

Virtualisierung Embedded: Viele Ziele – viele Wege

Virtualisierung ist im Embedded Computing angekommen. Doch noch viel weniger als in der kommerziellen IT gibt es den Königsweg zur „richtigen“ Virtualisierung. So vielfältig sich die Landschaft des Embedded Computing präsentiert, so unterschiedlich sind auch die Ansätze, die die Entwicklungsingenieure für ihre jeweiligen Anwendungen wählen.

Eines ist den meisten virtualisierten eingebetteten Anwendungsumgebungen gemein: Weil in der Regel von mindestens einer der Anwendungen Echtzeitfähigkeit verlangt wird, muss das gesamte System nach dem Start schnell betriebsbereit sein. Das schließt viele Lösungen aus, bei denen Windows, Linux oder ähnliche Betriebssysteme aus der kommerziellen IT die Kontrolle behalten; auch Embedded Linux oder Windows CE sind meist zu langsam. So läuft etwa der Hypervisor Xen auch auf den Embedded-Prozessoren von ARM. Weil das aber nur im Zusammenspiel mit Linux funktioniert und auch Embedded Linux für die meisten Realtime-Anwendungen als zu langsam gilt, fällt dieser Ansatz bei wirklichen Echtzeitanwendungen flach.

Hypervisor

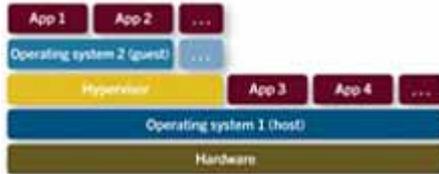
Zentrale Instanz in einer virtuellen Umgebung ist der Hypervisor. Diese Software verwaltet und steuert virtuelle Ressourcen; sie kontrolliert auch die Einhaltung der Separation zwischen Programmumgebungen. Fachleute unterscheiden zwischen Typ-1-Hypervisor und Typ-2-Hypervisor. Ersterer wird auch Bare-Metal-Hypervisor genannt, weil er direkt auf der Hardware (und oft mit direkter Virtualisierungsunterstützung des Prozessors) läuft; das darüberliegende Betriebssystem (oder die Betriebssysteme) unterliegt (unterliegen) seiner Kontrolle. Der Typ-2-Hypervisor läuft als Anwendung auf einem (Echtzeit)-Betriebssystem.

Hypervisor-Ansätze



Der Typ-1-Hypervisor liegt zwischen Prozessor-Hardware und Betriebssystem. (Grafik: QNX)

Beide Ansätze haben Vor- und Nachteile. Der Bare-Metal-Hypervisor gilt als sicherer und schneller in Echtzeitumgebungen, erfordert aber unter Umständen Modifikationen des darüberliegenden Betriebssystems.



Beispiel für einen Typ-2-Hypervisor. (Grafik: QNX)

Der Typ-2-Hypervisor gilt zwar als nicht so „schnell“, dafür als flexibler, und die dazugehörigen Anwendungen lassen sich leichter portieren. Manche Hypervisoren des Typs 1 nutzen direkt bestimmte Virtualisierungsunterstützungen des Prozessors, etwa die Embedded-Hypervisor-Technology auf manchen Freescale-Prozessoren, die TrustZone von ARM oder Intels VT.



PikeOS kommt vor allem in sicherheitskritischen Anwendungen zum Einsatz. (Grafik: Sysgo)

Solche Techniken unterstützen – je nach Architektur des Prozessors und Konzept des Herstellers – den Zugriff auf Prozessor-, Speicher- und I/O-Ressourcen. Ist diese Unterstützung praxisgerecht ausgeführt, so lässt sich dadurch der Aufwand für den Hypervisor reduzieren, Speicherzugriffe und I/O-Prozesse lassen sich beschleunigen und insgesamt die bei Echtzeitanwendungen wichtigen Latenzzeiten unter Umständen drastisch reduzieren.

Echtzeitbetriebssystem und Virtualisierungsplattform in einem ist beispielsweise das Betriebssystem PikeOS des Herstellers Sysgo, das vor allem in sicherheitskritischen Anwendungen wie Flugzeugen, Eisenbahnen oder Chemieanlagen zum Einsatz kommt. Das Betriebssystem bietet neben der inhärenten Virtualisierung gleichzeitig eine Partitionierung der darüberliegenden Anwendungen.

Tools für Virtualisierer

- PikeOS: Virtualisierungsplattform und Echtzeit in einem Betriebssystem www.sysgo.com/products/pikeos-rtos-and-virtualization/rtos-technology/
- Virtuelles File- & Printersharing: CIFS NQ (Network Quick) Client-Server File Sharing-Software von Visuality Systems www.real-time-systems.com/samba/index.php
- Realtime-Hypervisor: Fokus auf Intel-Hardware unter Windows www.real-time-systems.com/real-time_hypervisor/index.php
- Tools zur Virtualisierung von Echtzeitsystemen und Windows/Windows CE auf gemeinsamer Hardwareplattform www.acontis.com/produkte/virtualisierung-von-real-time-os-und-windows/index.php