

Vorwort zur vierten Auflage

Linux entwickelt sich unaufhaltsam weiter. Mehr Subsysteme, leichtere Portierbarkeit, besseres Realzeitverhalten und weitere Sicherheitsfunktionen. Die Liste lässt sich beliebig fortführen. In Version 4.x zeigt sich Linux professioneller, stabiler und sicherer, aber eben auch komplexer.

Professioneller, zugleich komplexer

Für den Entwickler wird es dadurch nicht einfacher. Gerade rechtzeitig hat daher Eben Upton 2012 den Raspberry Pi in den Markt gebracht. Das Besondere an diesem Minicomputer: Er ist mit dem Ziel entworfen worden, den Einstieg in die Computertechnik zu erleichtern und die Freude daran anzufachen. Und tatsächlich: Der Raspberry Pi vereinfacht vieles und eignet sich ausgezeichnet als Anschauungs- und Lehrobjekt. Problemlos kann unterschiedlichste Peripherie angeschlossen werden. Der Zugriff ist dabei ohne tiefgreifende Systemkenntnisse über beispielsweise PCI (Peripheral Interconnect) möglich.

Einfacher per Raspberry Pi

Das macht sich die vierte Neuauflage zunutze und zeigt relevante Interfaces wie beispielsweise diejenigen für Interrupts oder für den GPIO-Zugriff anhand von Codebeispielen, die direkt auf dem Raspberry Pi ablauffähig und daher mit sehr wenig Aufwand testbar sind. Die vierte Neuauflage nutzt den Raspberry Pi darüber hinaus, um in die Kernelcodeentwicklung für und auf dem Raspberry Pi einzuführen (cross- und native Entwicklung) und neue Konzepte vorzustellen, die wie beispielsweise Device Trees die Portierbarkeit des Kernels erweitern. Damit trägt sie auch dem Umstand Rechnung, dass die Mehrheit aktueller Linux-Systeme milliardenfach (!) auf Prozessoren der Firma ARM (Advanced Risc Machine) ablaufen.

Beispiele für PC und Raspberry Pi

Neben der Aufnahme der neuen Themenbereiche Kernel- und Treibercodeentwicklung für eingebettete Systeme und Cross-Entwicklung ist die vierte Neuauflage um neue Subsysteme ergänzt. Außerdem sind die bekannten Interfaces auf den aktuellen Stand gebracht worden. Wie gehabt werden wichtige Kernelinterfaces und Softwaremechanismen anhand vieler Codebeispiele anschaulich und zugleich kompakt vorgestellt. In Kombination mit dem vermittelten theoretischen Unterbau

Neue Subsysteme

hilft damit auch das vorliegende Buch, strukturiert das für die Codeentwicklung notwendige Wissen in kurzer Zeit (anwendbar) aufzubauen. Der aktualisierte und deutlich erweiterte Anhang mit der Beschreibung von über 750 Kernelfunktionen lässt das Buch gleichzeitig ein wertvolles Nachschlagewerk sein.

Open Source hilft.

Doch es gibt auch Grenzen bezüglich Umfang und Aktualität. Ein Blick in den Quellcode zeigt, dass es eine Reihe weiterer Subsysteme und Kernelfunktionen gibt, die hier nicht beschrieben sind. Und da sich der Linux-Kernel unaufhaltsam weiterentwickelt, haben die Entwickler die eine oder andere Kernelfunktion wohl in der Zeit zwischen Drucklegung und Ihrem Lesen geändert. Oft reicht es aus, den Quellcode nur um die Angabe einer Header-Datei zu ergänzen, manchmal tauschen Sie aber auch einen Parameter aus, selten verschwinden ganze Funktionen. Im Fall von Problemen helfen zumeist das Internet und vor allem auch der Linux-Quellcode weiter. Denn das ist eines der größten Assets, die Linux zu bieten hat: den offenen Quellcode.

Danke

Wir möchten die Gelegenheit nutzen und unseren Lesern für die wohlwollende Aufnahme der ersten drei Auflagen, für Korrekturhinweise und Anregungen danken. Möge auch die vierte Auflage Ihnen eine große Hilfe sein. Ein Dankeschön geht wieder an den dpunkt.verlag und hier insbesondere an Herrn Schönfeldt, der dieses Buch als Lektor bei allen Auflagen ausgesprochen freundlich und konstruktiv begleitet hat!

Kempen, im Oktober 2015

Jürgen Quade und Eva-Katharina Kunst