourcenschonend), LXQt (eine weitere leichte Desktop-Umgebung).
8. **Fenstermanager:**
   * Fenstermanager verwalten die Platzierung und das Erscheinungsbild von Fenstern. Sie sind oft minimalistisch und fokussieren auf die Effizienz.
   * Beispiele: i3 (Tiling-Fenstermanager), Openbox (Stacking-Fenstermanager), awesome (dynamischer Fenstermanager).
9. **Anwendungsprogramme:**
   * Linux bietet eine Vielzahl von Anwendungen, die vom Benutzer installiert und genutzt werden können. Diese Programme decken viele Anwendungsbereiche ab.
   * Beispiele:
     + **Texteditoren:** vim (leistungsfähiger Editor), nano (einfacher Editor), gedit (GNOME-Editor)
     + **Webbrowser:** Firefox (freier Webbrowser), Chrome (Google-Webbrowser)
     + **Büroanwendungen:** LibreOffice (umfangreiche Office-Suite)
     + **Multimedia-Software:** VLC (Vielseitiger Medienplayer), Audacity (Audio-Editor)
     + **Entwickler-Tools:** gcc (GNU Compiler Collection), gdb (GNU Debugger), make (Build-Automatisierung), IDE’s wie Visual Studio Code (Entwicklungsumgebung).
10. **Netzwerk-Tools:**
    * Tools zur Konfiguration und Verwaltung von Netzwerken, wie network-manager (netzwerkübergreifende Verwaltung), ifconfig (Netzwerkschnittstellenkonfiguration), ip (Werkzeug zur Netzwerkverwaltung), nmcli (NetworkManager-Kommandozeilenwerkzeug).
11. **Virtualisierung und Container:**
    * Tools und Technologien zur Erstellung und Verwaltung von virtuellen Maschinen und Containern, die eine isolierte Ausführung von Anwendungen ermöglichen.
    * Beispiele: KVM (Kernel-based Virtual Machine, für Virtualisierung), Docker (für Containerisierung), LXC (Linux Containers, für leichtgewichtige Virtualisierung).
    * **Container-Orchestrierung:**
      + Systeme zur Verwaltung von Containern in großem Maßstab, die Deployment, Skalierung und Betrieb von Containeranwendungen automatisieren.
      + Beispiele: Kubernetes (umfassendes Orchestrierungssystem), Docker Swarm (nativer Orchestrierungsmodus von Docker).
12. **Konfigurationsmanagement:**
    * Software, die zur Automatisierung der Verwaltung und Konfiguration von Computern verwendet wird. Diese Tools helfen dabei, Systeme konsistent und wiederholbar zu konfigurieren.
    * Beispiele: Ansible (agentenloses Konfigurationsmanagement), Puppet (automatisierte Verwaltung), Chef (Infrastruktur als Code), SaltStack (Remote-Ausführungs- und Konfigurationsmanagement).
13. **Protokollierung und Logging:**
    * Tools zur Protokollierung von Systemereignissen und Fehlern, die für die Fehlerbehebung und Überwachung wichtig sind.
    * Beispiele: journald (Teil von systemd für fortschrittliche Protokollierung), rsyslog (erweiterbares Protokollierungssystem), logrotate (Protokollrotation und -verwaltung).
14. **Backup- und Wiederherstellungstools:**
    * Tools zur Sicherung und Wiederherstellung von Daten, um Datenverlust zu vermeiden und Systeme wiederherzustellen.
    * Beispiele: rsync (inkrementelle Dateisynchronisation), Bacula (Netzwerk-Backup-Lösung), Clonezilla (Festplatten- und Partitionssicherung), Timeshift (Schnappschuss-Tool zur Wiederherstellung).
15. **Systemüberwachungs- und Diagnosetools:**
    * Tools zur Überwachung und Analyse der Systemleistung und zur Diagnose von Problemen.
    * Beispiele: top (Echtzeit-Systemüberwachung), htop (interaktiver Prozessmonitor), dstat (Systemressourcenüberwachung), iotop (E/A-Überwachung).
    * **Monitoring und Alerting:**
      + Tools zur Überwachung der Systemleistung und zur Benachrichtigung bei Problemen.
      + Beispiele: Nagios (Überwachung und Alarmierung), Zabbix (Netzwerküberwachung), Prometheus (zeitreihenbasierte Überwachung), Grafana (Dashboard und

Visualisierung).

1. **Sicherheitswerkzeuge:**
   * Tools zur Sicherstellung der Systemsicherheit, einschließlich Zugriffskontrolle und Schutzmechanismen.
   * Beispiele: firewalld (dynamische Firewall-Verwaltung), selinux (Security-Enhanced Linux für Zugriffskontrolle), AppArmor (Sicherheitsframework).
   * **Firewall-Management:**
     + Werkzeuge zur Verwaltung von Netzwerkzugriffsregeln und Firewalls.
     + Beispiele: ufw (Uncomplicated Firewall, einfach zu bedienende Firewall), nftables (neueres Firewall-Framework), iptables (älteres, aber weit verbreitetes Firewall-Framework).
2. **Data Analytics und Log Management:**
   * Systeme zur Analyse und Verwaltung von Protokolldaten, die für die Überwachung und Diagnose von Systemen wichtig sind.
   * Beispiele: ELK Stack (Elasticsearch, Logstash, Kibana, für Protokoll- und Datenanalyse), Graylog (zentrales Log-Management und Analyse).

Zusammen bilden diese Komponenten ein komplettes Ökosystem, das eine flexible und leistungsstarke Umgebung für eine Vielzahl von Anwendungen und Benutzeranforderungen bietet. Linux kann an verschiedene Bedürfnisse angepasst werden, von Server-Umgebungen bis hin zu Desktops für Endanwender.

# Standardsoftware in Debian/Ubuntu, Fedora und Arch Linux

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kategorie** | **Beispiele** | **Debian/Ubuntu** | **Fedora/Red Hat** | **Arch Linux** |
| **Kernel und Systemstart** | **Kernel**: Linux Kernel **Bootloader**: GRUB **Init-System**: systemd | Linux Kernel GRUB systemd | Linux Kernel GRUB systemd | Linux Kernel GRUB systemd |
| **Systembibliotheken** | **GNU C Library (glibc)** | glibc | glibc | glibc |
| **System- und Dienstprogramme** | **systemd**, **cron**, **init**, **sudo**, **ssh**, **iptables** | systemd, cron, sudo, ssh, iptables | systemd, cron, sudo, ssh, iptables | systemd, cron, sudo, ssh, iptables |
| **Shell** | **Bash**, **Zsh**, **Fish** | Bash | Bash | Bash |
| **Dateisysteme und Speicherverwaltung** | **Dateisysteme**: ext4, XFS, Btrfs **Netzwerkdateisysteme**: NFS, Samba **Storage-Management**: LVM, RAID-Tools (mdadm) | ext4 NFS, Samba LVM, mdadm | ext4, XFS NFS, Samba LVM, mdadm | ext4 NFS, Samba LVM, mdadm |
| **Paketverwaltung** | **Debian/Ubuntu**: apt, dpkg **Fedora/Red Hat**: dnf, yum, rpm **Arch Linux**: pacman | apt, dpkg | dnf, rpm | pacman |
| **Graphische Benutzeroberfläche (GUI)** | **X Window System (X11)**, **Wayland** **Desktop-Umgebungen**: GNOME, KDE Plasma, Xfce, LXQt | X11 GNOME, KDE Plasma, Xfce, LXQt | Wayland GNOME, KDE Plasma, Xfce, LXQt | Keine Standard-Desktopumgebung, minimaler Basis-Installationsprozess |
| **Fenstermanager** | **i3**, **Openbox**, **awesome** | Openbox | Keine Standardoption, können aber einfach installiert werden | Keine Standardoption, können aber einfach installiert werden |
| **Anwendungsprogramme** | **Texteditoren**: vim, nano, gedit **Webbrowser**: Firefox, Chrome **Büroanwendungen**: LibreOffice **Multimedia-Software**: VLC, Audacity **Entwickler-Tools**: gcc, gdb, make, Visual Studio Code | nano, gedit Firefox LibreOffice Totem gcc, make | nano, gedit Firefox LibreOffice Totem gcc, make | nano Keine Standardoption für Firefox, LibreOffice, VLC; können manuell installiert werden |
| **Netzwerk-Tools** | **network-manager**, **ifconfig**, **ip**, **nmcli** | network-manager, ifconfig, ip, nmcli | network-manager, ifconfig, ip, nmcli | network-manager, ifconfig, ip, nmcli |
| **Virtualisierung und Container** | **KVM**, **Docker**, **LXC** **Container-Orchestrierung**: Kubernetes, Docker Swarm | KVM, LXC | KVM, LXC | Keine Standardoption, können aber einfach installiert werden |
| **Konfigurationsmanagement** | **Ansible**, **Puppet**, **Chef**, **SaltStack** | Keine Standardoption, können aber einfach installiert werden | Keine Standardoption, können aber einfach installiert werden | Keine Standardoption, können aber einfach installiert werden |
| **Protokollierung und Logging** | **journald**, **rsyslog**, **logrotate** | journald, rsyslog | journald, rsyslog | journald, rsyslog |
| **Backup- und Wiederherstellungstools** | **rsync**, **Bacula**, **Clonezilla**, **Timeshift** | rsync | rsync | rsync |
| **Systemüberwachungs- und Diagnosetools** | **top**, **htop**, **dstat**, **iotop** **Monitoring und Alerting**: Nagios, Zabbix, Prometheus, Grafana | top | top | top |
| **Sicherheitswerkzeuge** | **firewalld**, **selinux**, **AppArmor** **Firewall-Management**: ufw, nftables, iptables | AppArmor ufw, iptables | SELinux firewalld, iptables | Keine Standardoption, können aber einfach installiert werden |

# Einsatzgebiete von Linux

Linux wird auf fast allen Servern eingesetzt aus mehreren Gründen:

1. **Stabilität und Zuverlässigkeit**:
   * Linux ist bekannt für seine Stabilität und Zuverlässigkeit, was besonders in Serverumgebungen wichtig ist, wo kontinuierlicher Betrieb erforderlich ist. Viele Linux-Server laufen jahrelang ohne Neustart oder Absturz.
2. **Sicherheit**:
   * Linux bietet robuste Sicherheitsfunktionen und eine aktive Community, die Sicherheitslücken schnell erkennt und behebt. Features wie SELinux (Security-Enhanced Linux) und AppArmor erhöhen die Systemsicherheit zusätzlich.
3. **Kosten**:
   * Linux ist Open Source und in den meisten Fällen kostenlos. Dies reduziert die Betriebskosten im Vergleich zu proprietären Betriebssystemen wie Windows Server.
4. **Flexibilität und Anpassungsfähigkeit**:
   * Linux kann an spezifische Anforderungen angepasst werden. Administratoren können den Kernel und andere Komponenten modifizieren, um die Leistung und Funktionalität zu optimieren.
5. **Breite Unterstützung durch Software und Tools**:
   * Viele serverseitige Anwendungen, Datenbanken, Webserver (wie Apache und Nginx), und Entwicklerwerkzeuge sind für Linux optimiert und bieten umfangreiche Unterstützung.
6. **Community und Support**:
   * Eine große, aktive Community bietet umfangreiche Ressourcen, Dokumentation und Unterstützung für Linux-Nutzer. Darüber hinaus bieten viele Unternehmen professionellen Support für Linux-Distributionen an (z.B. Red Hat, SUSE).

### Weitere Einsatzbereiche von Linux

1. **Supercomputer**:
   * Linux dominiert die Liste der leistungsstärksten Supercomputer weltweit. Die Flexibilität und Anpassungsfähigkeit von Linux machen es ideal für wissenschaftliche Berechnungen und Hochleistungsrechnen (HPC).
2. **Cloud-Computing**:
   * Die meisten Cloud-Infrastrukturen laufen auf Linux. Dienste wie Amazon Web Services (AWS), Google Cloud Platform (GCP) und Microsoft Azure nutzen Linux für viele ihrer Dienste.
3. **Internet of Things (IoT)**:
   * Linux-basierte Betriebssysteme wie Ubuntu Core und Yocto Project sind in vielen IoT-Geräten zu finden. Sie bieten eine stabile und sichere Plattform für vernetzte Geräte.
4. **Embedded Systems**:
   * Linux wird in vielen eingebetteten Systemen verwendet, z.B. in Routern, Smart-TVs, Automobilen und industriellen Steuerungssystemen. Seine Anpassungsfähigkeit und geringen Ressourcenanforderungen sind dabei von Vorteil.
5. **Mobilgeräte**:
   * Android, das auf Linux basiert, ist das weltweit am weitesten verbreitete Betriebssystem für Smartphones und Tablets.
6. **Desktops und Laptops**:
   * Obwohl der Marktanteil von Linux-Desktops im Vergleich zu Windows und macOS kleiner ist, wird es von vielen Technikbegeisterten, Entwicklern und in speziellen Umgebungen wie Bildungseinrichtungen und Regierungen genutzt.
7. **Unternehmensanwendungen**:
   * Viele Unternehmen setzen Linux für verschiedene Unternehmensanwendungen ein, darunter Datenbankmanagement, Anwendungsentwicklung, Netzwerkadministration und Virtualisierung (z.B. mit KVM, Docker).
8. **Forschung und Bildung**:
   * Universitäten und Forschungseinrichtungen nutzen Linux für eine Vielzahl von Anwendungen, von der Lehre über die Forschung bis hin zu den IT-Infrastrukturen der Institutionen.

Zusammengefasst ist Linux aufgrund seiner Stabilität, Sicherheit, Flexibilität und niedrigen Kosten in vielen Bereichen die bevorzugte Wahl. Es bietet eine ideale Plattform für eine breite Palette von Anwendungen, von kleinen eingebetteten Systemen bis hin zu den größten Supercomputern der Welt.