

Linux für Einsteiger

Teil 1

Das ist Linux – Erklärung der Entstehung und der Besonderheiten von Linux

Linux ist der Kernel* eines Betriebssystems, das aus Open Source* Programmen besteht. Ausgehend von der Initiative Richard Stallman's zur Entwicklung von freier und jedermann zugänglicher Software war es Linus Thorvalds, der Anfang der 90-er Jahre des 20. Jahrhunderts den Anstoß zur weltweiten Entwicklung eines Kernels passend zu den bereits vorhandenen freien Programmen, wie z.B. Editor*, Compiler*, Debugger* usw., gab.

Linux, wie wir es heute kennen, ist aus den Betriebssystemen der UNIX® Gruppe abgeleitet worden, es sind dabei viele Arbeitsweisen und Sicherheitsaspekte übernommen worden, die aus dem Großrechnerbereich stammen (Multiuser*- und Multitasking*-Fähigkeit). Zwangsläufig ist damit auch eine Entscheidung über die grundlegende Bedienung des Systems gefallen. Eine Benutzerverwaltung, die unterschiedlichen Nutzern differenzierte Nutzungsrechte gibt, ist zwar für einen Einzelplatzrechner mit nur einem Benutzer auf den ersten Blick überflüssig, der Vorteil ist jedoch, und damit auch die Philosophie die hinter Linux steht, dass die Beschränkung der Nutzungsrechte auf den geringstmöglichen Umfang (nur so viel wie nötig) ermöglicht wird. Damit können Bedienfehler, falsche Kommandos und auch andere Störfaktoren (z.B. Viren) nicht das gesamte System zerstören, es sind nur die Daten zu beeinflussen, für die das Recht zur Bearbeitung für den Nutzer vergeben wurde.

Freie Programme, die für ein vollständiges Betriebssystem nötig sind, und freie Anwendungsprogramme gibt es unzählig viele – wie soll sich ein Einzelner (Einsteiger) da zurechtfinden ?

Hier schlägt die Stunde der Distributionen*. Sinnvolle Zusammenstellungen von Programmpaketen gibt es viele (kommerzielle und nichtkommerzielle). Das für manche Distributionen Geld gezahlt werden muss, ist kein Widerspruch zu Open Source und freien Programmen. Sie bezahlen nämlich nicht für die Programme und deren Nutzung (Lizenz) an sich (ist in der GPL* untersagt), sondern für die Leistung der Zusammenstellung der Programme, der Herstellung der Datenträger und eventueller Handbücher und Telefonsupport. Ebenfalls ist es möglich, mit Hilfe der Open Source Programme neue Programme zu entwickeln, die nicht frei sind und kommerziell vertrieben werden können (Beispiel: MainActor – ein Videobearbeitungsprogramm).

Linux eignet sich durch seinen modularen und strukturierten Aufbau gut für ein Betriebssystem, welches überwiegend auf einem nur lesbaren Medium residiert → Stichwort Live-CD*. Das Linux Dateisystem, auf das wir später eingehen werden, macht dies einfach, da nur bestimmte

Verzeichnisse bzw. Dateien veränderbar sein müssen. Die Wahl einer Distribution für den Einstieg fiel deshalb auf die KNOPPIX-CD Version 3.7. Der Vorteil dieser Version ist die Lauffähigkeit auf fast allen Computersystemen mit x86*-Architektur. Bei Startschwierigkeiten hat der Entwickler Klaus Knopper etliche Cheat-Codes* vorgesehen, die bestimmte Einstellungen verändern und so trotzdem einen reibungslosen Betrieb ermöglichen. Die in diesem Kurs verwendeten Exemplare sind nach der Unterrichtseinheit für zu Hause gedacht, jeder kann damit auf seinem Rechner auch ohne vorhandene Linux Installation die Beispiele nachvollziehen und eigene Erkundungen durchführen. Zusätzlich leistet die CD auch wertvolle Dienste, falls es wegen Problemen auf einem Rechner nicht mehr möglich ist, Betriebssysteme von der Festplatte aus zu starten (Notfall-CD).

Knoppix startet ohne weitere Eingaben in einen graphischen Bildschirm, der gewisse Ähnlichkeiten mit einem gestarteten Windows® System hat. Dies soll Ein- und Umsteigern den Anfang erleichtern. Wir werden im Kursverlauf Alternativen kennenlernen, Linux ist frei konfigurierbar und zwingt dem Nutzer keine bestimmte Graphikoberfläche und deren Bedienung auf.

Das gestartete Knoppix erlaubt es uns, einen Blick auf ein Kernstück des Linux Systems zu werfen:

Der Verzeichnisbaum von Linux – oder – was finde ich wo ?

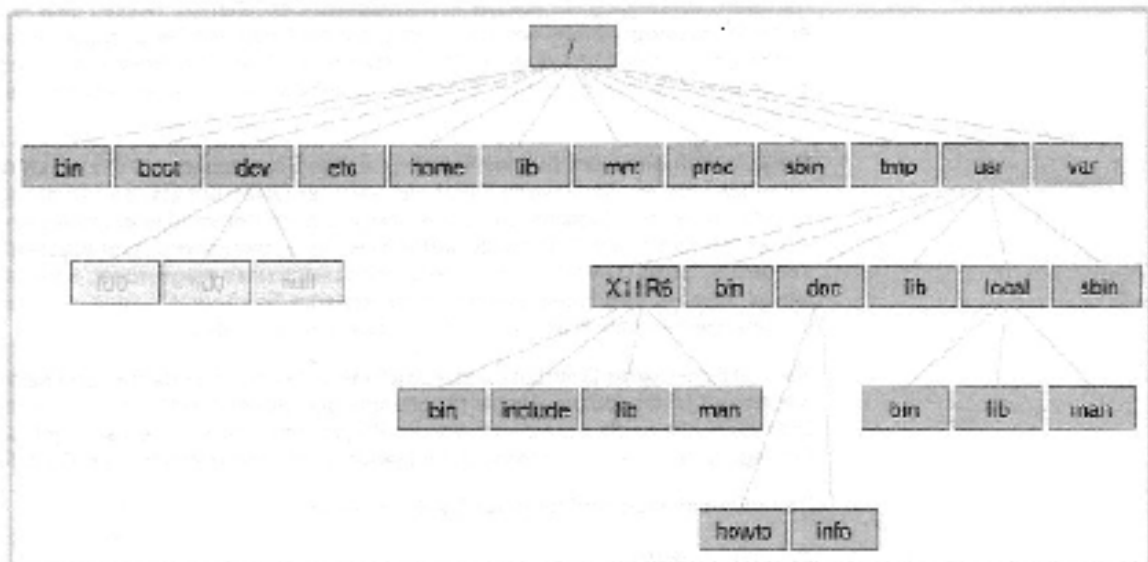


Abbildung 1: Die Linux-Verzeichnisstruktur

Das Linux Verzeichnissystem ist hierarchisch aufgebaut, ausgehend von einem Hauptverzeichnis, bezeichnet als „Root“ (dtsch. Wurzel) und gekennzeichnet durch einen „Slash“ /

Hier liegt die Stolperfalle für eingefleischte W*****-Nutzer. In Linux wird durchgehend sowohl für das Rootverzeichnis als auch als Verzeichnistrenner der Slash / und nicht der „Backslash“ \ benutzt !

Ein Blick auf die Unterverzeichnisse und deren Bedeutung:

/bin

Hier sind die wichtigsten Kommandos enthalten, um mit dem System zu arbeiten. Jeder Nutzer kann darauf zugreifen (Beispiele: ls, mkdir, ...).

/boot

Hier sind der oder die Linux Kernel zu finden, ebenso die Dateien des Bootmanagers*.

/dev

Hier finden sich alle Treiber zur Ansteuerung der Hardware (Un*x bzw. Linux Philosophie – alles ist eine Datei). Geräte werden durch Dateien repräsentiert.

/etc

enthält alle Konfigurationsdateien, sie sind im Klartext lesbar und mit einem Editor veränderbar (Rechtezuteilung vorausgesetzt).

/home

Hier sind alle Heimatverzeichnisse der Nutzer zu finden. Nur in seinem Heimatverzeichnis hat der einzelne Nutzer uneingeschränkte Dateirechte (lesen, schreiben, löschen, erstellen von Dateien und Verzeichnissen).

/lib

Hier sind die zum Systemstart nötigen Bibliotheken (engl. libraries) zu finden, genauso wie die nachladbaren Kernelmodule (in einem Unterverzeichnis).

/mnt

Hier können weitere Verzeichnishierarchien eingehängt (gemounted) werden. Der „Mount“-„Unmount“-Mechanismus von Un*x-Systemen wie Linux lässt praktisch ein beliebig erweiterbares Verzeichnissystem entstehen. Es gibt keine Beschränkungen bzgl. Größe und Zahl der Einträge. Vorteil: keine Begrenzung durch Zuordnung von Laufwerksbuchstaben (max. 26). Nachteil: auch Wechselmedien (Floppy, CD, DVD usw.) müssen vor Benutzung gemounted werden und dürfen nicht einfach aus dem System entfernt werden ohne unmount!

/root

Das Heimatverzeichnis des Administrators (root). Es wird nicht als Unterverzeichnis von /home geführt, da auch dann noch ein Arbeiten mit dem System möglich sein soll, wenn die evtl. auf einem anderen Datenträger

gespeicherten Homeverzeichnisse der normalen Nutzer nicht zur Verfügung stehen.

/sbin

Systemprogramme, die „Root“-Rechte erfordern (Beispiele: fdisk, init, shutdown)

/tmp

temporäre Dateien aller Nutzer werden hier gespeichert.

/usr

(Un*x-System-Ressources) ist die zweite Hierarchieebene des Systems. Im großen und ganzen werden hier die gleichen Unterverzeichnisse wie im Hauptverzeichnis erstellt mit ähnlicher Bedeutung. Zusätzlich:

| | |
|----------|--|
| /X11R6 | Programmhierarchie des Graphiksystems von Linux. |
| /include | Header-Dateien (werden für Programmherstellung aus Quelltexten gebraucht) |
| /local | dritte Hierarchieebene. Hier können beliebige Programmpakete abgelegt werden, die nicht von der Systemverwaltung erfasst werden. |
| /share | Architektur-unabhängige Dateien. |
| /src | Quelldateien zu Programmen (engl. source) |

/var

Variable Daten, die ständig verändert werden oder nur zeitweise existieren, werden unter diesem Verzeichnis abgelegt. Einige Unterverzeichnisse:

| | |
|--------|---|
| /cache | Zwischenspeicher von Programmen |
| /lock | Sperren (Datei, Geräte, usw.) |
| /log | Protokolle des Systems |
| /mail | Mailboxen der Nutzer |
| /run | Dateien zu laufenden Prozessen |
| /spool | Von Anwendungen zwischengespeicherte Daten, für die Druckerausgabe. |

z.B.

/opt

Ein von Linux eingeführtes Verzeichnis für große Programmpakete, die nicht zum üblichen Umfang einer Installation gehören. Fast alle kommerziellen Softwarepakete werden hier installiert, bei manchen Distributionen werden auch die Pakete für die Graphikoberfläche hier gespeichert.

/proc

enthält Kernel- und Prozessinformationen in einem virtuellen Dateisystem (verbraucht keinen Platz auf der Festplatte)

/sys

beim neuen Kernel (ab 2.6.x) für Systeminformationen

Wo werden die Dateien und Verzeichnisse physikalisch gespeichert ?

Nur die für einen ordnungsgemäßen Start des Linux Systems nötigen Dateien müssen unmittelbar vom Root-Dateisystem erreicht werden können, d.h. auf einem Speichermedium zusammenhängend verfügbar sein, z.B. eine Partition einer Festplatte. Sobald das grundlegende Linux System läuft, können weitere Verzeichnisse mit ihren Dateien über den Mount-Mechanismus hinzugefügt werden. In der Praxis können so Verzeichnisse und Unterverzeichnisse auf beliebige Datenträger verteilt werden. Zweckmäßig ist selbst auf Einzelplatzrechnern, eine getrennte Partition für das Homeverzeichnis vorzusehen. Der Vorteil: eine eventuelle Neuinstallation muss nicht die eigenen gespeicherten Daten beeinflussen. Umgekehrt kann eine volle Homepartition das System nicht zum Absturz bringen.

Linux Dateisysteme – welche gibt es ?

historisch: minix, extfs aus den Anfangstagen von Linux

- ext2fs - Standard unter Linux
- ext3fs - Weiterentwicklung ext2fs mit Journal
- reiserfs - für Linux entwickeltes Journaling*-Filesystem
- xfs, jfs - weitere Journaling-Filesysteme

weitere Dateisysteme, die Linux lesen und schreiben kann:

- msdos (fat) - DOS/W*****s Filesystem
- vfat - W*****s Filesystem, erweitert
- ntfs - neues W*****s Filesystem, noch nicht vollständig unterstützt (Schreiben experimentell, keine verschlüsselten Dateien)
- iso9660 - Dateisystem auf CD's
- nfs - Netzwerk-Filesystem, von Linux bevorzugtes System für im Netzwerk verteilte Dateien
- smbfs - Netzwerk-Filesystem zur Kommunikation mit der W*****s Welt

Warum verwendet Linux nicht fat oder vfat als Standard Dateisystem ?

Fat, bzw. vfat kennt nur 4 Dateiattribute, s=system, h=hidden, r=read-only, a=archiv und eine Zeitangabe, nämlich Datum und Zeit des letzten Schreibzugriffs.

Linux Dateisysteme kennen andere Attribute, unter anderem getrennte Lese-, Schreib- und Ausführungsrechte, getrennt nach Besitzer, Gruppe und übrige Nutzer, sowie drei Zeitangaben (Erstellung, letzte Änderung, letzter Zugriff).

Linux unterscheidet auch streng nach Groß- und Kleinschreibung, d.h. Datei.1 , datei.1 , dAtei.1 sind verschiedene Dateien. Dateinamen dürfen unter Linux 256 Zeichen lang sein, es sind fast alle Sonderzeichen im Namen zugelassen, jedoch wird die Benutzung nicht empfohlen, da es zu Problemen führen kann (z.B. Umlaute, die nicht auf jedem System gleich angezeigt werden, da unterschiedliche Zeichensatzkodierungen verwendet werden. Die Bezeichnung der Datei einschließlich aller Verzeichnisse darf maximal 1023 Zeichen lang sein.

Die Bedeutung des Punktes „.“ in Linux-Namen. Prinzipiell können beliebig viele Punkte in den Namen vorhanden sein, eine Sonderbedeutung hat nur der Punkt am Namensanfang

.XXXX die Datei wird normalerweise nicht angezeigt
.
.. steht für das übergeordnete Verzeichnis

Genauerer folgt später im Abschnitt Sicherheit und Dateirechtevergabe.

Die ersten praktischen Erfahrungen mit Linux

Das euch zur Verfügung gestellte Knoppix startet ohne weitere Nachfragen bis zur graphischen KDE-Oberfläche. Nur wenn das nicht funktioniert, sind zusätzliche Angaben beim Start der CD zu machen. Die Bedienung der meisten Elemente erfolgt intuitiv, ihr werdet damit zu recht kommen, wenn euch W****s vertraut ist. Einige Abweichungen sind jedoch schon auf den ersten Blick zu sehen:

KDE verwendet eine virtuelle Fensterverwaltung und kann somit die Anwendungen auf mehrere Bildschirmseiten verteilen.

Maus- oder Tastaturbedienung ?

Etliche Aktionen lassen sich sowohl mit der Maus, als auch mit der Tastatur erledigen, neben den Standardeinstellungen lassen sich auch eigene Tastenkombinationen vorgeben. Manchmal wird dadurch die Bedienung vereinfacht, und bei nicht funktionierender Maus dennoch die vollständige Bedienung des Systems ermöglicht.

Einfach- oder Doppelklick ?

Linux sieht standardmäßig den Einfachklick zur Ausführung von Befehlen vor, standesgemäß läßt Linux dem Benutzer jedoch die Wahlfreiheit, auf sein vielleicht gewohntes Klickverhalten zurückzukehren.

Wir wollen nun einmal eine neue Datei in einem neuen Verzeichnis anlegen, bearbeiten, ansehen und wieder alles löschen, und zwar sowohl mit graphischer Bedienoberfläche, als auch in der Textkonsole.

Der Dateimanager der KDE-Oberfläche ist der Konqueror. Die Bedienung ist nach Übung einfach, die gestellte Aufgabe lässt sich mit kombinierter Maus- und Tastaturbedienung erledigen.

Die Textkonsole rufen wir bei gestarteter KDE-Oberfläche aus der Kontrollleiste auf. Ohne die Kenntnis einiger Befehle geht es nicht weiter. Wir werden daher nach und nach die nötigen Befehle kennenlernen, damit keiner von der Vielfalt der möglichen Kommandos erschlagen wird.

Hier eine Auflistung der Befehle, die wir verwenden wollen:

pwd, ls, mkdir, cd, touch, edit, mcedit, cat, cp, mv, rmdir, rm

Wo sind die Hilfeseiten, die die Anwendung dieser Befehle erklären ?

Die Hilfetexte ruft man gewöhnlich über **man (befehl)** auf. Hier wird dann beschrieben, wie der Befehl anzuwenden ist. Leider ist nicht alles auf deutsch beschrieben (die Hauptsprache ist nun mal Englisch). Was kann man tun, wenn es keine Beschreibung gibt ? Die Suche nach Lösungen wird uns im Kurs immer wieder mal beschäftigen.

Zuerst schauen wir mal, wo wir uns im Verzeichnisbaum befinden:

| | |
|-------------------------------|---|
| pwd (enter) | Print Working Directory Anzeige des aktuellen Verzeichnisses |
| ls (enter) | list Anzeigen der vorhandenen Dateien und Verzeichnisse im aktuellen Verzeichnis |
| mkdir testverzeichnis (enter) | make directory Erstellen des neuen Verzeichnisses mit Namen „testverzeichnis“ |
| cd testverzeichnis (enter) | change directory Wechseln in das Verzeichnis „testverzeichnis“, es ist nun das aktuelle Verzeichnis |
| touch testdatei.txt | Setze das Änderungsdatum der Datei neu. Der Befehl wird hier dazu missbraucht, eine neue, leere Datei anzulegen. |
| mcedit testdatei.txt | Bearbeiten der Datei mit dem Editor aus dem Midnight Commander. Der Umweg über die Erstellung einer leeren Datei ist hierbei nötig, da mcedit keine Dateien erzeugen kann, sondern nur vorhandene bearbeiten. |

Alternativ lassen sich die beiden vorangegangenen Befehle auch durch den Aufruf von „editor testdatei.txt“ ersetzen. Die Bedienung des Editors aus UN*X Urzeiten ist jedoch nur etwas für Hartgesottene.

| | | |
|--------------------------------|--------------------------|---|
| cat testdatei.txt | concatenate files | Dateien aneinanderhängen und auf der Standardausgabe ausgeben |
| mv testdatei.txt dateitest.txt | move files | Datei verschieben oder umbenennen, ist das Ziel eine Datei, wird die Datei umbenannt, ansonsten wird die Datei in das neue angegebene Verzeichnis verschoben. |
| rm dateitest.txt | remove file or directory | löscht die Datei dateitest.txt |
| rmdir testverzeichnis | remove empty directory | löscht das Verzeichnis, wenn es leer ist. Für nicht leere Verzeichnisse ist rm zuständig. |

Das soll für den 1. Teil erstmal genug sein. Im Anhang ist ein Fragenkatalog, der bitte beantwortet das nächste Mal mitgebracht wird.

Erklärungen:

Kernel - Grundprogramm des Systems, teilt den verschiedenen Anwendungsprogrammen Rechenzeit und Arbeitsspeicher zu, regelt den Zugriff auf Geräte und Speichermedien und trennt sicherheitsrelevante Bereiche voneinander.

OpenSource - Programme, deren Quellcode offengelegt ist und der frei verwendbar ist

Editor - Programm, um Textdateien zu bearbeiten

Compiler - Programm, das Textdateien (Quellcode) in binäre, ausführbare Programme übersetzt. Binäre Programme können viel schneller ausgeführt werden als Scripte, die jedesmal interpretiert werden müssen.

Debugger - Programm, das ein anderes Programm überwacht starten, pausieren, stoppen kann. Der Entwickler des Programms kommt so Programmierfehlern auf die Spur.

Multiuser - Das Betriebssystem kann mehrere verschiedene Benutzer gleichzeitig am Rechner arbeiten lassen.

Multitasking - Der Rechner kann gleichzeitig mehrere verschiedene Programme ablaufen lassen.

Distribution - Die Zusammenstellung eines lauffähigen Betriebssystems mit ausgewählten Anwendungsprogrammen. Der Hersteller sorgt für einen problemloses Zusammenspiel aller Komponenten.

GPL - General Public Licence

Die unter Linux und GNU-Programmen meistbenutzte Lizenzform, die dafür sorgt, dass einmal freigegebene Programme unter dieser Lizenz unbeschränkt weiter bearbeitet und verteilt werden können.

Live-CD - Distributionsform eines Betriebssystems, dass sich ausschliesslich von CD betreiben lässt. Eine Festplatte ist nicht erforderlich.

x86 - diese Kombination steht stellvertretend für eine Microprozessorfamilie der Firma Intel (8088,8086,80186,80286,80386,...,Pentium I,II,III,IV usw.) Die Maschinensprachbefehle dieser Baureihe haben sich als Standard durchgesetzt und werden auch von anderen Herstellern benutzt,z.B. AMD.

Cheat-Codes - Eingaben auf der Kommandozeile, die bestimmte Einstellungen vornehmen und den Programmablauf beeinflussen.

Bootmanager - Programm zum Starten eines Betriebssystems, wobei eine Auswahl aus mehreren Systemen erfolgen kann.

Journaling-Filesystem - Dateisystem, das die Dateibewegungen (ändern, speichern, anlegen, löschen) mitverfolgt und getrennt protokolliert. Bei einer plötzlichen Betriebsstörung (Stromausfall, Hardwaredefekt, Absturz !?) kann anhand des Protokolls festgestellt werden, welche Datei(en) nicht korrekt bearbeitet wurden. Das Dateisystem kann in kürzester Zeit wieder in einen gebrauchsfähigen Zustand gebracht werden.