

## Thema 1: Die Linux-Community und Karriere im Open-Source-Umfeld

# 1.2 Lektion 1

<b>Zertifikat:</b>	Linux Essentials
<b>Version:</b>	1.6
<b>Thema:</b>	1 Die Linux-Community und Karriere im Open-Source-Umfeld
<b>Lernziel:</b>	1.2 Die wichtigsten Open-Source-Anwendungen
<b>Lektion:</b>	1 von 1

## Einführung

Eine Anwendung ist ein Computerprogramm, dessen Zweck nicht direkt an das Innenleben des Computers gebunden ist, sondern an Aufgaben, die vom Benutzer ausgeführt werden. Linux-Distributionen bieten viele Anwendungsmöglichkeiten für eine Vielzahl von Aufgaben, wie zum Beispiel Office-Anwendungen, Webbrowser, Multimedia-Player und -Editoren usw. Es gibt oft mehr als eine Anwendung oder ein Werkzeug, um eine bestimmte Aufgabe auszuführen. Es ist am Benutzer, die Anwendung auszuwählen, die seinen Bedürfnissen am besten entspricht.

## Softwarepakete

Fast jede Linux-Distribution bietet einen **vorinstallierten Satz von Standardanwendungen**. Neben diesen vorinstallierten Anwendungen verfügt eine Distribution über ein **Paket-Repository mit einer riesigen Sammlung von Anwendungen, die über ihren Paketmanager installiert werden**. Obwohl die verschiedenen Distributionen etwa dieselben Anwendungen anbieten, gibt es für verschiedene Distributionen verschiedene Paketverwaltungssysteme. **Debian, Ubuntu und Linux Mint** verwenden etwa die Tools **dpkg, apt-get und apt** zur Installation von Softwarepaketen, allgemein bekannt als **DEB-Pakete**.

Distributionen wie **Red Hat, Fedora und CentOS** verwenden stattdessen **rpm, yum und dnf**, die wiederum **RPM-Pakete** installieren. Da sich die Anwendungspakete für verschiedene Distributionsfamilien voneinander unterscheiden, ist es sehr wichtig, **Pakete vom richtigen Repository zu installieren**, die an die jeweilige Distribution angepasst sind. Der Benutzer muss sich üblicherweise nicht um diese Details kümmern, da der **Paketmanager** der Distribution die richtigen Pakete, die **Abhängigkeiten** und künftige Updates auswählt. Abhängigkeiten sind von den Programmen benötigte Hilfspakete. Wenn eine Bibliothek beispielsweise Funktionen zur Verarbeitung von JPEG-Bildern bereitstellt, die verschiedene Programme benötigen, wird eine solche Bibliothek üblicherweise in ein eigenes Paket gepackt, von dem alle Anwendungen, die diese Bibliothek benötigen, abhängen.

Die Befehle **dpkg** und **rpm** wirken auf einzelne Paketdateien. In der **Praxis** werden fast alle Aufgaben der Paketverwaltung mit den Befehlen **apt-get** oder **apt** auf Systemen, die DEB-Pakete verwenden, oder mit **yum** oder **dnf** auf Systemen, die RPM-Pakete verwenden, durchgeführt. Diese Befehle arbeiten mit Paketkatalogen, können neue Pakete und deren Abhängigkeiten herunterladen und nach neueren Versionen der installierten Pakete suchen.

## Paket-Installation

Angenommen Sie haben von einem Befehl namens `figlet` gehört, der vergrößerten Text auf dem Terminal ausgibt, und Sie wollen ihn ausprobieren. Sie erhalten nach Ausführung des Befehls `figlet` jedoch die folgende Meldung:

```
$ figlet
-bash: figlet: command not found
```

Das weist vermutlich darauf hin, dass das Paket nicht auf Ihrem System installiert ist. Wenn Ihre Distribution mit DEB-Paketen arbeitet, können Sie ihre **Repositories** mit `apt-cache search paketname` oder `apt search paketname` durchsuchen. Der Befehl `apt-cache` wird verwendet, um nach Paketen zu suchen und Informationen über verfügbare Pakete aufzulisten. Der folgende Befehl sucht nach allen Vorkommen der Zeichenfolge "figlet" in den Namen und Beschreibungen des Pakets:

```
$ apt-cache search figlet
figlet - Make large character ASCII banners out of ordinary text
```

Die Suche hat ein Paket namens *figlet* identifiziert, das dem fehlenden Befehl entspricht. Die Installation und Deinstallation eines Pakets erfordert spezielle Berechtigungen, die nur dem **Systemadministrator** gewährt werden: dem Benutzer namens `root`. Auf Desktop-Systemen können normale Benutzer Pakete installieren oder entfernen, indem sie den Befehl `sudo` den Installations-/Deinstallationsbefehlen voranstellen. Dafür müssen sie ihr Passwort eingeben, um fortzufahren. Für DEB-Pakete führen Sie die Installation mit dem Befehl `apt-get install paketname` oder `apt install paketname` aus:

```
$ sudo apt-get install figlet
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  figlet
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
```

An dieser Stelle wird das Paket heruntergeladen und auf dem System installiert. Alle Abhängigkeiten, die das Paket eventuell benötigt, werden ebenfalls heruntergeladen und installiert:

```
Need to get 184 kB of archives.
After this operation, 741 kB of additional disk space will be used.
Get:1 http://archive.raspbian.org/raspbian stretch/main armhf figlet armhf
2.2.5-2 [184 kB]
Fetched 184 kB in 0s (213 kB/s)
Selecting previously unselected package figlet.
(Reading database ... 115701 files and directories currently installed.)
Preparing to unpack .../figlet_2.2.5-2_armhf.deb ...
Unpacking figlet (2.2.5-2) ...
Setting up figlet (2.2.5-2) ...
update-alternatives: using /usr/bin/figlet-figlet to provide
/usr/bin/figlet (figlet) in auto mode
Processing triggers for man-db (2.7.6.1-2) ...
```

Nach Abschluss des Downloads werden alle Dateien an die richtigen Stellen kopiert und weitere Konfigurationen durchgeführt. Anschließend ist der Befehl verfügbar:

```
$ figlet Awesome!
```



In Distributionen, die auf RPM-Paketen basieren, erfolgt die Suche mit `yum search paketname` oder `dnf search paketname`. Nehmen wir an, Sie möchten einen etwas lockeren Text, illustriert mit einer Kuh, anzeigen, aber Sie sind sich nicht sicher, welches Paket diese Aufgabe erfüllen kann. Wie bei den DEB-Paketen akzeptieren auch die RPM-Suchbefehle beschreibende Begriffe

```
$ yum search speaking cow
Last metadata expiration check: 1:30:49 ago on Tue 23 Apr 2019 11:02:33 PM -03.
===== Name & Summary Matched: speaking, cow
=====
cowsay.noarch : Configurable speaking/thinking cow
```

Nachdem ein geeignetes Paket im Repository gefunden wurde, kann es mit `yum install paketname` oder `dnf install paketname` installiert werden:

```
$ sudo yum install cowsay
Last metadata expiration check: 2:41:02 ago on Tue 23 Apr 2019 11:02:33 PM -03.
Dependencies resolved.
=====
===
Package           Arch      Version      Repository
Size
=====
===
Installing:
cowsay            noarch    3.04-10.fc28 fedora      46
k

Transaction Summary
=====
===
Install 1 Package

Total download size: 46 k
Installed size: 76 k
Is this ok [y/N]: y
```

Auch hier wird das gewünschte Paket mit all seinen möglichen Abhängigkeiten heruntergeladen und installiert:

```
Downloading Packages:
cowsay-3.04-10.fc28.noarch.rpm          490 kB/s | 46 kB
00:00
=====
=====
```

```

Total                                     53 kB/s | 46 kB
00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing      :
1/1
Installing     : cowsay-3.04-10.fc28.noarch
1/1
Running scriptlet: cowsay-3.04-10.fc28.noarch
1/1
Verifying      : cowsay-3.04-10.fc28.noarch
1/1

Installed:
cowsay.noarch 3.04-10.fc28

Complete!

```

Der Befehl `cowsay` macht genau das, was sein Name verspricht:

```

$ cowsay "Brought to you by yum"
< Brought to you by yum >
-----
      \   ^__^
       \  (oo)\_______
          (__)\       )\/\
              ||----w |
              ||     ||

```

Obwohl sie nutzlos erscheinen mögen, bieten die Befehle `figlet` und `cowsay` eine Möglichkeit, die Aufmerksamkeit anderer Benutzer auf relevante Informationen zu lenken.

## Entfernen von Paketen

Die Befehle zum Installieren von Paketen dienen auch dazu, sie zu entfernen. Alle Befehle akzeptieren das Schlüsselwort `remove`, um ein installiertes Paket zu deinstallieren: `apt-get remove paketname` oder `apt remove paketname` für DEB-Pakete und `yum remove paketname` oder `dnf remove paketname` für RPM-Pakete. Der Befehl `sudo` wird auch benötigt, um das Entfernen durchzuführen. Um etwa das zuvor installierte Paket *figlet* aus einer DEB-basierten Distribution zu entfernen:

```

$ sudo apt-get remove figlet
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be REMOVED:
  figlet
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.
After this operation, 741 kB disk space will be freed.
Do you want to continue? [Y/n] Y

```

Nach der Bestätigung des Vorgangs wird das Paket aus dem System gelöscht:

```
(Reading database ... 115775 files and directories currently installed.)
Removing figlet (2.2.5-2) ...
Processing triggers for man-db (2.7.6.1-2) ...
```

Ähnlich ist die Vorgehensweise auf einem RPM-basierten System. Um das zuvor installierte Paket `cowsay` aus einer RPM-basierten Distribution zu entfernen:

```
$ sudo yum remove cowsay
Dependencies resolved.
=====
Package Arch Version Repository
Size
=====
Removing:
cowsay noarch 3.04-10.fc28 @fedora
76 k

Transaction Summary
=====
Remove 1 Package

Freed space: 76 k
Is this ok [y/N]: y
```

Auch hier wird eine Bestätigung angefordert und das Paket aus dem System gelöscht:

```
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing :
1/1
Erasing : cowsay-3.04-10.fc28.noarch
1/1
Running scriptlet: cowsay-3.04-10.fc28.noarch
1/1
Verifying : cowsay-3.04-10.fc28.noarch
1/1

Removed:
cowsay.noarch 3.04-10.fc28

Complete!
```

Die **Konfigurationsdateien der entfernten Pakete bleiben auf dem System erhalten**, so dass sie bei einer zukünftigen Neuinstallation des Pakets wieder verwendet werden können.

## Office-Anwendungen

Office-Anwendungen werden für die Bearbeitung von Dateien wie Texten, Präsentationen, Tabellenkalkulationen und anderen in einer Büroumgebung gebräuchlichen Formaten verwendet. Diese Anwendungen sind üblicherweise in Sammlungen organisiert, die als *Office Suites* bezeichnet werden.

**Lange Zeit war OpenOffice.org die unter Linux meistgenutzte Office Suite.** OpenOffice.org war eine Open-Source-Version der von Sun Microsystems herausgegebenen *StarOffice Suite*.

Einige Jahre später wurde Sun von der *Oracle Corporation* übernommen, die das Projekt wiederum an die *Apache Foundation* übertrug, und OpenOffice.org wurde in *Apache OpenOffice* umbenannt. In der Zwischenzeit veröffentlichte die *Document Foundation* auf derselben Code-Basis eine **weitere Office Suite** namens **LibreOffice**.

Die beiden Projekte haben die gleichen Grundfunktionen und sind kompatibel zu den Dokumentenformaten von *Microsoft Office*. **Das bevorzugte Dokumentenformat ist jedoch das Open Document Format, ein vollständig offenes und ISO-standardisiertes Dateiformat.** Die Verwendung von **ODF** stellt sicher, dass Dokumente zwischen Betriebssystemen und Anwendungen verschiedener Anbieter, wie beispielsweise *Microsoft Office*, übertragen werden können. Die Hauptanwendungen von OpenOffice/LibreOffice sind:

Writer	Textverarbeitung
Calc	Tabellenkalkulation
Impress	Präsentationen
Draw	Vektorzeichnung
Math	Mathematische Formeln
Base	Datenbank



**Sowohl LibreOffice als auch Apache OpenOffice sind Open Source Software, aber LibreOffice ist unter LGPLv3 lizenziert,** während Apache OpenOffice unter der Apache License 2.0 steht. **Der Lizenzunterschied hat zur Folge, dass LibreOffice Verbesserungen von Apache OpenOffice übernehmen kann,** aber Apache OpenOffice keine Verbesserungen von LibreOffice. Dies und eine aktivere Entwicklergemeinschaft sind die **Gründe, dass die meisten Distributionen LibreOffice als Standard-Office-Suite nutzen.**

## Webbrowser

Den meisten Benutzern dient ein Computer vor allem dazu, Zugang zum Internet zu gewährleisten. Heutzutage können Webseiten als vollwertige App fungieren, mit dem Vorteil, dass sie von überall her zugänglich sind, ohne zusätzliche Software installieren zu müssen. Das macht den Webbrowser zur wichtigsten Anwendung des Betriebssystems, zumindest für den durchschnittlichen Benutzer.

Die wichtigsten Webbrowser im Linux-Umfeld sind **Google Chrome und Mozilla Firefox.** Chrome ist ein von Google gepflegter Webbrowser, basiert aber auf dem Open Source Browser *Chromium*, der über den Paketmanager der Distribution installiert werden kann und vollständig mit Chrome kompatibel ist. Firefox wird von Mozilla, einer gemeinnützigen Organisation, verwaltet und ist ein Browser, dessen Ursprünge mit Netscape verbunden sind, dem ersten beliebten Webbrowser, der das Open-Source-Modell verwendet. Die Mozilla Foundation ist tief in der Entwicklung offener Standards verwurzelt, die dem modernen Web zugrunde liegen.

Mozilla entwickelt auch andere Anwendungen, wie den **E-Mail-Client Thunderbird**. Viele Benutzer entscheiden sich für Webmail anstelle einer dedizierten E-Mail-Anwendung, aber ein Client wie Thunderbird bietet zusätzliche Funktionen und integriert sich am besten mit anderen Anwendungen auf dem Desktop.

## Multimedia

Im Vergleich zu den verfügbaren Webanwendungen sind Desktop-Anwendungen nach wie vor die beste Option für die Erstellung von Multimedia-Inhalten. Multimedia-bezogene Aktivitäten wie Video-Rendering erfordern oft viele Systemressourcen, die am besten von einer lokalen Desktop-Anwendung verwaltet werden. Einige der beliebtesten Multimedia-Anwendungen unter Linux und ihre Einsatzbereiche sind:

**Blender**: Ein 3D-Renderer zur Erstellung von Animationen, mit dem auch 3D-Objekte exportiert werden können, die auf einem 3D-Drucker gedruckt werden sollen.

**GIMP**: Ein vollwertiges Bildbearbeitungsprogramm, das mit *Adobe Photoshop* vergleichbar ist, aber über eigene Konzepte und Werkzeuge verfügt. GIMP kann die meisten Bitmap-Dateien wie JPEG, PNG, GIF, TIFF und viele andere erstellen, bearbeiten und speichern.

**Inkscape**: Ein Vektorgrafik-Editor, ähnlich wie *Corel Draw* oder *Adobe Illustrator*. Das Standardformat von Inkscape ist SVG, ein offener Standard für Vektorgrafiken. SVG-Dateien können von jedem Webbrowser geöffnet werden und sind als Vektorgrafiken in flexiblen Webseiten-Layouts einsetzbar.

**Audacity**: Ein Audio-Editor, mit dem man filtern, Effekte anwenden und zwischen vielen verschiedenen Audioformaten wie MP3, WAV, OGG, FLAC etc. konvertieren kann.

**ImageMagick**: ImageMagick ist ein Kommandozeilenwerkzeug zur Konvertierung und Bearbeitung der meisten Bilddateitypen. Es dient auch dazu, PDF-Dokumente aus Bilddateien zu erstellen und umgekehrt.

Es gibt auch zahlreiche Anwendungen zur Multimedia-Wiedergabe. Die beliebteste Anwendung für die Video-Wiedergabe ist **VLC**, aber einige Benutzer bevorzugen andere Alternativen, wie **smplayer**. Die lokale Musikwiedergabe hat auch viele Optionen, wie *Audacious*, *Banshee* und *Amarok*, die auch eine Sammlung lokaler Sounddateien verwalten können.

## Server-Programme

Wenn ein Webbrowser eine Seite von einer Website lädt, verbindet er sich tatsächlich mit einem entfernten Computer und fragt nach einer bestimmten Information. In diesem Szenario wird der Computer, auf dem der Webbrowser läuft, als *Client* bezeichnet, der entfernte Computer als *Server*.

Der Server-Rechner, der ein gewöhnlicher Desktop-Computer oder spezielle Hardware sein kann, benötigt ein spezifisches Programm, das die bereitzustellenden Informationen verwaltet. Was die Bereitstellung von Webseiten betrifft, so setzen die meisten Server auf der ganzen Welt Open-Source-Serverprogramme ein. Dieses spezielle Serverprogramm wird als **HTTP-Server** bezeichnet (HTTP steht für *Hyper Text Transfer Protocol*), und die **beliebtesten sind Apache, Nginx und lighttpd**.



Die Aufgabe eines HTTP-Servers besteht darin, alle angeforderten Daten zu sammeln und an den Browser zurückzusenden, der dann den Inhalt, wie er durch das empfangene HTML-Dokument (HTML steht für *Hyper Text Markup Language*) und andere unterstützende Dateien definiert ist, anordnet. Daher umfasst das Rendern einer Webseite Operationen sowohl auf der Serverseite wie auch auf der Clientseite. Beide Seiten können benutzerdefinierte Skripte verwenden, um bestimmte Aufgaben zu erfüllen. Auf der **HTTP-Serverseite** ist es durchaus üblich, die Skriptsprache **PHP** zu verwenden. **JavaScript ist die Skriptsprache, die auf der Clientseite (dem Webbrowser) verwendet wird.**

**Serverprogramme** können alle Arten von Informationen bereitstellen: Es ist nicht ungewöhnlich, dass **ein Serverprogramm Informationen von anderen Serverprogrammen anfordert**, z.B. wenn ein HTTP-Server Informationen von einem Datenbankserver benötigt.

Wenn beispielsweise eine dynamische Seite angefordert wird, fragt der HTTP-Server in der Regel eine Datenbank ab, um alle erforderlichen Informationen zu sammeln, und sendet den dynamischen Inhalt an den Client zurück. Ähnlich verhält es sich, wenn sich ein Benutzer auf einer Website registriert: Der HTTP-Server sammelt die vom Client gesendeten Daten und speichert sie in einer Datenbank.

**Eine Datenbank ist ein organisierter Satz von Informationen.** Ein Datenbankserver speichert Inhalte in formatierter Form und ermöglicht das Lesen, Schreiben und Verknüpfen großer Datenmengen in hoher Geschwindigkeit und Zuverlässigkeit. Open-Source-Datenbankserver werden in vielen Anwendungen eingesetzt, nicht nur im Internet, sondern auch in lokalen Anwendungen, die Daten durch Verbindung zu einem lokalen Datenbankserver speichern können. Der häufigste Typ von Datenbank ist die *relationale Datenbank*, in der die Daten in vordefinierten Tabellen organisiert sind. **Die beliebtesten relationalen Open-Source-Datenbanken sind MariaDB (entstanden aus MySQL) und PostgreSQL.**

## Data Sharing

In lokalen Netzwerken, wie sie in Büros und zu Hause zu finden sind, ist es wünschenswert, dass Computer nicht nur auf das Internet zugreifen, sondern auch miteinander kommunizieren. Manchmal fungiert ein Computer als Server, manchmal derselbe Computer als Client. Das ist zum Beispiel notwendig, wenn man auf **Dateien auf einem anderen Computer im Netzwerk zugreifen möchte** — zum Beispiel auf eine auf einem Desktop-Computer gespeicherte Datei von einem tragbaren Gerät aus — ohne die Mühe, sie auf ein USB-Laufwerk oder dergleichen zu kopieren.

**Zwischen Linux-Maschinen wird häufig NFS (Network File System) verwendet.** Das NFS-Protokoll ist der Standardweg, um Dateisysteme in Netzwerken zu teilen, die nur mit Unix/Linux-Maschinen ausgestattet sind. NFS ermöglicht es einem Computer, **eines oder mehrere seiner Verzeichnisse mit bestimmten Computern im Netzwerk zu teilen**, damit er Dateien in diesen Verzeichnissen lesen und schreiben kann.

**NFS kann sogar verwendet werden, um den Verzeichnisbaum eines ganzen Betriebssystems mit Clients zu teilen, die es zum Booten verwenden.** Diese Computer, die sogenannten *Thin Clients*, werden meist in großen Netzwerken



eingesetzt, um die Wartung jedes einzelnen Betriebssystems auf jeder Maschine zu vermeiden.

Wenn **andere Arten von Betriebssystemen** an das Netzwerk angeschlossen sind, empfiehlt es sich, ein für alle verständliches **Data-Sharing-Protokoll** zu verwenden. Diese Aufgaben erfüllt *Samba*. **Samba implementiert ein Protokoll für den Dateiaustausch über das Netzwerk**, das ursprünglich für das Windows-Betriebssystem entwickelt wurde, heute aber mit allen gängigen Betriebssystemen kompatibel ist. **Mit Samba können Computer im lokalen Netzwerk nicht nur Dateien, sondern auch Drucker teilen.**

In einigen lokalen Netzwerken wird die Berechtigung, die bei der Anmeldung auf einer Workstation erteilt wird, von einem **zentralen Server**, dem **Domänencontroller**, vergeben. Er verwaltet den Zugriff auf verschiedene lokale und entfernte Ressourcen.

**Der Domänencontroller ist ein Dienst aus dem Active Directory von Microsoft. Linux-Workstations können sich mit einem Domänencontroller über Samba oder ein Authentifizierungssystem namens SSSD verbinden. Ab Version 4 kann Samba auch als Domänencontroller in heterogenen Netzwerken fungieren.**

Wenn es darum geht, eine Cloud-Computing-Lösung zu implementieren, die verschiedene Methoden des webbasierten Datenaustauschs bietet, sollten zwei Alternativen in Betracht gezogen werden: **ownCloud** und **Nextcloud**. Beide Projekte sind sehr ähnlich, da Nextcloud ein Spin-off von ownCloud ist, was bei Open-Source-Projekten nicht ungewöhnlich ist. Solche Spin-offs bezeichnet man als *Fork*. Beide bieten die gleichen Grundfunktionen: **File Sharing und Synchronisation, kollaborative Arbeitsbereiche, Kalender, Kontakte und E-Mails** — alles über Desktop-, Mobil- und Webschnittstellen. Nextcloud bietet auch private Audio-/Videokonferenzen an, während sich ownCloud mehr auf die gemeinsame Nutzung von Dateien und die Integration mit Software von Drittanbietern konzentriert. Viele weitere Funktionen sind als Plugins verfügbar, die bei Bedarf später aktiviert werden können.

Sowohl ownCloud als auch Nextcloud bieten eine kostenpflichtige Version mit zusätzlichen Funktionen und erweitertem Support an, die sich von anderen kommerziellen Lösungen dadurch unterscheidet, dass sie **Nextcloud oder ownCloud kostenlos auf einem privaten Server installieren können**, ohne sensible Daten auf einem unbekannten Server vorzuhalten. Da alle Dienste auf HTTP-Kommunikation basieren und in PHP geschrieben sind, muss die Installation auf einem zuvor konfigurierten Webserver wie Apache erfolgen. Wenn Sie die Installation von ownCloud oder Nextcloud auf Ihrem eigenen Server in Betracht ziehen, stellen Sie sicher, dass HTTPS auch alle Verbindungen zu Ihrer Cloud verschlüsselt.

## **Netzwerkadministration**

Kommunikation zwischen Computern ist nur möglich, wenn das Netzwerk ordnungsgemäß funktioniert. Die Netzwerkkonfiguration erfolgt in der Regel durch eine Reihe von Programmen, die auf dem Router laufen und für die Einrichtung und Überprüfung der Netzwerkfähigkeit zuständig sind. Voraussetzung dafür sind zwei grundlegende Netzwerkdienste: **DHCP (Dynamic Host Configuration Protocol) und DNS (Domain Name System).**

**DHCP ist dafür verantwortlich, dem Host eine IP-Adresse zuzuweisen**, wenn ein Netzkabel angeschlossen ist oder wenn das Gerät in ein drahtloses Netzwerk eintritt. Bei der Verbindung mit dem Internet stellt der DHCP-Server des ISP dem anfragenden Gerät eine IP-Adresse zur Verfügung. Ein DHCP-Server ist auch in lokalen Netzwerken sehr nützlich, um allen angeschlossenen Geräten automatisch IP-Adressen zuzuweisen.

Wenn **DHCP nicht konfiguriert** ist oder nicht ordnungsgemäß funktioniert, muss die IP-Adresse jedes mit dem Netzwerk verbundenen Geräts **manuell konfiguriert** werden. Es ist nicht sinnvoll, die IP-Adressen in großen Netzwerken oder sogar in kleinen Netzwerken manuell einzustellen; deshalb verfügen die meisten Netzwerkrouter über einen vorkonfigurierten DHCP-Server.

Die IP-Adresse wird benötigt, um mit einem anderen Gerät in einem IP-Netzwerk zu kommunizieren, aber Domainnamen wie `www.lpi.org` sind viel einfacher zu merken als eine IP-Adresse wie `203.0.113.165`. Der Domainname allein reicht jedoch nicht aus, um die Kommunikation über das Netzwerk herzustellen, weshalb der **Domainname von einem DNS-Server in eine IP-Adresse übersetzt werden muss**: Die IP-Adresse des DNS-Servers wird vom DHCP-Server des ISP bereitgestellt und von allen angeschlossenen Systemen zur Übersetzung von Domainnamen in IP-Adressen verwendet.

Sowohl die DHCP- als auch die DNS-Einstellungen lassen sich über die vom Router bereitgestellte Webschnittstelle ändern. So ist es zum Beispiel möglich, die IP-Zuweisung nur auf bekannte Geräte zu beschränken oder bestimmten Maschinen eine feste IP-Adresse zuzuordnen. Auch der vom ISP bereitgestellte Standard-DNS-Server lässt sich darüber ändern. Einige DNS-Server von Drittanbietern, wie sie von Google oder **OpenDNS** bereitgestellt werden, liefern bisweilen schnellere Antworten und bieten zusätzliche Funktionen.

## Programmiersprachen

Alle Computerprogramme (Client- und Serverprogramme, Desktop-Anwendungen und das Betriebssystem selbst) werden in einer oder mehreren Programmiersprachen erstellt, wobei es sich bei den Programmen um eine einzelne Datei oder ein komplexes System aus Hunderten von Dateien handeln kann, die das Betriebssystem als Befehlssequenz behandelt, die vom Prozessor und anderen Geräten interpretiert und ausgeführt wird.

Es gibt zahlreiche Programmiersprachen für sehr unterschiedliche Zwecke, und Linux-Systeme stellen viele davon bereit. Da Open-Source-Software auch die Quellen der Programme umfasst, bieten Linux-Systeme Entwicklern perfekte Voraussetzungen, um Software zu verstehen und nach ihren eigenen Bedürfnissen zu modifizieren oder zu erstellen.

Jedes Programm beginnt als Textdatei, dem *Quellcode*. Dieser Quellcode ist in einer mehr oder weniger menschenlesbaren Sprache geschrieben, die beschreibt, was das Programm tut. Ein Computerprozessor kann diesen Code nicht direkt ausführen; bei *kompilierten Sprachen* wird der Quellcode daher in eine *binäre Datei* umgewandelt, die dann vom Computer ausgeführt werden kann. Ein Programm namens *Compiler* ist für die Konvertierung von Quellcode in eine ausführbare Form verantwortlich. Da die kompilierte Binärdatei für einen Prozessortyp spezifisch ist,

muss das Programm möglicherweise neu kompiliert werden, um auf einem anderen Computertyp ausgeführt zu werden.

Bei *interpretierten Sprachen* muss das Programm nicht vorher kompiliert werden. Stattdessen liest ein *Interpreter* den Quellcode und führt bei jedem Programmstart seine Befehle aus. Das macht die Entwicklung einfacher und schneller, aber dafür sind interpretierte Programme tendenziell langsamer als kompilierte.

Hier einige der beliebtesten Programmiersprachen:

JavaScript:

JavaScript ist eine Programmiersprache, die vor allem auf Webseiten verwendet wird. Ursprünglich waren JavaScript-Anwendungen sehr einfach, wie Formularvalidierungsroutinen. Heute wird JavaScript zur Erstellung sehr komplexer Anwendungen nicht nur im Web, sondern auch auf Servern und mobilen Geräten eingesetzt.

C:

Die Programmiersprache C ist eng mit Betriebssystemen, insbesondere Unix, verbunden, wird aber verwendet, um jede Art von Programm auf fast jede Art von Gerät zu schreiben. Die großen Vorteile von C sind Flexibilität und Geschwindigkeit. Der gleiche in C geschriebene Quellcode kann auf verschiedenen Plattformen und Betriebssystemen mit wenig oder gar keiner Änderung kompiliert werden. Nach der Kompilierung läuft das Programm jedoch nur auf dem Zielsystem.

Java:

Der Hauptaspekt von Java ist, dass Programme, die in dieser Sprache geschrieben wurden, portabel sind, was bedeutet, dass das gleiche Programm auf verschiedenen Betriebssystemen ausgeführt werden kann. Trotz des Namens hat Java nichts mit JavaScript zu tun.

Perl:

Perl ist eine Programmiersprache, die meist zur Verarbeitung von Textinhalten verwendet wird. Ihre besondere Stärke sind reguläre Ausdrücke, was Perl zu einer Sprache insbesondere für Textfilterung und -analyse macht.

Shell:

Die Shell, insbesondere die Bash-Shell, ist nicht nur eine Programmiersprache, sondern eine interaktive Schnittstelle zur Ausführung anderer Programme. Shell-Programme, *Shell-Skripte* genannt, können komplexe oder sich wiederholende Aufgaben in der Kommandozeilenumgebung automatisieren.

Python:

Python ist eine sehr beliebte Programmiersprache bei Studenten und Fachleuten, die nichts direkt mit Informatik zu tun haben. Obwohl Python über fortgeschrittene Funktionen verfügt, ist es aufgrund seines einfachen Ansatzes ein guter Einstieg in die Programmierung.

PHP:

PHP wird meist als serverseitige Skriptsprache zur Generierung von Inhalten für das Web verwendet. Die meisten Online-HTML-Seiten sind keine statischen Dateien, sondern dynamische Inhalte, die vom Server aus verschiedenen Quellen, wie z.B. Datenbanken, generiert werden. PHP-Programme — manchmal auch nur PHP-Seiten oder PHP-Skripte genannt — dienen oft dazu, diese Inhalte zu generieren. Der Begriff LAMP kommt aus der Kombination eines Linux-Betriebssystems, eines Apache HTTP-Servers, einer MySQL (oder MariaDB) Datenbank und PHP-Programmierung. LAMP-Server sind eine sehr beliebte Lösung für den Betrieb von Webservern. Neben PHP können auch alle zuvor beschriebenen Programmiersprachen zur Implementierung solcher Anwendungen verwendet werden.

C und Java sind kompilierte Sprachen, wobei der in C geschriebene Quellcode in binären Maschinencode umgewandelt wird, während der Java-Quellcode in *Bytecode* umgewandelt wird, der in einer speziellen Softwareumgebung namens *Java Virtual Machine* ausgeführt wird. JavaScript, Perl, Shell-Skript, Python und PHP sind interpretierte Sprachen, die auch als *Skriptsprachen* bezeichnet werden.

## Geführte Übungen

1. Identifizieren Sie für jeden der folgenden Befehle, ob er zum *Debian-Paketverwaltungssystem* oder zum *Red Hat-Paketverwaltungssystem* gehört:

dpkg	
rpm	
apt-get	
yum	
dnf	

2. Welcher Befehl könnte verwendet werden, um Blender auf Ubuntu zu installieren? Wie kann das Programm nach der Installation ausgeführt werden?
3. Welche Anwendung aus der LibreOffice Suite dient der elektronischen Tabellenkalkulation?
4. Welcher Open-Source-Webbrowser bildet die Grundlage für die Entwicklung von Google Chrome?
5. SVG ist ein offener Standard für Vektorgrafiken. Welche ist die beliebteste Anwendung zum Bearbeiten von SVG-Dateien in Linux-Systemen?
6. Schreiben Sie für jedes der folgenden Dateiformate den Namen einer Anwendung, die die entsprechende Datei öffnen und bearbeiten kann:

png	
doc	
xls	
ppt	
wav	

7. Welches Softwarepaket ermöglicht den Dateiaustausch zwischen Linux- und Windows-Rechnern über das lokale Netzwerk?

## Offene Übungen

1. Sie wissen, dass Konfigurationsdateien auch dann erhalten bleiben, wenn das zugehörige Paket aus dem System entfernt wird. Wie können Sie das Paket namens *cups* und seine Konfigurationsdateien automatisch aus einem DEB-basierten System entfernen?
2. Angenommen Sie haben viele TIFF-Bilddateien und möchten diese in JPEG konvertieren. Welches Softwarepaket könnte verwendet werden, um diese Dateien direkt auf der Kommandozeile zu konvertieren?
3. Welches Softwarepaket müssen Sie installieren, um Microsoft-Word-Dokumente öffnen zu können, die Ihnen von einem Windows-Benutzer zugesandt wurden?
4. Jedes Jahr veranstaltet linuxquestions.org eine Umfrage über die beliebtesten Linux-Anwendungen. Finden Sie unter <https://www.linuxquestions.org/questions/2018-linuxquestions-org-members-choice-awards-128/> heraus, welche Desktop-Anwendungen bei erfahrenen Linux-Anwendern am beliebtesten sind.

## Zusammenfassung

In dieser Lektion haben Sie gelernt:

- Die Paketmanagementsysteme, die in den wichtigsten Linux-Distributionen verwendet werden
- Open-Source-Anwendungen, die gängige Dateiformate bearbeiten können
- Die Serverprogramme, die vielen wichtigen Internet- und lokalen Netzwerkdiensten zugrunde liegen
- Verbreitete Programmiersprachen und deren Anwendungen