

# Vorwort

Linux ist inzwischen als PC-Betriebssystem etabliert. Große Unternehmen setzen es ebenso ein wie zahlreiche Internet-Provider. Aber immer mehr Unternehmen nutzen die Fähigkeiten von Linux auf ganz andere Weise. Sie verstecken den Linux-Rechner in einer mehr oder weniger hässlichen Blech- oder Kunststoffkiste und betrauen ihn mit für den Normalanwender ungewöhnlichen Aufgaben, nämlich als Firewall, Stageserver, DSL-Router oder gar zu Steuerungszwecken für Kaffee- oder Sandwichautomaten. Tastatur und Maus fallen unter den Tisch, ebenso der Monitor; es gibt nur noch ein paar Knöpfe, ein Minidisplay und einen Münzprüfer. Bei der Ausgabe geht es ähnlich spartanisch zu, einige Relais und Motoren – sonst nichts. Es handelt sich somit um „eingebettete Systeme“, englisch „Embedded Systems“, für die es zahlreiche weitere Anwendungen gibt.

Als die PCs noch 386er-Prozessoren besaßen und das Betriebssystem MS-DOS hieß, wagten sich viel mehr Menschen daran, auch mal ein paar Knöpfe und Relais an den PC anzuschließen und mit einem Programm die Modellbahn zu steuern (oder was auch immer). Mit Windows ließ das nach. Vielleicht war das System einfach zu komplex, die Hardware nicht mehr so leicht „erreichbar“, und bis das Relais klapperte, vergingen viele Stunden an Programmierarbeit.

In den Neunzigern kam Linux, erst zaghaft, dann mächtig im Serverbereich und nun – wieder etwas zaghaft, weil man ja so an das Betriebssystem aus Redmont gewöhnt ist – im Desktop-Bereich. Im Gegensatz zu den frühen Versionen wird Linux heute fast nur noch mit der grafischen Benutzeroberfläche bedient. Nur bei Serveranwendungen wird aus Sicherheitsgründen auf das X-Window-System verzichtet, was natürlich auch bei Embedded-Systemen sinnvoll ist. Vielleicht förderte die Distanz zur Kommandozeile auch die Ansicht, hier sei die Sache mit den Schaltern und Relais auch wieder zu kompliziert. Ist es aber nicht!

Zwar verbietet ein Betriebssystem wie Linux, das als Mehrbenutzer- und Multitaskingbetriebssystem konzipiert ist, erst mal den direkten Zugriff auf die Hardware (es wäre auch lustig, wenn zwei Programme gleichzeitig in den Bildschirmspeicher schreiben oder etwas an den Drucker schicken würden), aber mit etwas Mut und Hintergrundwissen kann man Linux schnell zur Kooperation überreden. Manchmal reicht schon ein 10-Zeilen-C-Programm. Wie das funktioniert, werde ich Ihnen in diesem Buch zeigen. Dabei werden Sie in begrenztem Umfang auch einen Ausflug in die Entwicklung von Treibern machen.

Weil heutzutage Computerbenutzer, die mit einem Lötkolben in der Hand auf die Welt gekommen sind, immer seltener werden, kommt auch die Hardware mit fertigen Schaltungen und Anregungen zu deren Erweiterung nicht zu kurz. Etwas Grundwissen sollte schon vorhanden sein (Grundlagen der Elektrotechnik und Elektronik sowie Erfahrungen mit dem Zerlegen des heimischen Videorecorders werden meist reichen). Ab und zu wird auch möglicherweise schon vergessenes Wissen mit eingeschobenen kurzen Grundlagenpassagen aufgefrischt. Weitere Grundlagenartikel sind von der Webseite zum Buch abrufbar. Ich verweise in diesem Zusammenhang auch ausdrücklich auf die Beachtung der einschlägigen VDE-Vorschriften. Für den Einsteiger in die Materie eignet sich besonders das Buch „Elektronik“ von Dieter Zastrow. Wenn Sie jedoch schon wissen, wie man einen Mikrowellengrill zu einer Phaserkanone umbaut, können Sie einige Abschnitte überspringen.

Auch bei der Programmierung gehe ich davon aus, dass Sie die Sprache C zumindest halbwegs beherrschen. Wer C kann, der lernt Perl recht flott (zumindest oberflächlich, bis zum Perl-Guru dauert es dann einige Jahre). Weil diese Sprache gerade beim Gestalten einer Benutzerschnittstelle oder beim Verarbeiten von Zeichenketten viel bequemer als C ist, wird sie eingesetzt. Manchmal bietet sich Anlass für eine Mischung der beiden: der Basiszugriff erfolgt in C, ein Webinterface per CGI dann in Perl.

Welche Linux-Distribution Sie wählen, ist relativ egal. Die Distributionen unterscheiden sich teilweise in den angebotenen Paketen und teilweise in der Verzeichnis- und Konfigurations-Struktur. Einem mit Linux vertrauten Fachmann sollte es nicht schwerfallen, die Beispiele und Skripten des Buches entsprechend anzupassen. Ich habe diverse Distributionen im Einsatz, wobei Debian den Löwenanteil stellt. Bei den meisten Distributionen sind passende Binärpakete direkt verfügbar und lassen sich von CD oder über das Netz installieren. Wenn Speicher oder Plattenplatz knapp sind, bietet das zweite Kapitel die nötigen Hilfen für die Auswahl einer möglichst „mageren“ Distribution.

Apropos Information: Ich habe das Literaturverzeichnis und die Linklisten im Anhang in verschiedene Rubriken unterteilt, so dass Sie leichter erkennen können, welche Inhalte bei den einzelnen Büchern oder Webseiten zu erwarten sind.

Erlauben Sie mir noch eine Randbemerkung: Das Buch ist selbstverständlich geschlechtsneutral verfasst. Da mir jedoch das sogenannte „Binnen-I“ wenig lesbar scheint („SystemadministratorInnen“) und die Angabe beider Geschlechter („Systemadministratorinnen und -administratoren“) dem Lesefluss nicht eben dient, habe ich mich auf die männliche Variante beschränkt. Abgesehen davon bekommt nur ein Pathologe den Systemadministrator *innen* zu sehen ...

Nicht zuletzt möchte ich mich bei allen bedanken, die mir direkt oder indirekt bei der Arbeit an diesem Buch geholfen haben, und auch bei jenen, die wegen der Arbeit am Manuskript vernachlässigt wurden. Mein besonderer Dank gilt Frau Metzger und Frau Weilhart vom Hanser-Verlag für die gute Zusammenarbeit, Herrn Sommer für das Finden aller Tippfehler, Hermann Zapf für die Palatino, Donald E. Knuth für  $\TeX$ , Leslie Lamport für  $\LaTeX$  und Linus Torvalds für Linux.

Etliche Themen konnte ich nur anschneiden oder Anwendungen skizzieren, weil sonst der Umfang dieses Buchs alle Grenzen gesprengt hätte. So sind auch Kompo-

nennten aus dem industriellen Bereich spärlich vertreten, denn es spielt meist keine Rolle für die Anwendung, ob beispielsweise ein Sensor „nackt“ eingesetzt wird oder ob er sich in einem Edelstahlgehäuse mit Schraubanschluss befindet. Es fehlt auch der gesamte Komplex „embedded Internet“, der genügend Stoff für ein eigenes Buch böte. In diesem Zusammenhang sind Sie als Leserin und Leser gefragt. Wenn Sie etwas vermissen, das Ihrer Meinung nach unbedingt in die folgenden Auflage gehört, oder wenn Sie einen Fehler finden, schreiben Sie mir doch eine E-Mail.

Weil der Stoff dieses Buchs an manchen Stellen etwas trocken ist, endet jedes Kapitel mit einer „Befeuchtungshilfe“ in Form eines Cocktail-Rezeptes. Für das Mixen brauchen Sie keinen Computer, sondern einige wenige Utensilien, die nicht viel kosten: einen Shaker, auch Schüttelbecher genannt, ein Rührglas für alle Drinks, die nicht geschüttelt, sondern gerührt werden („Gerührt oder geschüttelt, Mr. Bond?“), einen langstieligen Barlöffel und einen Meßbecher. Statt Rührglas kann man auch einen großen Becher vom Küchenmixer nehmen. Nützlich ist noch ein Eiscrusher zum Zerkleinern von Eiswürfeln.

Übrigens – gerade als Mixanfänger sollten Sie sich genau an die Mengenvorgaben halten. International üblich ist die Angabe der Flüssigkeitsmenge in Zentilitern (cl). 1 cl entspricht 10 ml, d.h. 0,01 l. Weitere Mengen sind: 1 Spritzer (dash) = 0,05 cl, 1 Schuss = 1 cl (1/2 Schnapsglas), 1 Barlöffel = 0,5 cl (etwa 1 Kaffeelöffel) und 1 Esslöffel = 1,5 cl.



*München, Sommer 2007*  
*Jürgen Plate (plate@netzmafia.de)*  
*<http://www.netzmafia.de/>*