

2 Linux-Grundlagen

In diesem Kapitel erfahren Sie

- ▶ was Linux und UNIX kennzeichnet und wie sie entstanden sind
- ▶ worin die Unterschiede zwischen Linux und Windows liegen
- ▶ was ein Kommandointerpreter bzw. eine Shell ist
- ▶ wie die grafische Oberfläche KDE aufgebaut ist

2.1 Linux und UNIX - Entstehung und Geschichte

Typische Schwierigkeiten von Einsteigern

Systemadministratoren, die von anderen, grafisch orientierten Betriebssystemen auf UNIX oder Linux umsteigen, haben zu Beginn häufig Probleme, sich in das "neue" Betriebssystem hineinzudenken. Diese Probleme stammen zum größten Teil von folgenden Tatsachen:

- ✓ Die Konzepte hinter Linux/UNIX sind für Windows-erfahrene Benutzer häufig ungewohnt. Linux bzw. UNIX haben ihre Wurzeln in den 60er- und 70er-Jahren, wurden seitdem kontinuierlich weiterentwickelt und sind dementsprechend ausgereift. Oft spiegeln sie jedoch den ursprünglichen Ansatz wider, der heute nicht mehr unbedingt nachvollziehbar ist. So ist einem Benutzer heute z. B. nicht ohne weiteres klar, warum eine Diskette oder eine CD-ROM unmountet, also ausgehängt werden muss, bevor er sie entnehmen kann. Dieses Konzept stammt aus der Zeit der Bandlaufwerke und wurde bis heute aus Sicherheits- und Performancegründen beibehalten, ja sogar auf alle Datenträger übertragen.
- ✓ Die Funktionalität ist bei UNIX/Linux-Systemen stärker gewichtet als bei anderen Systemen. So kann z. B. jeder Vorgang automatisiert werden, sodass Sie beispielsweise vollautomatisch je nach Uhrzeit unterschiedliche Routing-Tabellen für die Netzwerkkonfiguration einsetzen lassen können. Für den Benutzer bedeutet dies: Er muss erst lernen, den anderen, ungewohnten Ansatz umzusetzen, ist dann aber schnell zu mehr und komplexeren Aktionen fähig. Die Lernkurve bei Linux verläuft also anfangs flacher, steigt aber nach kurzer Eingewöhnungsphase exponentiell steil an, da die einmal gelernten Konzepte an vielen verschiedenen Einsatzorten eingesetzt werden können.
- ✓ Linux gilt als ein System, das nur oder meistens an der Befehlszeile bedient werden muss. Moderne Systeme zeichnen sich jedoch gerade durch mächtige grafische Tools aus, die über einen großen Funktionsumfang verfügen. Die Befehlszeile ist nicht zwingend notwendig, wird aber den meisten UNIX-Administratoren im Laufe der Zeit als schnelles, einfaches und mächtiges Werkzeug vertraut.

Mit diesem Wissen im Hintergrund ist es in der Regel einfacher, die Abläufe, die Konzeption und die Befehlsstruktur von Linux/UNIX zu verstehen und zu beherrschen. Alles, was Sie hier lernen, können Sie in der Regel in zahlreichen anderen Situationen wieder verwenden.

Geschichte - Den Grundstein bildet UNIX

Die Entwicklung dieses Betriebssystems begann vor über 30 Jahren: An den Bell Laboratories von **AT & T** wurde 1969 das erste **UNIX**-Betriebssystem entwickelt und ab 1971 eingesetzt.

Im Gegensatz zu proprietären Betriebssystemen, die einem Hersteller gehören, der diese auch weiterentwickelt und vermarktet, wurde UNIX bald von den verschiedensten Herstellern weiterentwickelt.

Die besonderen Merkmale des Betriebssystems UNIX sind:

- ✓ Das hierarchisch aufgebaute Dateisystem
- ✓ Die Möglichkeit, Prozesse im Hintergrund ablaufen zu lassen, sowie die Multitasking- und Multiuser-Fähigkeit
- ✓ Das Vorhandensein einer identischen Schnittstelle für Daten-, Geräte- und Interprozesskommunikation
- ✓ Die Filtertechnik (Piping)
- ✓ Eine große Menge an Werkzeugen und Hilfsprogrammen
- ✓ Eine sehr hohe Portabilität bei gleichzeitiger Stabilität

Die vielen verschiedenen Hersteller führten dazu, dass es kein einheitliches UNIX-System gab. 1996 bildete sich aus diesen Gründen dann **The Open Group**, um die Entwicklung von UNIX-Systemen einheitlicher zu gestalten.

Es entstanden schließlich zwei Hauptvarianten von UNIX-Systemen:

- ✓ Auf der Basis der Variante AT & T System V
- ✓ Auf der Basis der Variante BSD v4 (Berkeley Software Distribution)

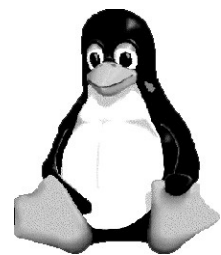
Auf diese lassen sich bis heute fast alle UNIX-Derivate zurückführen.

Linux - ein nachprogrammiertes UNIX-System

Linux ist trotz einer weit verbreiteten Meinung kein UNIX-System, sondern nur ein UNIX-ähnliches System. Das bedeutet, dass es die Form und Funktion von UNIX nachahmt, aber aus keinem der lizenzierten Quellcodes stammt, wie z. B. **System V** oder **BSD**.

- ✓ Der finnische Student **Linus Torvalds** entwickelte Anfang der neunziger Jahre gemeinsam mit anderen Entwicklern Linux und stellte den Quellcode über das Internet jedem frei zur Verfügung.
- ✓ Unzählige Programmierer in der ganzen Welt ergänzen diesen Code ständig um neue Funktionalitäten, wie Sie auch in anderen Systemen erhalten sind.
- ✓ Da niemals proprietärer Code portiert oder eingebunden wird oder wurde, darf Linux unter der Bedingung, dass der vollständige Quellcode mit ausgeliefert wird, frei weitergegeben werden.

Linux ist ein frei verfügbares Multitasking- und Multiuser-Betriebssystem, das ständig weiterentwickelt wird. Es bietet mittlerweile die gesamte Funktionalität, die von modernen Betriebssystemen erwartet werden kann. Echtes preemptives Multitasking, virtuelle Speicherverwaltung, dynamisch nachladbare Bibliotheken, Multiprozessorsupport und andere moderne Konzepte machen das UNIX-orientierte Betriebssystem zur optimalen Lösung für viele Einsatzgebiete.



Linux-Maskottchen Tux

Die Palette der Benutzer reicht von privaten Anwendern über Schulungsunternehmen, Universitäten, Forschungszentren bis hin zu kommerziellen Anwendern und Unternehmen, die in Linux eine echte Alternative zu anderen Betriebssystemen sehen. Derzeit erfährt Linux seine größten Zuwachsraten im gesamten Internet-/Server-Bereich, in dem es inzwischen aufgrund seiner Netzwerk-Performance, Sicherheitsfeatures und frei erhältlichen Programme eine Spitzenposition eingenommen hat.

- ✓ Linux wurde von Anfang an unter die **GPL**, die General Public License, gestellt.
- ✓ Die GPL entstand im Rahmen des GNU-Projekts. **GNU** steht für "GNU is not UNIX". Dieses Projekt bemüht sich unter anderem um die Entwicklung und die Verteilung freier Software. Die Grundidee ist, dass jede Software frei und änderbar sein sollte, um die Freiheit des Benutzers zu gewährleisten. Die GPL-Lizenz garantiert jedem den kostenlosen Zugang zum Quellcode dieses Betriebssystems.
- ✓ Linux kann frei und kostenlos verteilt, eingesetzt und erweitert oder weitervertrieben werden, wobei der Quellcode aber immer veröffentlicht werden muss.
- ✓ Unzählige Entwickler haben Einblick in sämtliche Quellcodes und können dadurch sehr einfach neue Funktionen integrieren bzw. Programmierfehler schnell finden und eliminieren. So können z. B. Treiber für neue Hardware sehr schnell integriert werden.

Linux-Distributionen

Einige Unternehmen scheinen Linux zu verkaufen, obwohl es doch angeblich frei verfügbar ist. Es wird aber in Wirklichkeit nicht der frei verfügbare Quellcode, sondern es werden nur die zusätzlichen Tools (bekanntestes Beispiel bei SuSE Linux: YaST) und zum Teil der Support für die jeweilige Distribution verkauft. Zusätzlich ist in der Regel pro Distribution mindestens ein umfassendes Handbuch inbegriffen, und das Betriebssystem Linux samt Utilities befindet sich auf einer oder mehreren CD-ROMs. Der Benutzer erspart sich damit die Mühe, die Quellcodes und Zusatzprodukte selbst aus dem Internet zusammensuchen zu müssen. Außerdem enthalten alle Distributionen relativ fortgeschrittene Installationsprozeduren, sodass auch ein unerfahrener Benutzer schnell eine Linux-Installation durchführen kann.

Eine Besonderheit der letzten Jahre stellen die Community-Distributionen dar. SuSE hat, wie Redhat vor einigen Jahren schon, eine abgespeckte Linux-Version der Community ohne lizenzpflichtige Software veröffentlicht und hilft tatkräftig bei der Weiterentwicklung. Diese vollkommen freien Versionen heißen openSUSE oder SuSE Linux Open Source Edition, auch SuSE Linux OpenSource Software only (OSS) genannt. Selbst Novell und SuSE scheinen sich beim Namen nicht vollständig einig zu sein. Die Software ist an verschiedenen Stellen unter verschiedenen Namen aufgeführt. In diesem Buch sprechen wir von SuSE Linux 11 OSS.



SuSE Linux 11

Parallel dazu gibt es ein kostenpflichtiges Produkt von openSUSE 11.0, das 90 Tage Installations-Support, Handbücher und nicht der GPL unterliegende Software beinhaltet. OpenSUSE 11.0 kann auch von der Webseite <http://www.opensuse.org> heruntergeladen werden.

Die Version 11.0 enthält ein überarbeitetes Installationssystem, bessere Unterstützung von neuer Hardware und die neue grafische Oberfläche KDE 4.0, die auf die Version 4.1 über das Internet aktualisiert werden kann.

Speziell für Firmen gibt es die Produkte SUSE Linux Enterprise Desktop und SUSE Linux Enterprise Server mit Version 10 (zur Zeit der Überarbeitung dieses Buches), die weitergehenden Support und besonders getestete Software mitbringen.

2.2 Aufbau des Linux-Systems

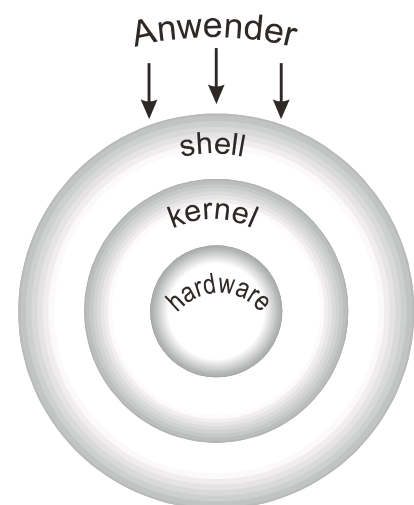
Komponenten

Die Hauptkomponenten eines Linux-Systems sind:

- ✓ Der Systemkern: der Kernel
- ✓ Ladbare Treiber für die Hardware, die Kernelmodule
- ✓ Ein Kommandointerpreter (wie die Shell)
- ✓ Programme, wie Dienstprogramme oder grafische Benutzeroberflächen (wie KDE)

Die Aufgaben des Systemkerns

Die innerste Schicht des Betriebssystems wird meist "Kernel" (engl. für Kern) genannt. Der Betriebssystemkern befindet sich direkt über der Hardware und hat die Aufgabe, die Hardware zu betreiben und zu nutzen.



Bedeutung der Shell

Im Kernel sind alle eigentlichen Betriebssystemaufgaben realisiert:

- ✓ Die Kommunikation mit den Hardware-Komponenten des Computers
- ✓ Um einen geregelten Ablauf der Programme zu ermöglichen, stellt der Kernel diesen die erforderlichen Hardware-Ressourcen zur Verfügung.
- ✓ Der Kernel kontrolliert die Kommunikation von Programmen untereinander, mit der Hardware und mit den Benutzern.
- ✓ Der Kernel ist zuständig für die Verwaltung der dem System bekannten Daten und bestimmt, in welcher Form und an welcher Stelle die Daten abgelegt werden.
- ✓ Der Kernel ermöglicht es, in gewünschter Weise auf diese Daten zugreifen zu können; er sorgt für Datenschutz in einer durch den Anwender festgelegten Form.
- ✓ Der Kernel ermöglicht, dass mehrere Aufgaben gleichzeitig durchgeführt werden können (Multitasking) und mehrere Anwender zur gleichen Zeit mit dem System arbeiten können (Multiuser-Betrieb).

Der Kernel umfasst eine große Anzahl von Funktionen, die der Benutzer aber nicht direkt verwenden kann. Die Funktionen stehen nur eingebettet in Systemprogrammen zur Verfügung. Der von den Linux-Distributoren mitgelieferte Kernel ist so allgemein gestaltet, dass er auf den meisten Rechnern problemlos funktioniert. Sie können den Kernel Ihres Linux-Systems aber auch exakt für Ihre Hardware maßschneidern.



Die Aufgaben des Kommandointerpreters

Um Anweisungen an das Betriebssystem zu geben (z. B., um eine Datei zu kopieren), benötigt der Benutzer eine Eingabemöglichkeit. Dazu wird ein sogenannter Kommandointerpreter mit einer einfachen, textbasierten Oberfläche verwendet. In diesen Kommandointerpreter kann der Anwender Befehle eingeben, die an das Betriebssystem weitergeleitet werden. Die Rückmeldungen des Betriebssystems (z. B.: "die Datei wurde kopiert" oder "ein Fehler trat bei der Ausführung des Befehls auf") werden im Kommandointerpreter wieder ausgegeben. Um eine Shell zu erhalten, müssen Sie das Programm `konsole` aufrufen (im Startmenü unter FAVORITEN ganz unten) oder sich an einer der Anmeldekonsolen anmelden, die auch ohne grafische Oberfläche zur Verfügung stehen.

Weil er wie eine Schale den Kernel umgibt, nannten die UNIX-Entwickler den Kommandointerpreter "Shell" (engl. für Schale).



Die Bedienung einer Shell ist unkompliziert, bietet jedoch fast unbegrenzte Möglichkeiten. Wie Sie im Beispiel rechts sehen, führt die Eingabe eines einfachen Befehls mit der abschließenden Bestätigung mit **RETURN** zu einer Aktion des Betriebssystems.

Der Befehl `ls` z. B. listet Ihnen Dateien und Verzeichnisse auf. In Zeile 2 bis 3 erhält der Benutzer diese Liste.

Aber auch komplexere Operationen sind auf einfache Art möglich: Der Befehl `mail` im Beispiel rechts fragt Mailboxen nach einem E-Mail-Eingang ab, und auch grafische Anwendungen wie der Internetbrowser Firefox können von einer Befehlszeile aus gestartet werden.

Arbeiten mit der Shell

Das Aussehen der Konsole können Sie im Menü **EINSTELLUNGEN** ändern. Im Beispiel ist das Erscheinungsbild "SCHWARZ AUF WEISS" eingestellt.

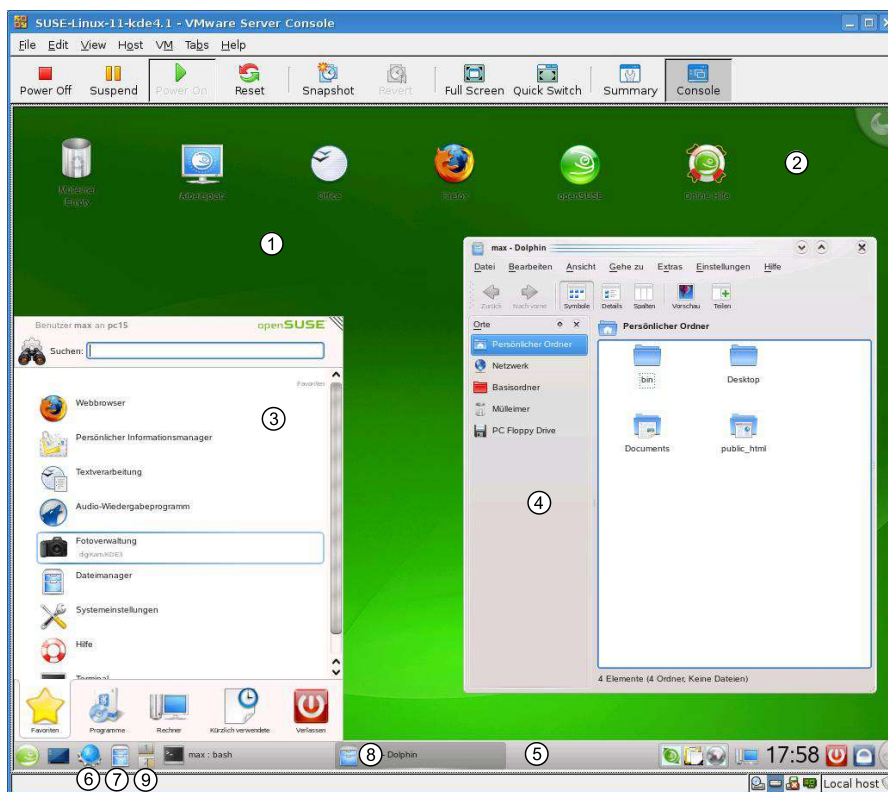
Zusätzlich zu ihrer Funktion als Kommandointerpreter stellt die Shell auch eine Programmiersprache dar, mit Variablen und ähnlichen Kontrollstrukturen, wie sie in anderen Programmiersprachen bekannt sind. Programme, die durch Shellprogrammierung entstanden sind, werden Shellskripte genannt. Die Shell bietet als Programmiersprache neben den Befehlen weitere wichtige Programmierelemente, wie z. B. Variablen, Schleifen, Wiederholungen, Bedingungen, Funktionsaufrufe, Alias- und Historiefunktion.

Die grafische Oberfläche KDE

Befehlszeilen sind nützlich, schlank und bieten fast unbegrenzte Funktionalität. Benutzerfreundlicher, intuitiver und mittlerweile Standard sind jedoch grafische Benutzeroberflächen (GUIs, Grafical User Interfaces). Das heute verbreitete Konzept von Fenstern (Windows), Symbolen (Icons) und Menüs (Menus) wurde von XEROX in den 1970er-Jahren entwickelt und von Firmen wie Apple mit dem Macintosh™ 1984 und später von Microsoft mit dem Betriebssystem Windows™ bekannt. Einer der Vorteile von Linux ist, dass es eine strikte Trennung zwischen Betriebssystem und grafischer Oberfläche gibt. Deshalb kann Linux auch komplett ohne eine grafische Oberfläche voll funktionsfähig betrieben werden.

Unter Linux gibt es eine schier uferlose Vielfalt an grafischen Oberflächen, die Sie auch alle unter SuSE Linux verwenden können. Standardoberfläche ist jedoch KDE (mittlerweile schon in der Version 4), die auch bei der Installation von SuSE Linux 11 OSS komplett installiert und konfiguriert wird. Außerdem steht Ihnen auch noch die Version 3.5 von KDE zur Installation zur Verfügung. Die Bedienung unterscheidet sich im Wesentlichen nicht von der GUIs anderer Betriebssysteme.

In der folgenden Abbildung finden Sie alle wesentlichen Bestandteile der grafischen Oberfläche KDE mit der über das Internet aktualisierten installierten Version 4.1 unter SuSE Linux 11 OSS.



- ✓ ① Arbeitsfläche oder "Desktop"
- ✓ ② Desktopsymbole oder "Icons"
- ✓ ③ Das aufgeklappte Haupt- oder Startmenü
- ✓ ④ Eine gestartete Anwendung, hier das Programm Dolphin
- ✓ ⑤ Die Startleiste mit u. a. folgenden Symbolen
- ✓ ⑥ Das Schnellstart-Icon des Konqueror-Internetbrowsers
- ✓ ⑦ Das Schnellstart-Icon des Dolphin- Dateimanagers
- ✓ ⑧ Umschalter zwischen den laufenden Programmen
- ✓ ⑨ Umschalter zwischen den Arbeitsoberflächen

Anmelden und Abmelden

Vor jeder Arbeit an einem Linux-System steht die Authentifizierung (Login) durch eine Anmeldung mit Loginname und Passwort, wenn Sie bei der Installation nicht die automatische Anmeldung einstellen (was höchstens dann sinnvoll ist, wenn der Rechner in einem abschließbaren Raum steht).

Da im System viele Daten in Zwischenspeichern abgelegt sind, sollte vor dem Ausschalten das System immer sauber heruntergefahren werden. Unter dem Startmenü von KDE erhalten Sie unter dem Punkt VERLASSEN folgende Varianten:

- ✓ Abmelden: Sie melden sich ab, das System bleibt aktiv. Ein anderer Benutzer kann sich danach anmelden.
- ✓ Sperren: Der Bildschirm bleibt bis zur Eingabe des eigenen Passwortes gesperrt.
- ✓ Benutzer wechseln: Eine parallele Anmeldung auf einem anderen XWindow-Desktop ist damit möglich. (Alternativen: Tastenkombination **STRG ALT F8** bzw. **STRG ALT F9**) Mit der Tastenkombination **STRG ALT F7** wechseln Sie zum ursprünglichen Desktop.
- ✓ Suspend to RAM: Der Status des Systems wird dabei in den Hauptspeicher geschrieben, und danach wird das System bis auf den Hauptspeicher zum Stromsparen in den Ruhezustand gebracht. So kann es sehr schnell wieder ohne Booten aktiviert werden. In der Regel ist dies nur bei bestimmter Hardware wie bei Notebook-Rechnern erfolgreich.
- ✓ Suspend to Disk: Der Hauptspeicher wird als komprimiertes Abbild in die Swap-Partition geschrieben, und der Rechner wird ausgeschaltet. Dieser Vorgang wird auch als Tiefschlaf oder Hibernation ("Überwinterung") bezeichnet. Die Swap-Partition sollte dazu mindestens so groß wie der Hauptspeicher sein. Beim erneuten Einschalten wird schneller als beim normalen Booten der genaue Zustand des Systems so wiederhergestellt, wie er vor dem Abschalten vorhanden war. Bei diesem Vorgang erhalten Sie keine Möglichkeit, ein eventuell anderes Betriebssystem zu starten. Nicht jede Hardware ist hierfür geeignet; neue Rechner haben dabei aber kaum noch Probleme.
- ✓ Herunterfahren: Das System wird heruntergefahren und der Rechner danach ausgeschaltet.
- ✓ Neustart: Das System wird heruntergefahren und der Rechner danach neu gestartet.

2.3 Unterschiede zu Windows

Doppelter und einfacher Mausklick

Unter Linux und UNIX hat sich schon seit den 80er-Jahren der einfache Mausklick zum Start eines Programms eingebürgert. Ein Doppelklick auf ein Symbol startet das zugehörige Programm zweimal. Klicken Sie also immer nur einmal auf Symbole, Menüeinträge etc. Unter KDE4 können Sie mit dem Programm `systemsettings` bzw. den Systemeinstellungen eine ähnliche Bedienung wie unter Windows einstellen.

Virtuelle Desktops und Terminals

Der wohl auffälligste Unterschied zu Microsoft Windows ist die Möglichkeit, mit mehreren Desktops zu arbeiten. Unter KDE bedeutet dies, Sie haben immer mehrere - standardmäßig 2 - Arbeitsflächen oder virtuelle Desktops zur Verfügung, wobei auf jedem beliebig viele Fenster geöffnet sein können und die Fensterpositionen für jeden Desktop einzeln gespeichert werden.

Zusätzlich laufen textbasierte Anmelde-Konsolen immer im Hintergrund: Sie können mit Tastenkombinationen wie **STRG ALT F1** bis **STRG ALT F6** zu diesen Befehlszeilen wechseln und sich an diesen "virtuellen" Terminals, im Textmodus - also ohne grafische Benutzeroberfläche - anmelden. Mit der Tastenkombination **ALT F7** gelangen Sie wieder in die grafische Oberfläche zurück.

Vor jeder Arbeit an einem Linux-System steht die Authentifizierung (Login) durch eine Anmeldung mit Loginname und Passwort. Auch wenn Sie schon an KDE angemeldet sind, müssen Sie sich auf den virtuellen Terminals noch einmal anmelden.



Case Sensitivity

Unter Microsoft-Systemen wird in der Regel keine Unterscheidung zwischen Groß- und Kleinbuchstaben vorgenommen. Die Datei `test.txt` und die Datei `Test.txt` stellen unter Windows eine einzige Datei dar, während es unter Linux/UNIX zwei verschiedene Dateien sind.

Befehlszeile, Skripte und Automatisierung

Die Entwicklung von UNIX war immer eng an die Programmiersprache C angelehnt. Deshalb haben sich als Standard-Benutzerschnittstelle C-basierte Shells mit eigenen Skriptsprachen durchgesetzt. Diese skriptfähigen Befehlszeilen stehen Ihnen auch heute unter grafischen Oberflächen zur Verfügung und dienen dem erfahrenen Systemadministrator in sogenannten Skripten (mehrzeilige Arbeitsanweisungen in der Syntax der jeweiligen Shell) zur Automatisierung und Vereinfachung von beliebigen Abläufen. Sie können damit unter Linux viele Aktionen, die Sie durchführen, auch automatisch ausführen lassen. Alle Programme, auch die der grafischen Oberfläche oder der Netzwerksteuerung, funktionieren nach diesem Konzept.

Konfiguration in Textdateien

Unter Windows wird die Konfiguration des Systems in der Regel in Eingabemasken und grafischen Dialogen durchgeführt. Als Alternative steht zum Teil die Registry-Datei zur Verfügung. Unter Linux stehen Ihnen ebenso grafische Konfigurationsprogramme wie YaST unter SuSE zur Verfügung, doch schreiben diese die Konfiguration nur in Textdateien, die Sie jederzeit

- ✓ von Hand editieren können
- ✓ auf ein anderes System kopieren und dort verwenden können
- ✓ beliebig kopieren und sichern können
- ✓ mit anderen Konfigurationstools bearbeiten können
- ✓ von automatisierten Programmen wie Ihren eigenen Skripten bearbeiten lassen können
- ✓ mit Daten von z. B. Datenbank- oder Verzeichnisservern ergänzen lassen können

Fehlen von Laufwerksbuchstaben und Einhängen, "Mounten" von Datenträgern

Unter UNIX und Linux existieren keine Laufwerksbuchstaben, wie Sie sie von Microsoft Betriebssystemen her kennen. Der Verzeichnisbaum ist hierarchisch gegliedert und hat eine Wurzel namens **Root-** oder **Wurzel-Verzeichnis**, unterhalb dessen die Dateisysteme auf den Datenträgern Ihres Systems in beliebigen Verzeichnissen eingehängt werden können. Eingehängt bedeutet: Der Inhalt dieses Dateisystems wird an dieser Stelle im Verzeichnisbaum dargestellt, nachdem der Datenträger eingebunden wurde. Ist der Datenträger nicht eingebunden, befindet sich an diesem Ort ein lokales (leeres) Verzeichnis. Der Prozess des Einhängens heißt **Mounten** und muss vor jedem Zugriff auf das Dateisystem durchgeführt werden. Der Zugriff erfolgt zur Steigerung der Performance über einen Puffer, der vor dem Entfernen des Datenträgers beim sogenannten Aushängen oder **Unmounten** auf den Datenträger geschrieben wird. Der normale Vorgang z. B. beim Schreiben einer Diskette ist unter Linux also: einlegen - mounten - schreiben - unmounten - entfernen. Dies gilt für alle Datenträger unter Linux, vom USB-Stick bis zum eingebundenen Windows-Laufwerk auf einem Server. Bei SuSE Linux 11.0 OSS erfolgt ein Mounten in der Regel über den Automounter. Siehe hierzu auch Kapitel 16.

Datei- und Benutzerrechte

Linux-Dateisysteme bieten mehr und andere Dateiattribute als die, die Sie unter Windows finden. In Kombination mit Access Control Lists (ACLs) können Sie sich selbst komplexe Rechte-Systeme maßgeschneidert konfigurieren. Als Standard finden Sie unter Linux die Dateiattribute *r* Lesen (read), *w* Schreiben (write) und *x* Ausführen (execute; "betreten" oder "wechseln in" bei Verzeichnissen), für jede der drei Benutzergruppen *u* Benutzer (user), *g* Benutzergruppe (group) und *o* alle anderen Benutzer (others). Dateinamenerweiterungen, die eine Datei automatisch zu einem ausführbaren Programm machen (wie .exe unter Windows), gibt es nicht. Ein Benutzer oder Administrator muss einer Programmdatei grundsätzlich immer erst das Dateiaattribut "ausführbar" *x* zuweisen, damit ein Programm gestartet werden kann.

Alles ist eine Datei - Everything is a file

Unter Linux wird jede Hardwarekomponente, jede Befehlsausgabe, jeder Prozess, alles was im Betriebssystem existiert, wie eine Datei behandelt oder durch eine (oft symbolische) Datei dargestellt. Dadurch können Sie z. B.

- ✓ direkt Daten an Drucker oder Soundkarte "kopieren"
- ✓ bitweise Daten aus Datenträgern lesen und kopieren
- ✓ mit den gleichen Tools in Dateien, der Prozessliste und in der Ausgabe von Programmen nach Mustern wie einem Benutzernamen suchen

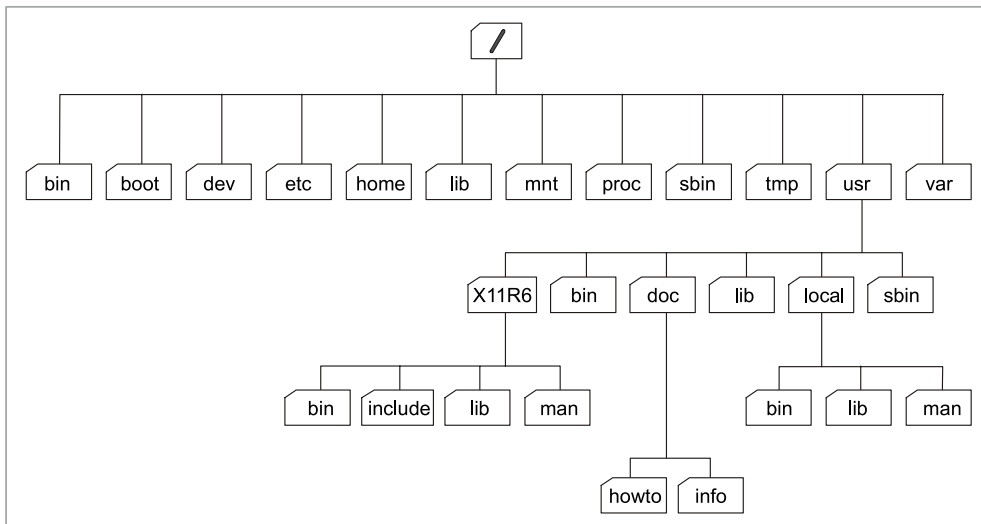
2.4 Technische Konzepte eines UNIX-Systems

Multitasking

Alle Prozesse, die unter Linux ausgeführt werden können, werden auch als Tasks bezeichnet. Unter Linux können mehrere Tasks gleichzeitig bearbeitet werden, das heißt, ein Benutzer kann mehrere Programme oder Jobs gleichzeitig ausführen. Alle unter Linux laufenden Programme werden als Prozesse bezeichnet. Ein Prozess kann also sowohl ein Anwendungsprogramm, eine Aktion, mit der editiert oder kompiliert wird, die Shell selbst oder ein Hintergrundprozess sein. Ein solcher Systemprozess, der im Hintergrund abläuft, wird auch als **Dämon** oder **daemon** bezeichnet. Tasks können entweder aktiv ("running") oder inaktiv ("sleeping") sein. Inaktive Tasks warten entweder auf eine Ressource (z. B. eine Ein- oder Ausgabe), oder ihnen wurde die CPU vom Scheduler entzogen.

Standardhierarchie

Für die Organisation gespeicherter Daten benutzt Linux ein **hierarchisches Dateisystem** (oder file system). Diese Dateistruktur wird durch den Linux Filesystem Standard festgelegt und kann mit einem auf den Kopf gestellten Baum verglichen werden. Oben befindet sich die Wurzel, und von dort aus verzweigen sich immer tiefer nach unten Inhaltsverzeichnisse und Dateien, genau wie die Äste und Zweige eines Baumes.



Der Standard-Verzeichnisbaum unter Linux/UNIX

Durch die Platzierung von Benutzerdateien an bestimmten Stellen in dieser Baumstruktur entsteht eine systematische Gliederung. Dadurch sind die gespeicherten Daten leichter zu lokalisieren und zu verwalten. Diese Standardhierarchie des Verzeichnisbaums ist unter allen UNIX-Systemen ähnlich aufgebaut.

Dateitypen

Unter Linux gibt es verschiedene Typen von Dateien. Die vier wichtigsten sind:

Normale Datei (File)	Ketten von Zeichen bzw. Datenbytes, denen ein Name zugeordnet wurde, unter dem der Anwender seine Daten wiederfindet
Verzeichnis (Directory)	Dateien werden in sogenannten Verzeichnissen abgelegt, die aber ihrerseits wiederum Unterverzeichnisse mit weiteren Dateien enthalten können. Verzeichnisse können theoretisch beliebig tief geschachtelt werden. Jedes Verzeichnis enthält mindestens zwei Einträge, nämlich das Symbol ".", das einen Verweis auf das Verzeichnis selbst darstellt, und das Symbol "..", einen Verweis auf das übergeordnete Verzeichnis.
Geräte-datei (Special File)	Alle Geräte, die unter einem Linux-System angeschlossen sind, also die gesamte Hardware, können über spezielle Dateien, die sogenannten Geräte-dateien angesprochen werden. Sie bilden die Schnittstelle zu den im Kernel geladenen Gerätetreibern. Jedes Gerät hat einen Eintrag im Verzeichnis /dev in Form eines Gerätenamens.

Symbolische Links

Mithilfe eines symbolischen Links wird eine neue Datei angelegt, die auf eine bereits vorhandene Datei oder ein Verzeichnis verweist. Dadurch besteht die Möglichkeit, diese Datei unter verschiedenen Namen anzusprechen. Dazu muss der Dateinhalt nicht kopiert werden, auch wenn die Dateien sich an verschiedenen Stellen im Dateibaum befinden.

Pfadangaben

Mithilfe des Pfadnamens findet der Benutzer alle Unterverzeichnisse und Dateien eindeutig wieder. Der Pfadname setzt sich aus den einzelnen Verzeichnisnamen zusammen, die durch Schrägstriche "/" getrennt sind, z. B. */usr/local/bin*.

Absoluter Pfad

Beschreibt den Weg zu einer Datei oder einem Verzeichnis immer ausgehend vom Hauptverzeichnis. Der absolute Pfad beginnt immer mit einem Schrägstrich (/). Der eigene Standort spielt keine Rolle.

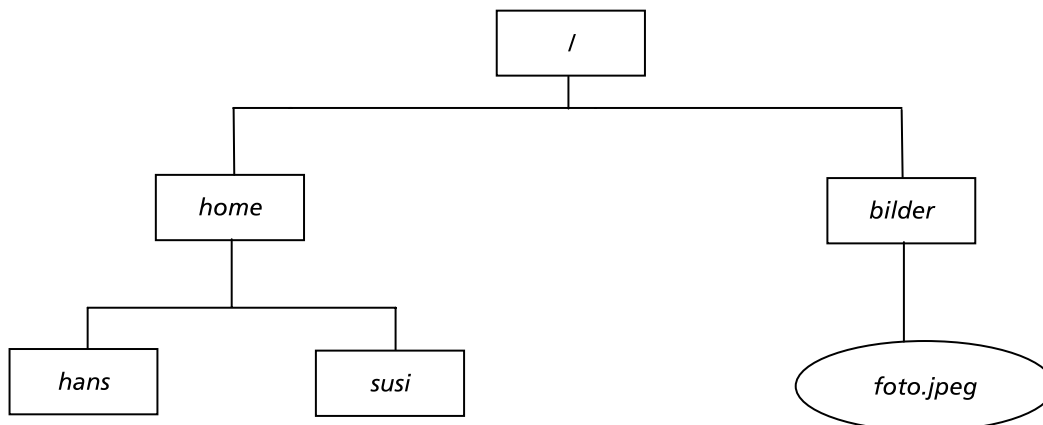
Relativer Pfad

Beschreibt den Weg zu einer Datei oder einem Verzeichnis ausgehend vom aktuellen Standort des Benutzers im Dateisystem

Bei der relativen Pfadangabe ist außerdem zu beachten:

- ✓ Wird das übergeordnete Verzeichnis angesprochen, geschieht dies mit zwei Punkten (..).
- ✓ Wird das aktuelle Verzeichnis angesprochen, geschieht dies mit einem Punkt (..).

Im folgenden Beispiel soll der Unterschied zwischen dem absoluten und dem relativen Pfad am Beispiel des Kommandos `rm`, das zum Löschen von Dateien benutzt wird, aufgezeigt werden. Ihr aktueller Standort ist */home/hans*. Sie wollen die Datei *foto.jpeg* im Verzeichnis */bilder* löschen.



Absoluter Pfad: `rm /bilder/foto.jpeg`

Relativer Pfad: `rm ../../bilder/foto.jpeg`

2.5 Benutzer und Rechte

Multiuserfähigkeit

Linux ist ein Multiusersystem. Mehrere Benutzer können unter Linux unabhängig voneinander und zur gleichen Zeit dasselbe Rechnersystem verwenden. Dazu bedienen diese Anwender entweder ein an das System angeschlossenes Terminal oder werden über einen anderen Rechner (z. B. einen Windows- oder Linux-PC) mit dem System verbunden. Durch Timesharing und Multitasking wird für jeden Benutzer der Eindruck erzeugt, als einziger Benutzer am System zu arbeiten. Wesentlicher Bestandteil einer solchen Betriebssystem-Architektur ist der Scheduler (das Hauptsteuerprogramm), der die zeitliche Verteilung der Rechnerzeit für die einzelnen Benutzer steuert und überwacht.

Der Systemverwalter root (Superuser)

Um am System arbeiten zu können, benötigt ein Anwender einen Benutzernamen und ein Passwort. Der Benutzername und ein bestimmter Bereich im System für die Datenablage des Benutzers, normalerweise das Verzeichnis `/home/Benutzername`, werden in der Regel durch den Administrator eingerichtet.

Der Administrator, der auch Superuser oder root genannt wird, hat eine besondere Rolle im System:

- ✓ Er wird bei der Installation automatisch eingerichtet.
- ✓ Er besitzt alle Rechte im System.
- ✓ Er kann weitere Benutzer einrichten und verwalten.
- ✓ Er kann Systemdaten sichern, ändern und vieles mehr.

Es gibt so gut wie nichts, was ein Superuser unter Linux nicht tun darf; daher ist es wichtig, Umsicht walten zu lassen, sobald Sie sich als root angemeldet haben. Unter anderem können Sie als Superuser das ganze System löschen - auch versehentlich.

Zugriffsrechte

Unter Linux gibt es für Dateien und Verzeichnisse zugewiesene Rechte. Damit werden Informationen vor unberechtigtem Zugriff oder vor Beschädigung durch Dritte geschützt. Nach dem Anmeldevorgang landet der berechtigte Anwender dann in seinem Arbeitsverzeichnis oder Homedirectory. Innerhalb des Systems gibt es aber noch weiter gehende Regelungen, um Dateien und Verzeichnisse eines Benutzers zu schützen. Dabei kann jeder Benutzer die Rechte an seinen Dateien und Verzeichnissen vergeben, da er der Besitzer dieser Daten ist.

Es gibt drei Arten von Zugriffsrechten, die bei Dateien und Verzeichnissen aber eine unterschiedliche Bedeutung haben können:

Recht	Datei	Verzeichnis
r (read) = Leserecht	Der Inhalt der Datei darf gelesen und kopiert werden.	Der Inhalt des Verzeichnisses darf (z. B. mit dem Befehl <code>ls</code>) gelistet werden.
w (write) = Schreibrecht	Der Inhalt einer Datei darf verändert, ergänzt oder gelöscht werden.	Der Inhalt eines Verzeichnisses darf geändert werden, d. h., es dürfen Dateien oder Verzeichnisse neu angelegt, umbenannt oder gelöscht werden.
x (execute) = Ausführungsrecht	Die Datei kann als Befehl ausgeführt werden, indem ihr Name aufgerufen wird (vorausgesetzt, es handelt sich auch um eine ausführbare Datei).	Wenn das Ausführungsrecht bei einem Verzeichnis gesetzt ist, bedeutet es, dass es möglich ist, mit dem Befehl <code>cd</code> in dieses Verzeichnis zu wechseln.

Diese Zugriffsrechte können den drei verschiedenen Benutzerklassen zugeteilt werden:

Bezeichnung	Bedeutung
u (user)	Besitzer (auch owner genannt)
g (group)	Gruppe, zu der die Datei (das Verzeichnis) gehört
o (other)	Alle übrigen Benutzer des Systems

Die Rechte an Dateien und Verzeichnissen werden beim Anlegen mit einer Standardvoreinstellung vergeben. Der Besitzer kann anschließend die Rechte für jede Gruppe neu festlegen. Nur der Systemadministrator kann die Rechte von Dateien und Verzeichnissen ändern, die ihm nicht gehören.

Der Befehl `ls -l` listet Ihnen alle Dateien des aktuellen Verzeichnisses in einer ausführlichen Liste auf. In der ersten Spalte der Ausgabe sehen Sie Dateityp und Dateirechte in 10 Zeichen dargestellt.

Als erstes Zeichen steht:

- ✓ - für normale Datei
- ✓ d für directory, also Verzeichnis
- ✓ l für link
- ✓ b oder c für eine block- oder zeichenorientierte Gerätedatei
- ✓ p für FIFO (Named Pipe)
- ✓ s für Sockets

Die Zeichen 2 - 10 stehen in Dreiergruppen für die Rechte der unterschiedlichen Benutzerklassen in der Reihenfolge User - Group - Others:



Der User *mmuster* darf das Verzeichnis *Documents* in der vierten Zeile der Ausgabe lesen, schreiben und betreten (*rwx*), Mitglieder der Gruppe *users* dürfen das Verzeichnis lesen und betreten, aber nicht schreiben (*r-x*), ebenso alle anderen User. Näheres zu den Rechten finden Sie im Kapitel 17.

Rechte, die für ein Verzeichnis gesetzt sind, haben immer eine höhere Priorität als Rechte, die für eine Datei in diesem Verzeichnis gesetzt sind. Wenn Sie für ein Verzeichnis nur Leserecht und für eine darin befindliche Datei das Schreibrecht haben, können Sie dieses Schreibrecht nicht in Anspruch nehmen, da das Recht des Verzeichnisses eine höhere Priorität besitzt.

```
mmuster@linux:~> ls
Beide_Zeilen  Desktop  hallo.txt  public_html  temp  Zeile_1
bin          Documents  mbox      screenshots  tools  Zeile_2
mmuster@linux:~> ls -l
insgesamt 20
-rw-r--r-- 1 mmuster users  34 2005-10-18 12:50 Beide_Zeilen
drwxr-xr-x 2 mmuster users  48 2005-10-11 18:54 bin
drux----- 2 mmuster users 320 2005-10-11 18:59 Desktop
drwxr-xr-x 2 mmuster users  80 2005-10-11 18:54 Documents
-rw-r--r-- 1 mmuster users  37 2005-10-18 20:14 hallo.txt
-rw----- 1 mmuster users 3280 2005-10-19 12:41 mbox
drwxr-xr-x 2 mmuster users  80 2005-10-11 18:54 public_html
drwxr-xr-x 6 mmuster users 144 2005-10-17 17:09 screenshots
drwxr-xr-x 3 mmuster users 192 2005-10-18 13:02 temp
drwxr-xr-x 6 mmuster users 224 2005-10-18 12:40 tools
-rw-r--r-- 1 mmuster users  17 2005-10-18 12:50 Zeile_1
-rw-r--r-- 1 mmuster users  17 2005-10-18 12:50 Zeile_2
mmuster@linux:~>
```

Kurze und Ausführliche Dateiliste im Home-Verzeichnis des Users *mmuster*

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.