

Für Systeme im Embedded-Bereich gelten besondere Spielregeln: Technisch müssen sich die Applikationen mit begrenzten Hardwareressourcen, mit wenig Speicher und Energie arrangieren. Die Anwendungsbereiche sind hingegen breit gefächert – ob Waschmaschine, Herzschrittmacher, Multimedia-System, Operationssaal, Maschinensteuerung oder Auto: Überall arbeiten die meist unsichtbaren kleinen Systeme, ausgerüstet mit Betriebssystemen, die häufig Echtzeitanforderungen erfüllen müssen, zumal Embedded-Systeme zunehmend auch in sicherheitsrelevante Bereiche vordringen. Darüber hinaus steigert sich die Komplexität der Systeme, sie bekommen Grafik-Schnittstellen sowie Benutzer-Interfaces und sie vernetzen sich.

In sicherheitskritischen Bereichen gehört Echtzeit zu den Rahmenbedingungen, die es zu erfüllen gilt: Denn ein Motor muss unverzüglich stoppen, wenn es zu überhöhten Drehzahlen kommt, ein Airbag muss sich sofort öffnen. Auch im Operationssaal wäre es fatal, wenn die Embedded-Systeme dort geruhsam swappen würden, während der Patient nach Atem ringt. Echtzeitbetriebssysteme müssen also nicht nur zuverlässig arbeiten, sondern auch Deadlines einhalten.

Realtime: Ein weites Definitionsfeld

Umgangssprachlich wird der Begriff „Echtzeit“ dabei häufig mit „sofort“ gleichgesetzt, was streng genommen nicht stimmt: Es geht um die definierbare Zeitspanne, in denen Systeme reagieren müssen – vorhersagbar und immer gleich. Unter den Bedingungen von harter Echtzeit ist es ein Fehler mit meist gravierenden Auswirkungen, wenn das System seine Antwortzeit überzieht.

Stotternde Motoren sind gefährlich, ruckelnde Video-Vorführungen nicht unbedingt. Laufen Bild und Ton nicht synchron, ist das zwar ärgerlich, aber in den meisten Bereichen nicht existenziell. Viele Anwendungen stellen geringere deterministische Anforderungen, was sich in dem Begriff „weiche“ Echtzeit niederschlägt. Systeme, die weiche Echtzeitanforderungen erfüllen, müssen nicht unbedingt das vollständige Ergebnis liefern, und auch nicht zwangsläufig im definierten Zeitrahmen.

Wie die Marktübersicht zeigt, ist das Angebot an Echtzeitbetriebssystemen entsprechend der vielfältigen Anwen-

Echtzeit-Features beim Embedded Computing



Beizeiten

Barbara Lange

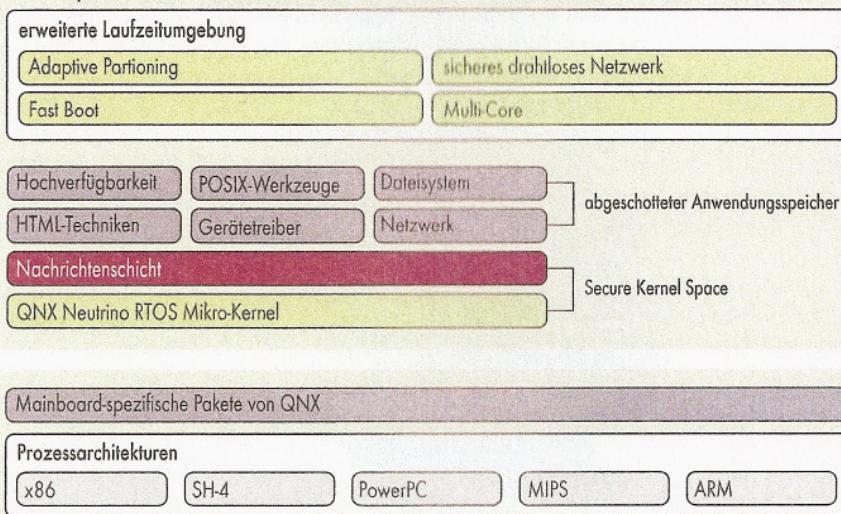
Vor allem im Embedded-Umfeld sind Echtzeitbetriebssysteme unerlässlich: So darf sich etwa der Airbag nicht lange mit System-Updates aufhalten, sondern muss sich bei Gefahr unverzüglich und garantiert öffnen. Ein Überblick.

dungsbereiche und unterstützten Plattformen groß. Die gewählte Hardware sowie die Konfiguration des Gesamtsystems bestimmen dabei Echtzeitverhalten und Antwortzeiten. Die meisten Anbieter bieten Versionen für diverse CPU-Architekturen und Branchen an, darunter Automotive, Medizintechnik, Maschinen- und Anlagenbau sowie In-

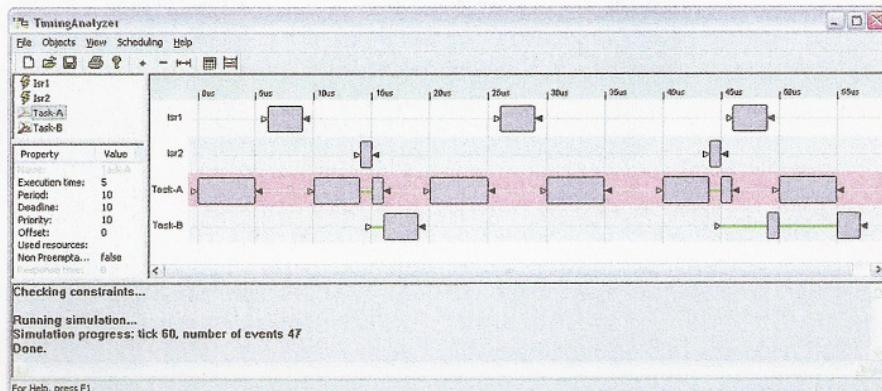
dustrie. Das hängt unter anderem mit den in der Regel hohen branchenspezifischen Standards für eine Zertifizierung der erstellten Produkte zusammen, die es zu berücksichtigen gilt.

Das Unternehmen QNX zum Beispiel hat sein Betriebssystem Neutrino kürzlich für Common Criteria ISO/IEC 15408 Evaluation Assurance Level 4+

Betriebssystem – QNX Neutrino RTOS



Das Echtzeitbetriebssystem Neutrino von QNX läuft auf 32-Bit-CPUs wie x86, Mips, PowerPC und ARM xScale (Abb. 1).



Der TimingAnalyzer von Vector Informatik erlaubt die Simulation von Scheduling-Tabellen und erleichtert damit die Bestimmung der zeitlichen „Worst Case“-Situation beim Testen (Abb. 2).

(EAL 4+) und gemäß IEC 61508 Safety Integrity Level 3 (SIL 3) zertifizieren lassen. SIL steht im Bereich der funktionalen Sicherheit für die verschiedenen Stufen der „Sicherheits-Integritätslevel“, die ein System erfüllen muss, um das Risiko einer Fehlfunktion so klein wie möglich zu halten. So können Kunden auf der Grundlage dieser zertifizierten Plattform Applikationen erstellen, die hohe Anforderungen an die Zuverlässigkeit erfüllen müssen. Für das zweite Quartal 2012 hat das Unternehmen eine Version für medizinische Geräte (IEC 62304) angekündigt.

Beim Neutrino RTOS Safe Kernel handelt es sich um einen Mikro-Kernel (siehe Kasten „Für Echtzeitarchitekten“), der viele Funktionen, die normalerweise ein Bestandteil des Kernels sind, als einzelne Prozesse im User-Space implementiert. Wenn es in den

ausgelagerten Komponenten einen Fehler gibt, beeinflusst dies nicht die Stabilität des Kernels.

Neutrino, das Echtzeitbetriebssystem von QNX, läuft auf 32-Bit-CPUs wie x86, Mips, PowerPC und ARM xScale. Im Jahr 2007 hat QNX Software Systems den Source-Code des Mikro-Kernels ihres RTOS (Real Time

Operating System) geöffnet. Dessen „Safe Kernel“ erfüllt die Anforderungen von IEC 615081, Safety Integrity Level 3. So können Kunden auf der Grundlage dieser zertifizierten Plattform Applikationen erstellen, die hohe Anforderungen an die Zuverlässigkeit erfüllen müssen (siehe Abb. 1).

Eine branchenspezifische Spezifikation für ein Embedded-Echtzeitbetriebssystem im Bereich Automobile definiert seit 1997 OSEK (Offene Systeme und deren Schnittstellen für die Elektronik im Kraftfahrzeug, siehe „Onlinequellen“, [a]). Dabei handelt es sich um eine statische Spezifikation, genau zugeschnitten auf die jeweiligen Hardwareressourcen in der Steuergeräteentwicklung im Bereich Automotive, wo man OSEK seit 1997 findet. Mittlerweile setzt das AUTOSAR-Projekt (Automotive Open System Architecture [b]) die Arbeit an OSEK fort. Zu den beteiligten Betriebssystemanbietern gehören Elektrobit, Embedded Office, Enea, ETAS und Vector Informatik. Letztere bieten mit ihrem TimingAnalyzer eine Erweiterung der beiden preemptiven Betriebssysteme osCAN und Microsar an (s. Abb. 2). Es gibt auch eine freie Version namens FreeOSEK, die an der FH Regensburg entwickelt wurde.

Echtzeit und Nicht-Echtzeit parallel

Viele Hersteller bieten Multi-Core- und Virtualisierungstechniken, um ebenfalls im Embedded-Bereich mehrere Realtime-Tasks oder Betriebssysteme gleichzeitig auf einem System laufen zu lassen. Denn zunehmend müssen Embedded-Applikationen zum einen Echtzeitbedingungen erfüllen, zum anderen aber auch auf andere Applikationen zugreifen können, um beispielsweise eine Bedienoberfläche zu steuern.

So nutzt Wind Rivers VxWorks, ein hart-echtzeitfähiges Betriebssystem, beispielsweise Multi-Core- und Hyper-

iX-TRACT

- Der Markt für Echtzeitbetriebssysteme deckt die vielfältigen Bedürfnisse unterschiedlicher Branchen, Sicherheitsstufen, Hardwarearchitekturen und Einsatzbereiche ab.
- Mit Virtualisierungstechniken können mehrere echtzeitfähige wie nicht zeitkritische Betriebssysteme auf einer Embedded-Architektur existieren.
- Seit der Kernel-Version 2.6.24. sind große Teile des „Realtime Preemption Patch“ Bestandteil des Linux-Standard-Kernels.

Anbieter von Echtzeitbetriebssystemen

| Betriebssystem | Hersteller | Webseite | Echtzeitverhalten ¹ | Lizenzmodell |
|---|---|--|---|---|
| μC/OS-II; μC/OS-III; μC/OS-OSEK; μC/OS-MMU | Micrium (Distributor: Embedded Office) | www.embedded-office.de, www.micrium.com | Hard-RT / 100 bis 120 µs | propriätär, Source-Code wird mitgeliefert |
| EB tresos | Elektrobit | www.electrobit.com | k. A. | propriätär |
| eCosPro | eCosCentric | www.ecoscentric.com | Hard-RT / 0,67 µs | modifizierte GNU GPL |
| ElinOS | Sysgo AG | www.sysgo.com | wie andere Linux-Systeme | GNU GPL |
| embOS | Segger Microcontroller | www.segger.com | Hard-RT / 1 µs (ARM, 200 MHz) | propriätär, Source-Code erhältlich |
| Euros | Euros – Embedded Systems | www.euros-embedded.com | Hard-RT / 10 µs | propriätär |
| Integrity | Greenhills Software | www.ghs.com | Hard-RT / je nach Architektur | propriätär |
| LynxOS, LynxOS-178, LynxOS-SE | LynxWorks | www.linuxworks.com | k. A. | propriätär |
| Microsar | Vector Informatik | www.vector.com | Hard-RT / k. A. | propriätär |
| Microware OS-9 | Radisys (Distributor: Microsys) | www.radisys.com/germany | k. A. | propriätär, Source-Code erhältlich |
| Monta Vista Linux | Montavista | www.mvista.com | k. A. | GNU GPL |
| Neutrino | QNX Software Systems GmbH & Co. KG | www.qnx.com/company/germany | Hard-RT / 0,5 bis 2,6 µs z.B. ARM Cortex: 0,5 µs) | propriätär, Teile des Source-Codes sind offen |
| Nucleus | Mentor Graphics Deutschland GmbH | www.mentor.com/germany | Hard-RT / ja | propriätär |
| OSE, OSEck | Enea | www.enea.de | k. A. | propriätär |
| PikeOS | Sysgo AG | www.sysgo.com | Hard-RT / < 1 µs | propriätär |
| Realtime Linux (OSADL recommends 2.6.33.7.2-r130) | OSADL | www.osdl.org/Realtime-Linux.projects-realtime-linux.0.html | Hard-RT / max. 100 000 Taktzyklen (z. B. 1-GHz-CPU: 100 µs) | GNU GPL v2 |
| Red Hat Enterprise MRG | Red Hat | www.redhat.com/mrg/ | Soft-RT / 8 µs | GNU GPL |
| RMOS3 | Siemens AG | www.siemens.de/rmos3 | Hard-RT / 10 µs | propriätär |
| RTA-OSECK3, RTA-OS3.0 | ETAS | www.etas.com/de | Hard-RT / Autosar-4.0-konform | propriätär |
| RTOS-32 | On Time Software | www.on-time.com | Hard-RT / < 5 µs | propriätär, Source-Code erhältlich |
| RTOS-UH | IEP | www.iep.de | Hard-RT / 1 µs (1 GHz Power-PC) | propriätär |
| SMX RTOS | Micro Digital (Distributor: Embedded Tools) | www.smxrlos.com, www.embedded-tools.de | Hard-RT / 78 Taktzyklen (z.B. 0,4 µs auf ARM 200 MHz) | propriätär |
| SUSE Linux Enterprise Real Time | SUSE Linux GmbH | www.suse.com/de-de/products/realtime/ | Hard-RT / 10 bis 100 µs | GNU GPL |
| Symbi; LinOS | Miray Software | www.miray.de, www.symbi.com | Hard-RT / ARM PXA-320: 0,2 ms | propriätär |
| Thread X | Express Logic | www.expresslogic.de, www.rtos.com | k. A. | propriätär, Source-Code wird mitgeliefert |
| VxWorks | Wind River Systems | www.windriver.com/de | Hard-RT / < 10 µs | propriätär |
| Wind River Linux | Wind River Systems | www.windriver.com/de | wie andere Linux-Systeme | k. A. |
| Windows Embedded Compact 7 | Microsoft | www.microsoft.com/windowsembedded | k. A. | propriätär |

Alle Daten sind Herstellerangaben, ¹Definition Hard- und Soft-RT siehe Text, dahinter minimale garantierter Antwortzeiten. Die tatsächlich erreichbaren Werte hängen stark von der Kombination Hard- und Software ab.

visor-Techniken. Es kann allein, als symmetrisches (SMP), asymmetrisches Multiprozessorsystem (AMP) oder als Gast-Echtzeitbetriebssystem im Wind River Hypervisor laufen, als Embedded Hypervisor vom Typ I direkt auf der Hardware. Mit ihm können Anwender sowohl sicherheitsrelevante als auch unkritische Applikationen in unterschiedlichen Partitionen auf einer Hardwareplattform nutzen. Die Verteilung der CPU-Ressourcen übernimmt der Scheduler des Hypervisors.

Nach Unternehmensangaben will man mit dieser Erweiterung des Hypervisors die Konsolidierung von zertifizierten und nicht zertifizierten Anwendungen auf derselben Single- oder Multi-Core-CPU erleichtern. Dazu gehören die Normen IEC 61508 für Industriesysteme, CENELEC EN 50128 SIL 4 für Transportsysteme, und ISO 26262 ASIL-D für den Bereich Automotive.

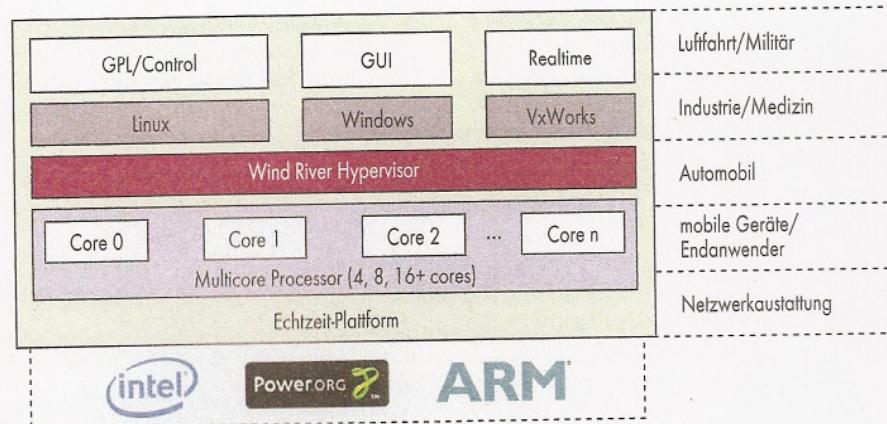
Auch Sysgo setzt auf Virtualisierung und bietet mit PikeOS ein Echtzeitbetriebssystem, das auf einem Mikro-Kernel basiert und gleichzeitig die Koexistenz von mehreren, unabhängigen

gen Betriebssystemen auf einer Hardwareplattform unterstützt, indem es voneinander unabhängige Partitionen für Programme, Laufzeitumgebungen oder Gastbetriebssysteme bereitstellt. So kann zum Beispiel die Linux-Distribution von Sysgo virtualisiert in einer Partition von PikeOS laufen. Dies hilft laut Hersteller bei einer Zertifizierung nach Industriestandards, wenn hohe An-

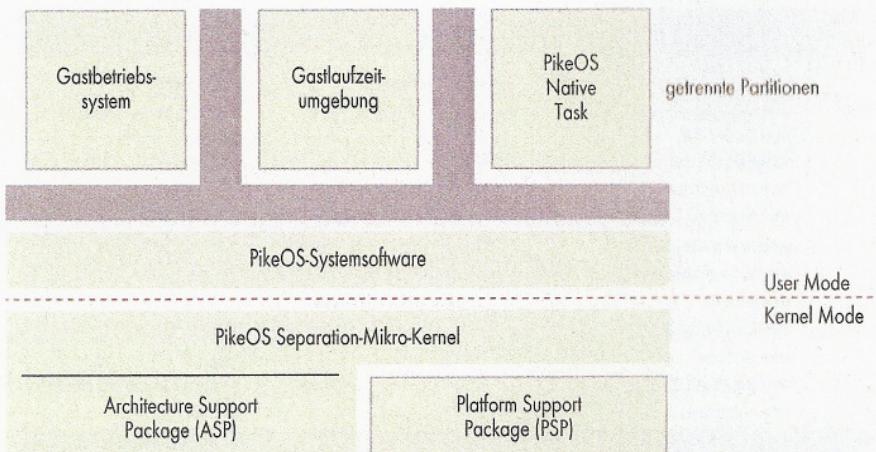
sprüche an die Betriebssicherheit oder Datensicherheit zu erfüllen sind.

Linux hält Termine ein

Open Source ist auch für die Echtzeitwelt ein Thema geworden. Zum einen bieten Hersteller wie Wind River, LynxWorks, Sysgo oder Symbi eigene



Mit dem Hypervisor von Wind River können Anwender sowohl sicherheitsrelevante als auch unkritische Applikationen in unterschiedlichen Partitionen auf einer Hardware-Plattform nutzen (Abb. 3).



Sysgos Echtzeitbetriebssystem PikeOS basiert auf einem Mikro-Kernel und unterstützt als Virtualisierungsplattform die Koexistenz mehrerer, unabhängiger Betriebssysteme auf einer Hardwareplattform (Abb. 4).

Embedded-Distributionen oder haben durch Veröffentlichungen von Quellcode den Open-Source-Gedanken in anderer Form in ihre Produktstrategie aufgenommen, zum Beispiel QNX. Auf der anderen Seite verfügen Distributionen wie SUSE, Red Hat oder Ubuntu über Echtzeiterweiterungen für Linux, das spätestens seit dem Kernel 2.6 echtzeitfähig geworden ist. Aber für den Einsatz des freien Betriebssystems in der Industrie gelten zusätzliche Anforderungen, die über den Server- und Desktop-Markt hinausgehen: Es muss über lange Zeit einen Support für zahlreiche Hardwareplattformen geben, Gerätetreiber sind zu entwickeln, und es stehen nur geringe Hardwareressourcen zur Verfügung.

Seit der Version 2.6.24 sind große Teile des „Realtime Preemption Patch“ Bestandteil des Linux-Kernels. Maßgeblich beteiligt an dieser Entwicklung ist die Genossenschaft „Open Source Automation Development Lab“, kurz OSADL [c], die sich um die Anpassung von Linux für den industriellen Einsatz im Maschinenbau und der Automatisierungsindustrie kümmert. Das

von OSADL gepflegte PREEMPT_RT-Echtzeit-Linux verhält sich präemptibel, das heißt, ein entsprechend privilegierter Prozess aus dem User-Modus kann das Betriebssystem unterbrechen, selbst wenn das System gerade im Kernel-Modus Aufgaben ausführt. Aktuell empfiehlt OSADL den „RT-Preempt realtime Linux Kernel 2.6.33.7.2-rt30“ als letzte stabile Version.

Zu den Mitgliedern der 2005 gegründeten Genossenschaft gehören Anwenderunternehmen und Hersteller wie ELTEC Electronik, Kontron, TRUMPF, Homag Holzbearbeitungssysteme, Heidelberger Druckmaschinen oder Roche Diagnostics. Auf Seiten der Betriebssystemanbieter ist Sysgo dabei. Seine Linux-Distribution ELinOS nutzt den Realtime-Preempt Kernel von OSADL und das freie Projekt Xenomai (siehe „Auswahl freier Echtzeitprojekte“).

Die generellen Vorteile von Open Source halten damit auch Einzug in den industriellen Bereich: Dazu gehören eine rasche Fehlerbeseitigung dank der Community und eine Unabhängigkeit von einzelnen Softwareherstellern. Durch den Besitz des Quellcodes

behielten die Anwenderunternehmen die Kontrolle über die Funktionsweise ihrer Systeme.

Dank der gemeinsamen Entwicklung und Offenlegung von Treibersoftware kann sich ein Unternehmen, so OSADL, stärker seiner Kernkompetenz widmen. Schon während der Hannover Messe 2008 betonte Open-Source-Evangelist Bruce Perens auf der Veranstaltung „Open Source meets industry“, dass das Alleinstellungsmerkmal eines Unternehmens sich nur in etwa fünf Prozent der Software ausdrücke. Dieser Anteil sei nicht für Open Source geeignet – die „übrigen“ 95 % aber sehr wohl [1].

Aber bei allen Vorteilen: Wer garantiert, dass die technischen Daten von Linux richtig sind? Damit beschäftigt sich das OSADL-Testzentrum „QA Farm Realtime“, das die Echtzeitperformance von 24 Prozessoren unter Linux misst. Feststellen will man, ob Systeme, die diese Hardware nutzen, in einer vordefinierten Zeit auf externe Ereignisse reagieren. Laut OSADL ist dieses wesentlich aufwendiger als die „normalen“ Tests des nicht echtzeitfähigen Standard-Linux-Kernels, daher können die Standard-Entwickler diese Aufgabe nicht in vollem Umfang leisten. Auf seinen Projektseiten stellt OSADL Testdaten zum Download bereit – darunter Echtzeitdaten von ARM-, MIPS-, PowerPC- und x86-basierten Systemen.

Echtzeiterweiterungen für Linux

Auch Anbieter von Enterprise-Linux-Distributionen haben ihre Systeme echtzeitfähig gemacht. So optimiert Red Hat mit seiner Erweiterung „MRG“ (Messaging, Realtime und Grid) das hauseigene Enterprise Linux (RHEL) im Hinblick auf Echtzeit-, Messaging- und Cluster-Eigenschaften. In dieser Version setzen

Auswahl freier Echtzeitprojekte

| Projekt | Webseite | Lizenz |
|-----------------------------|---------------------------|------------------|
| Atomthreads | atomthreads.com | BSD-Lizenz |
| ecos | ecos.sourceforge.org | modified GNU GPL |
| EmboX | code.google.com/p/embbox/ | BSD |
| FreeOSEK | openseek.sourceforge.net | GPLv3 |
| freeRTOS | www.freertos.org | modified GNU GPL |
| Realtime for Debian | debian.pengutronix.de | GNU GPL |
| RTLinuxFree | www.rtlinuxfree.com | GNU GPL |
| TinyOS | www.tinyos.net | BSD |
| Ubuntu RealTime-Erweiterung | wiki.ubuntu.com/RealTime | GNU GPL |
| Xenomai | www.xenomai.org | GPLv2 |

Onlinequellen

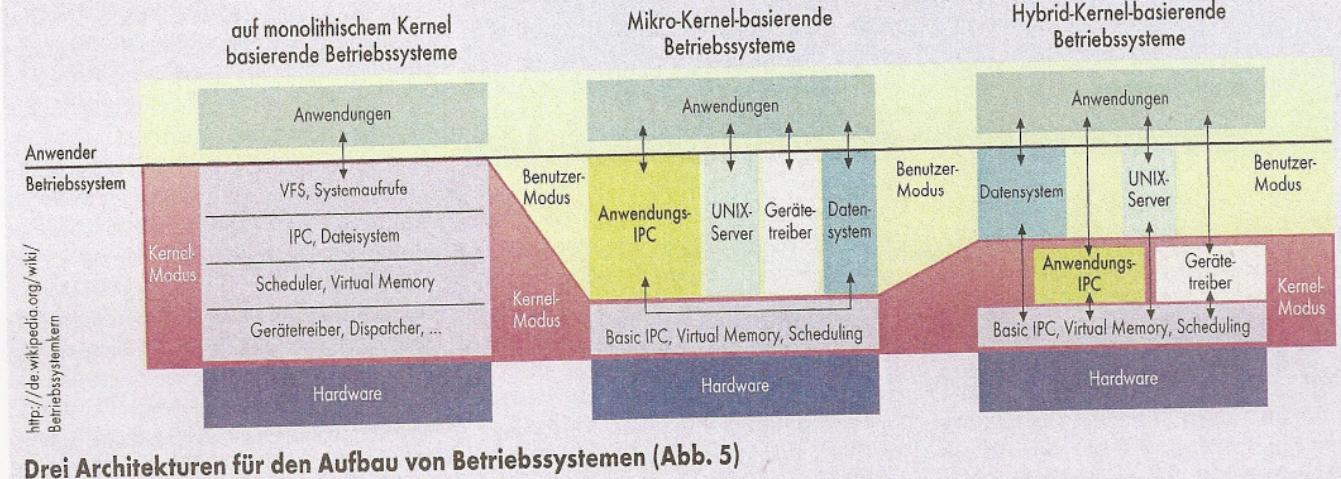
- [a] OSEK
www.osek-vdx.org
- [b] AUTOSAR
www.autosar.org
- [c] OSADL
www.osadl.org
- [d] OSADL-Testdaten
www.Osadl.Org/QA
- [e] NASA Jet Propulsion Lab
www.jpl.nasa.gov

Für Echtzeitarchitekten

Generell besteht eine Aufgabe von Betriebssystemen darin, die Rechenzeit der CPUs auf mehrere Tasks aufzuteilen. Während Time-Sharing-Systeme jedem Prozess eine Zeitscheibe zuordnen, funktionieren Echtzeitbetriebssysteme anders, denn sie müssen auf äußere Ereignisse reagieren können. Daher arbeitet die CPU hier die Tasks nach Prioritäten und Ereignissen ab. Der Scheduler verteilt die CPU-Zeit an die verschiedenen Tasks und hält sich an die festgelegten Regeln. Den Anstoß zur Verarbeitung liefert das zu ihm gehörende Ereignis – zum Beispiel, wenn die Bremse im Auto blockiert. Dann muss das System berechenbar und immer innerhalb einer definierten Zeitspanne reagieren.

Vom Gesichtspunkt der Architektur (siehe Abbildung 5) aus gesehen eignen sich Mikro-Kernel-basierte Betriebssysteme gut für Echtzeitanforderungen. Sie verfügen nur über grundlegende Funktionen wie den Scheduler, Ressourcenverwaltung und Grundfähigkeiten zur Kommunikation. Was darüber hinausgeht, lagert das System in Module aus.

Von den Angeboten in der Marktübersicht, die wie immer keinen Anspruch auf Vollständigkeit erhebt, setzen unter anderem PikeOS, Neutrino, Symbi oder ThreadX auf einem Mikro-Kernel auf. Linux ist mit dem „Realtime Preemption Patch“ echtzeitfähig geworden, indem die Entwickler den normalen Kernel um Echtzeitfunktionen ergänzen. Interrupts führt ein solcher RT-Kernel in Threads aus, die sich priorisieren lassen. Ein höher priorisierte Prozess kann einen anderen jederzeit unterbrechen („prä-empten“). Linux-Distributionen wie Red Hat, SUSE oder Ubuntu nutzen standardmäßig den aktuellen Realtime-Patch, bei anderen lässt sich der Kernel bei Bedarf leicht austauschen. Sysgo fährt mit seinem ElinOS zweigleisig und verwendet dort wahlweise besagten RT-Patch oder die Echtzeitweiterung Xenomai, die den normalen Linux-Kernel um einen priorisierten, zweiten für Echtzeitaufgaben ergänzt. Der konkrete Einsatz von Linux hängt aber nicht nur von der Technik und von Antwortzeiten ab, sondern auch von Lizenzfragen (Stichwort: GPL-Copyleft).



Drei Architekturen für den Aufbau von Betriebssystemen (Abb. 5)

die Amerikaner einen Linux-Kernel ein, der auf Linux 3.0-rt basiert. Systeme mit diesem Kernel können nach Unternehmensangaben innerhalb von acht Mikromenschen auf Ereignisse reagieren.

Einsatz findet das System bei der Deutschen Börse in Frankfurt (Aktienbörsen) und Eurex (Terminbörsen). Es soll dort dabei helfen, Risiken in Finanztransaktionen frühzeitig zu entdecken. Bislang haben die Beteiligten alle 10 Minuten eine verschlüsselte E-Mail mit Risikoinformationen erhalten. Die neue Risikomanagementplattform erlaubt nun einen Zugriff auf alle für das Risikomanagement relevanten Daten „nahezu in Echtzeit“, so Red Hat. Dafür hat die Deutsche Börse Systems eine Schnittstelle zwischen den Backend-Systemen von Eurex und den Risikomanagementsystemen der angeschlossenen Banken installiert. Die dafür geänderten Libraries wurden der Community zur Verfügung gestellt.

Mithilfe der Echtzeitweiterung für SUSEs Linux Enterprise Real Time (SLERT) kann man besser in die Sterne

gucken: Das Jet Propulsion Lab [e] nutzt SLERT 11 SP1 in seinem Hale-Teleskop im Palomar-Observatorium. Damit steuern die NASA-Forscher die über 3000 Einzelspiegel des Spiegelteleskops mit einer Frequenz von bis zu 2 kHz. Für Linux gibt es zusätzlich viele Echtzeit-Projekte – einige von ihnen sind in der Tabelle „Auswahl freier Echtzeitprojekte“ genannt.

Fazit

Der Markt für Echtzeitbetriebssysteme deckt ein vielfältiges Spektrum an Produkten für unterschiedliche Branchen, Zertifizierungen, Hardwareplattformen und Anwendungsbereiche ab. Zunehmend verfügen aktuelle Anwendungen sowohl über einen zeitkritischen als auch über einen nicht zeitkritischen Teil, zum Beispiel zur Anzeige eines User-Interface. Um mit dieser Aufgabe umgehen zu können, bieten inzwischen viele Hersteller Multi-Core- und Virtualisierungstechniken sowie Hypervisor,

um auch im Embedded-Bereich mehrere Betriebssysteme auf einem System laufen zu lassen. Seit Linux echtzeitfähig geworden ist, hält es Einzug in die Industrie – zum einen mit kommerziellen Distributionen von Seiten der Betriebssystemhersteller, zum anderen durch eine Integration großer Teile des „Realtime Preemption Patch“ in den Linux-Kernel.

BARBARA LANGE

ist IT-Journalistin und Inhaberin des Redaktionsbüros kurz und einfach in Lengede.

Literatur

- [1] Barbara Lange: Immer kleiner; Betriebssysteme für Industrie-PCs; iX extra 2/2009; www.heise.de/ix/downloads/05/6/9/9/7/6/1/ie0901.pdf

Alle Links: www.ix.de/ix1204093