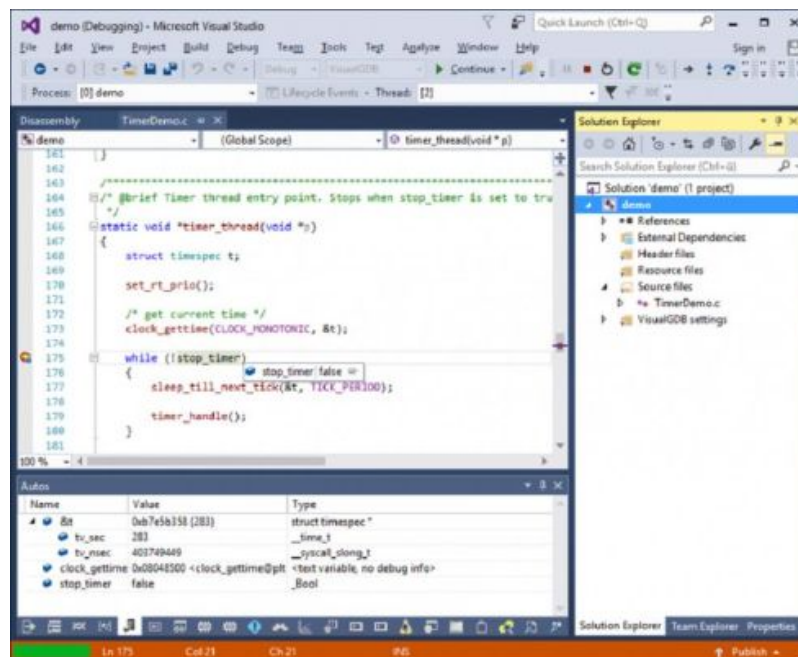


Neu: RT-Linux 64 Bit als Lösung für Echtzeitanwendungen unter Windows

13.11.2020, 15:54 | IT, New Media & Software

Pressemitteilung von: *acontis technologies GmbH*



Immer mehr Anwendungen, insbesondere in den Bereichen Industrieautomation, Medizin und Datenerfassung und -messung, erfordern deterministisches Echtzeitverhalten.

Windows unterstützt eine deterministische Echtzeit-Operation nicht nativ.

Es gibt jedoch viele so genannte "Windows-Echtzeiterweiterungen", die Entwicklern eine Lösung bieten können, um die Windows-Funktionalität voll auszunutzen und gleichzeitig die Echtzeitanforderungen ihrer Anwendung zu erfüllen.

Nachstehend finden Sie einen Leitfaden für Kunden, die Echtzeitanwendungen unter Windows betreiben müssen:

- **Windows und Echtzeit: Hintergrund**

Hintergrundinformationen über Windows und Echtzeit-Betriebssysteme

- **Anforderungen um Echtzeit zu erzielen**

Was ist zu beachten, wenn Echtzeitanwendungen unter Windows laufen sollen

- **Windows Echtzeit-Erweiterungen im Vergleich**

Welche Lösungen sind verfügbar, was sind die Unterschiede

- **Wie RT-Linux alle Anforderungen erfüllt**

Warum Echtzeit-Linux heute die beste Wahl ist

Windows und Echtzeit: Hintergrund

Der Unterschied zwischen einem Standard-Betriebssystem (OS) und einem Echtzeit-Betriebssystem (RTOS), welche typischerweise in vielen eingebetteten Systemen zu finden sind, liegt in der Reaktionszeit auf externe Ereignisse.

Standard-Betriebssysteme bieten typischerweise eine nicht-deterministische, in einigen Fällen weiche Echtzeitreaktion, bei der es keine Garantien dafür gibt, wann die jeweilige Aufgabe abgeschlossen ist; dagegen steht die schnelle Reaktion auf Benutzer- bzw. Bediener-Anforderungen im Vordergrund. Ein RTOS unterscheidet sich dadurch, dass es eine harte Echtzeitreaktion liefert, die eine schnelle und deterministische Reaktion auf externe Ereignisse ermöglicht. Der

Unterschied zwischen beiden Arten von Betriebssystemen kann anhand eines einfachen Beispiels verdeutlicht werden - vergleichen Sie die Bearbeitung eines Dokuments durch einen Benutzer auf einem Windows-PC mit dem Betrieb einer Präzisionsmotorsteuerung.

Beim Umschalten zwischen Tasks muss das RTOS diejenige Task auswählen, die als nächstes auf Grund ihrer Ausführungspriorität (der Wichtigkeit) ausgeführt werden soll. Es stehen hierbei verschiedene Scheduling-Algorithmen zur Verfügung, darunter Round Robin, kooperatives und hybrides Scheduling. Um jedoch ein reaktionsfähiges System bereitzustellen, verwenden die meisten RTOS einen präemptiven Scheduling-Algorithmus.

In einem präemptiven System wird jeder Aufgabe ein individueller Prioritätswert zugeordnet. Je schneller die erforderliche Reaktion erfolgt, desto höher ist die Prioritätsstufe. Wenn in diesem Modus gearbeitet wird, ist die Aufgabe, die zur Ausführung kommt, die Aufgabe mit der höchsten Priorität. Dies führt zu einem System mit hoher Reaktionsfähigkeit.

Auf Windows-Systemen könnte man vermuten, dass wenn Software in einen Windows-Treiber eingebettet wird, diese Software dann deterministische bzw. Echtzeit-Anforderungen erfüllen würde.

Windows führt jedoch Funktionsaufrufe an das Betriebssystem so aus, dass eine maximale durchschnittliche Rechenleistung und schnelle interaktive Benutzerreaktionen erzielt werden. Es gibt jedoch keine Garantie für die Antwortzeit. Dies gilt sowohl für normale Windows-Anwendungen, selbst wenn Threads mit der Echtzeit-Priorität von Windows

laufen, als auch für Windows-Treiber, da die zugrunde liegenden Mechanismen des Windows-Betriebssystems für beide die gleichen sind.

Anforderungen, um Echtzeit zu erzielen

Das Laufzeitsystem

Um eine deterministische Echtzeitleistung zu gewährleisten, muss das zugrundeliegende Laufzeitsystem, das für die Ausführung der Echtzeitoperationen verantwortlich ist, völlig unabhängig von Windows sein. Es muss eine Vielzahl von Funktionalitäten bereitstellen, um typische Echtzeitanforderungen zu erfüllen, wie z.B.:

- Speicherverwaltung (Heap und Stack)
- Multithreading: deterministischer Echtzeit-Scheduler
- Multi-Processing: Ausführen mehrerer unabhängiger Anwendungsinstanzen
- Abschottung des Laufzeitsystems von den Applikationen
- Thread-/Prozess-Synchronisierung und Kommunikationsmechanismen
- Multi Core-Unterstützung (Symmetrisches Multiprocessing)

Zusammengefasst sind diese Anforderungen typische Merkmale, die ein Echtzeitbetriebssystem bietet und somit, dass für das Echtzeitverhalten in Windows ein Echtzeitbetriebssystem erforderlich ist. Folglich kann ein Laufzeitsystem, das zumindest einige der oben genannten Merkmale bietet, auch als Echtzeitbetriebssystem (RTOS) bezeichnet werden, und dieses RTOS muss völlig unabhängig von Windows sein, wo auch immer Echtzeitanforderungen bestehen.

Isolation

Isolation ist eine Schlüsselanforderung, um einen zuverlässigen und sicheren Betrieb von Echtzeitanwendungen neben Windows zu erreichen. Neben der Isolation des Laufzeitsystems von den ausgeführten Echtzeitanwendungen muss auch das Laufzeitsystem von Windows isoliert und geschützt werden. Um eine maximale Isolation zu erreichen, sollten MMU- und Virtualisierungstechnologie eingesetzt werden.

Garantie der Echtzeit

Auch wenn es sich bei dem in einer Windows-Echtzeiterweiterung verwendeten Laufzeitsystem um ein Echtzeitbetriebssystem handelt, gibt es eine Reihe von Aspekten, die zu berücksichtigen sind, um diese Funktionen bei der Ausführung neben Windows auf einer modernen PC-Architektur zu erhalten. Der Zugriff auf gemeinsam genutzte Hardwareressourcen, wie Cache und RAM, ist einer der kritischsten Aspekte. Auch System-Management-Interrupts (SMIs) können zu inakzeptablen Verzögerungen führen. Daher muss sich das Laufzeitsystem mit dieser Problematik auseinandersetzen. Der Einsatz der Cache Allocation Technologie hilft, Probleme mit gemeinsam genutztem Cache zu überwinden, während die Virtualisierungstechnologie zur Unterdrückung von SMIs eingesetzt werden kann. Darüber

hinaus sollte eine Windows-Echtzeiterweiterung auch Werkzeuge zur Analyse des Echtzeitverhaltens der verwendeten Hardware zur Verfügung stellen.

C/C++-Laufzeitbibliothek

Die C/C++-Laufzeitbibliothek ist eine Library, die vom Compiler verwendet wird, um Funktionen des Laufzeitsystems aufzurufen. Das C/C++-Laufzeitsystem implementiert u.a. eingebaute Funktionen und andere grundlegende Verhaltensweisen der Programmiersprache C/C++. Neben der grundlegenden Unterstützung der Programmiersprache C/C++ gehen die heutigen Anforderungen teilweise weiter darüber hinaus und fordern den Einsatz von Bibliotheken höherer Ebene wie STL oder Boost.

Entwicklungsumgebung

Der Hauptzweck einer Entwicklungsumgebung besteht darin, verschiedene Komponenten von Softwareanwendungen für die Entwicklung und Weiterentwicklung der Anwendung bereitzustellen. Eine integrierte Entwicklungsumgebung (Integrated Development Environment, IDE) ist ein Werkzeug, das alle erforderlichen Anwendungen wie Compiler, Linker und Debugger unter einer einzigen grafischen Umgebung integriert. Was Entwickler an einer solchen IDE schätzen, ist ihre ansprechende Benutzeroberfläche und ihr Leistungsumfang. Unter Windows ist die Microsoft Visual Studio-IDE zum De-facto-Standard für die Anwendungsentwicklung geworden, und daher muss eine Windows-Echtzeiterweiterung in der Lage sein, diese IDE auch für Echtzeitanwendungen zu nutzen.

Popularität, Flexibilität und Portabilität

Um den größtmöglichen Nutzen aus dem RTOS zu ziehen, sollte die Windows-Echtzeiterweiterung ein weit verbreitetes RTOS verwenden, eines mit vielen Funktionen, Produkten von Drittanbietern und Dokumentation, das flexibel einsetzbar ist. Wenn die gleiche Echtzeitanwendung auch stand-alone auf einem PC ohne Windows oder sogar auf einem ARM-basierten eingebetteten System ausgeführt werden kann, wird dies in Zukunft ein Höchstmaß an Flexibilität gewährleisten.

Windows-Echtzeit-Erweiterungen

Echtzeiterweiterungen für Windows können auf unterschiedliche Art und Weise implementiert werden, es gibt wichtige Vorteile und Kompromisse, die zu berücksichtigen sind. Im Gegensatz dazu, was Benutzer glauben könnten, machen Echtzeiterweiterungen nicht Windows selbst zu einem Echtzeitbetriebssystem (RTOS). Bestehende Windows-Anwendungen werden sich nicht automatisch wie Echtzeitanwendungen verhalten, wenn eine Windows-Echtzeiterweiterung genutzt wird. Stattdessen wird für das Echtzeitverhalten ein Echtzeit-Laufzeitsystem benötigt - und eine solche Laufzeit-Umgebung ist faktisch ein RTOS.

Typischerweise implementieren Windows-Echtzeiterweiterungen ein proprietäres RTOS parallel ausgeführt zu Windows oder manchmal sogar eingebettet in Windows. Diese RTOS-Lösungen werden nie oder selten in anderen Arten von Anwendungen eingesetzt.

Bei diesen Lösungen muss jede Echtzeitanwendung auf diesem proprietären RTOS laufen. Dies kann für Entwickler eine Herausforderung darstellen, da sie die proprietäre Umgebung für die Entwicklung ihrer Echtzeitanwendungen erlernen und nutzen müssen. Außerdem müssen sie sich auf das Vorhandensein von Funktionen verlassen, die vom Hersteller dieses RTOS implementiert werden sollten, diese Funktionen fehlen manchmal oder es braucht Zeit, bis diese vollständig verfügbar sind.

LxWin - die Lösung von acontis setzt auf den Industrie-Standard

Die acontis Echtzeiterweiterung LxWin arbeitet anders als eine traditionelle Windows-Echtzeiterweiterung und bietet eine interessante Alternative, die Anwender und Implementierer von Echtzeitanwendungen in Betracht ziehen sollten. Erstens verwendet LxWin einen Ansatz der die Echtzeitanwendungen und den Nicht-Echtzeit-Windows-Teil stärker voneinander isoliert. Anstatt ein proprietäres RTOS zu verwenden, bettet LxWin einen harten Echtzeit-Linux-Kernel auf einem Echtzeit-Hypervisor ein. Dieser Hypervisor ist der Schlüssel und gewährleistet eine viel bessere Isolierung von Windows im Vergleich zu traditionellen Echtzeit-Erweiterungen.

RT-Linux ist das de-facto Standard Industrie RTOS

Darüber hinaus und am wichtigsten ist, dass LxWin für den Echtzeit-Part kein proprietäres RTOS sondern Open-Source-Linux verwendet. Dies ist ein Schlüsselmerkmal, da sich RT-Linux in den letzten Jahren zum am weitesten verbreiteten Betriebssystem für industrielle Echtzeitanwendungen entwickelt hat. Das bedeutet, dass Entwickler alle gängigen Tools und Pakete, die bereits von der Open-Source-Gemeinschaft für Linux zur Verfügung stehen, verwenden können und kein neues proprietäres RTOS erlernen müssen.

Der Einsatz von Linux als RTOS innerhalb einer Windows-Echtzeiterweiterung ist für industrielle Anwendungen, die Echtzeitverhalten benötigen, sehr sinnvoll. Ursprünglich wurde die Echtzeit-Linux-Entwicklungsaktivität durch das Open Source Automation Development Lab (OSADL) vorangetrieben, und große Unternehmen wie AMD, Intel, BMW, Roche, Phoenix Contact und KUKA waren alle Mitglieder dieser Organisation. Im Jahr 2016, als die Entwicklung von Echtzeit-Linux immer populärer wurde, übernahm die Linux Foundation die Leitung des Projekts, um den Mainline-Linux-Kernel selbst zu einem Echtzeit-Betriebssystem zu machen. Es wird erwartet, dass dieses Ziel irgendwann später in diesem Jahr oder im Jahr 2021 endgültig umgesetzt ist.

Acontis befindet sich in einer einzigartigen Position, um den Trend für industrielle Anwendungen und die Verteilung der verwendeten Betriebssysteme zu erkennen. Die EtherCAT-Master-Software, EC-Master, die heute mehr als 25 verschiedene Betriebssysteme unterstützt, muss typischerweise in einem Echtzeitbetriebssystem laufen, um die beste und schnellste Leistung zu erzielen. Heute ist Linux als die mit großem Abstand beliebteste Wahl geworden, mehr als 35 % aller EC-Master-Kunden setzen Linux ein und immer mehr neue Kunden entscheiden sich für Linux, so dass wir erwarten, dass dieser Prozentsatz noch deutlich steigen wird.

Wir sehen diesen Trend auch auf dem breiteren industriellen Markt, der über reine EtherCAT-Anwendungen hinausgeht. Führende globale Unternehmen, die Maschinen-, Bewegungs- und Robotersteuerungen sowie SPS, Halbleiterfertigungsmaschinen und Test- und Messtechniklösungen anbieten, haben alle damit begonnen, RT-Linux anstelle anderer, meist auch veralteter Echtzeitbetriebssysteme zu implementieren.

Schließlich wird RT-Linux im kommenden neuen Echtzeit-Ethernet-Standard, der als Time Sensitive Networking (TSN) bekannt ist, eine Schlüsselrolle als wichtigstes Betriebssystem spielen.

Wie RT-Linux alle Anforderungen erfüllt

Unter Berücksichtigung der oben genannten Anforderungen an eine Windows-Echtzeiterweiterung liegt es auf der Hand, dass RT-Linux heute die beste Wahl für den Einsatz als Windows Echtzeiterweiterung ist.

Das Laufzeitsystem

RT-Linux bietet alle Funktionalitäten, um die heutigen Anforderungen an ein RTOS zu erfüllen. Es ist eines der leistungsfähigsten, wenn nicht sogar das leistungsfähigste Betriebssystem auf dem Markt. Abhängig von den tatsächlichen Bedürfnissen kann der Footprint auch deutlich verkleinert werden, so dass es nur typische Komponenten enthält, die für Echtzeitanwendungen benötigt werden. Linux wird nicht nur in Servern eingesetzt; es wird auch in kleinen eingebetteten Geräten wie Smart-Uhren oder Fernsehern verwendet, bei denen der Speicherbedarf immer ein wichtiges Kriterium ist. RT-Linux ist somit nicht nur klein sondern bei Bedarf auch sehr leistungsstark.

Für anspruchsvolle Anwendungen unterstützt LxWin auch Symmetrisches Multiprocessing, wobei mehrere CPU-Kerne für die Echtzeitanwendung verwendet werden können.

Isolation

RT-Linux selbst verfügt über eine sichere und zuverlässige Isolierung zwischen den Echtzeitanwendungen und dem Linux-Kernel. Obwohl für Echtzeitanwendungen unkritisch, werden auch Exploits wie Meltdown etc. berücksichtigt. Darüber hinaus sind Windows und RT-Linux durch MMU-Mechanismen sicher voneinander isoliert, eine noch stärkere Isolierung kann durch die Aktivierung der Intel-Virtualisierungstechnologie erreicht werden.

Echtzeit-Analyse

LxWin bietet ein einfaches Werkzeug zur Analyse der Echtzeitfähigkeiten. Ein einfaches 3-Ampel-System zeigt auf einfache Art und Weise an, ob das System für Echtzeitanwendungen ausreichend ist und wo ggf. Schwachstellen sind. Acontis hat kürzlich damit begonnen, eine Auswahl an vorvalidierter Hardware zu erstellen, die für den Einsatz in

Echtzeitanwendungen geeignet ist.

Um die Echtzeitfähigkeiten zu verbessern, können sowohl CAT als auch Intel VT eingesetzt werden.

C/C++-Laufzeitbibliothek, Treiber, Software von Drittanbietern

Neben der Tatsache, dass Linux selbst ein Echtzeit-Betriebssystem ist, bringt der Einsatz weitere nützliche Vorteile mit sich. Funktionen wie Sockets, eine vollständige POSIX-API, User-Space/Kernel-Space-Isolation, Multitasking und Multiprocessing sind verfügbar. Für fast jede heute auf dem Markt erhältliche Hardware ist ein Linux-Treiber verfügbar. Das bedeutet, dass Entwickler nicht darauf warten müssen, dass ein bestimmter Treiber vom Hersteller einer proprietären Echtzeit-Erweiterungsplattform implementiert und freigegeben wird. Des Weiteren gibt es bei Linux eine Fülle von Software-Bibliotheken, Protokoll-Stacks und Middleware (sowohl Open-Source- als auch kommerzielle Closed-Source-Produkte): C++ STL, Boost, Corba, OPC-UA, DDS, AVB/TSN, EtherCAT, PROFINET, Software-SPS, Motion-Control-Bibliotheken, CNC-Lösungen und viele, viele mehr.

Für Kunden, die die EtherCAT-Technologie einsetzen, bietet acontis die EC-Win Windows EtherCAT-Plattform an, die auf LxWin basiert. Schließlich können Kunden, die ein EtherCAT-Netzwerk in Echtzeit simulieren wollen, die acontis EtherCAT-Simulator-Software EC-Simulator unter Windows mit LxWin betreiben.

Entwicklungsumgebung

Von zentraler Wichtigkeit für die Entwicklung und das Debugging von Echtzeitanwendungen ist die Unterstützung von Microsoft Visual Studio. Projektassistenten helfen Kunden, Echtzeitanwendungen auf einfache Weise zu erstellen – genau so einfach wie das Erstellen von Windows-Anwendungen.

Auch das Debugging erfolgt auf dieselbe Art und Weise wie das Debugging regulärer Windows-Anwendungen.

Alternativ können auch Kunden, die mit Eclipse vertraut sind, diese leistungsfähige IDE nutzen, sowohl unter Linux als auch unter Windows.

Popularität, Flexibilität und Portabilität

Linux ist heute das beliebteste Betriebssystem für eingebettete Systeme. Es ist auf Intel x86 (sowohl 32-Bit als auch 64-Bit) verfügbar. Anwendungen, die für die PC-Plattform geschrieben wurden, können leicht auf die im Embedded und Industriebereich sehr populäre ARM-Architektur portiert werden, da das Betriebssystem die Wiederverwendbarkeit der vorhandenen Quellen gewährleistet.

Kunden, die Microsoft Windows Embedded Compact (ehemals Windows CE) verwenden, sind derzeit gezwungen, auf eine Alternativlösung umzusteigen, da Microsoft keine Updates und keinen Support für diese Systeme mehr anbietet. Der beliebteste Migrationspfad ist RT-Linux und man kann die Verwendung von LxWin in Betracht ziehen, wenn Windows-Funktionen erhalten bleiben sollen.

Herausforderungen bei der Erstellung eines Linux-Kernels für Echtzeitanwendungen

Es gibt dennoch einige Herausforderungen bei der Verwendung von Linux für Echtzeitanwendungen. Eine der schwierigsten Aufgaben ist die Konfiguration des Linux-Kernels, damit das Echtzeitverhalten gewährleistet ist. Bis der Mainline-Linux-Kernel vollständig als Echtzeit-Kernel realisiert ist, müssen Entwickler heute den Kernel patchen und speziell konfigurieren, um Echtzeit zu erreichen. Dies wird oft als RT-PREEMPT-Patch bezeichnet, aber es können viele Patches sein und nicht alle funktionieren perfekt.

Mit LxWin brauchen sich Entwickler jedoch überhaupt nicht mit der Konfiguration und dem Bau des Linux-Kernels zu beschäftigen. Der Linux-Kernel, der mit LxWin ausgeliefert wird, ist bereits entsprechend gepatcht, konfiguriert und als harter Echtzeit-Kernel mit deterministischem Verhalten getestet. Als Ergebnis können extrem kurze Zykluszeiten von bis zu 50 Mikrosekunden erreicht werden.

Zusammenfassung der wichtigsten Funktionen von LxWin

- Verfügbar für Windows 7 und folgende Windows-Versionen (32 Bit und 64 Bit)
- Schnelle Reaktionszeit innerhalb von Mikrosekunden und volle Unabhängigkeit von Windows, auch wenn Windows abstürzt (BSOD), arbeitet der Echtzeiteil weiter!
- Optimierte Nutzung von Mehrkernsystemen (skalierbar von einem bis zu mehreren CPU-Kernen für den Echtzeiteil)
- Intuitive und beliebte Echtzeit-API (Prozesse, Threads, Ereignisse, Sockets, POSIX-Unterstützung, ...)
- Unterstützung der Microsoft-Entwicklungstools (Visual Studio)
- Komfortable Debugging-Umgebung für den Echtzeiteil unter Verwendung der Microsoft-Werkzeuge,

Echtzeitanwendungen werden wie eine normale Windows-Anwendung entwickelt und debuggt



Portrait

acontis technologies ist ein führender Branchenanbieter von hoch-performanten Windows-Echtzeiterweiterungen, Echtzeit-Virtualisierungssoftware und Echtzeit-Ethernet-Kommunikationssoftware für den Einsatz in unterschiedlichsten Märkten. Die Windows-Echtzeiterweiterungen und die Echtzeit-Virtualisierungssoftware erfüllen harte Echtzeitanforderungen für den Betrieb fortschrittlichster industriellen Steuerungs-, Robotik-, Prozesssteuerungs-, Halbleiter-, Hardware-in-the-Loop-Simulations und Hochgeschwindigkeitsdatenerfassungssoftware sowie andere Anwendungen mit geringer Latenzzeit. Acontis technologies bietet diese Echtzeitleösungen seit über 25 Jahren mit Vertrieb und Support an Standorten in ganz Europa, Nordamerika und Asien an.

News-ID: 1107718 • Views: 1909 (Stand: 09.10.2026)

Link zur Pressemitteilung:

<https://www.openpr.de/news/1107718/Neu-RT-Linux-64-Bit-als-Loesung-fuer-Echtzeitapplikationen-unter-Windows.html>