

Neue Echtzeit-Konzepte für Windows

Jürgen Rall

Windows XP hat sich im professionellen Workstation-Einsatz weitgehend als Standard-Betriebssystem durchgesetzt. Mit der Entwicklung von XP wurde zudem die Anpassung an die erweiterten Eigenschaften portabler Plattformen und neuer Kartentechnologien in punkto Energieverwaltung und Plug&Play-Management realisiert. Auf Grund seiner Sicherheitsmechanismen, den Kommunikationseigenschaften, sowie seiner umfangreichen Visualisierungsfähigkeiten bietet es dem Benutzer einen komfortablen und sicheren Umgang mit dem Betriebssystem. Um diese Eigenschaften des Betriebssystems auch im industriellen Einsatz nutzen zu können, müssen jedoch Prozesse für Kommunikation und Datenaufbereitung deterministisch abgearbeitet werden können – ein Vorgang, der mit den passenden Tools auch gut zu bewältigen ist.

Die Multitasking-Struktur von Windows XP widerspricht dem deterministischen Zeitverhalten einer Task-Verarbeitung. Unter XP kommt eine Kombination unterschiedlicher Task-Verteilungsmechanismen zum Einsatz. Tasks der niedrigsten Prioritätsstufe (PASSIVE_IRQL) werden typischerweise mit einer Unterbrechung von 10 ms nach dem Zeitscheiben-Verfahren (Round Robin) abgearbeitet, können jedoch durch Tasks höherer Priorität asymmetrisch unterbrochen werden. Die Tatsache, dass Tasks höchster Priorität (z. B. ISR -> DEVICE_IRQL) nicht-preemptiv abgearbeitet werden, verhindert zudem ein deterministisches Zeitverhalten bei der Task-Abarbeitung. Das XP-Multitasking ist für eine optimale Gesamt-Verarbeitungsleistung ausgelegt, jedoch nicht für Prozessbearbeitung in Echtzeit.

Ein Betriebssystem muss in seiner Struktur für eine deterministische Datenverarbeitung bestimmte Kriterien erfüllen. Wichtige systembedingte Voraussetzungen für die Echtzeit-Verarbeitung hierbei sind:

- ▷ Preemptives Multitasking
- ▷ Unterbrechbare Interrupt-Verarbeitung
- ▷ Keine Beeinflussung durch andere Prozesse

- ▷ Zeitscheibenbasierter Taskwechsel
- ▷ Mögliche Zuordnung der höchsten Prioritätsebene für Echtzeit-Task

Durch die Kombination des Betriebssystems XP mit einem echtzeitfähigen Sub-Kernel-System (Echtzeit-Erweiterung) lassen sich die Vorteile beider Systeme vereinen. So lassen sich die Visualisierungs- und Kommunikations-Eigenschaften von XP mit dem deterministischen Zeitverhalten des echtzeitfähigen Subsystems z. B. für Steuerungsaufgaben und Protokollverarbeitung ideal nutzen.

Heute werden unterschiedliche Echtzeit-Erweiterungen für Windows XP angeboten, die jedoch oft auf proprietäre Lösungen zugeschnitten sind. Bei reinen Software-Lösungen muss in derartigen Systemen auf Grund des restriktiven Betriebs-Managements von XP allerdings meist mit erheblichen Abweichungen (Jitter) gerechnet werden.

Während der Einsatz früherer Echtzeit-Betriebssysteme auf geschlossene Systeme begrenzt war, kommen solche Betriebssysteme heute auch in offenen Plattformen zum Einsatz: Sie übernehmen zusätzlich Aufgaben zur Visualisierung und Kommunikation.

Echtzeit wird immer dann benötigt, wenn zeitkritische dynamische Systeme zuverlässig verwaltet werden müssen. Die Verletzung des deterministischen Zeitverhaltens führt bei sensiblen Systemen meist zu einem kritischen bis katastrophalen Zustand (z. B. bei der Linear-Steuerung oder bei der Füllmengen-Überwachung), während nicht-sensitive Systeme wie eine Geschwindigkeitsregelung bei sporadischer Verletzung temporär in einen kritischen Zustand wechseln oder stabil bleiben. Oftmals wird Determinismus auch nach Kriterien

beurteilt, die sich nicht nach den physikalischen Voraussetzungen eines Systems richten.

Soll ein Projektleiter beispielsweise eine Abfüllstation für Getreide realisieren, bei der über einen Schieber der Auslass für das Getreide gesteuert wird, dann muss der Schieber bei einer definierten Füllhöhe geschlossen werden. Ein Sensor misst dabei die Füllhöhe des Getreides auf einem darunter stehenden Anhänger. Die Befüllung muss innerhalb von 15 Minuten erfolgen, wobei eine Reaktionszeit von 30 Sekunden (vom Auftreten des Sensor-Signals bis zum Schließen des Schiebers) noch akzeptabel ist. Ein Überlauf von Getreide führt zu keinem physikalischem Schaden des Systems. In dem obigen Beispiel kann die Entscheidung für das Betriebssystem XP durchaus richtig sein, wenn eine Überschreitung der Reaktionszeit tolerabel ist. Wird in dem selben Beispiel jedoch eine betriebswirtschaftliche Randbedingung eingeführt, kann die Entscheidung trotz gleicher physikalischer Voraussetzung falsch werden – beispielsweise, wenn der Kunde Garantieleistungen beim Überlaufen des Getreides fordert.

Es stellt sich somit die Frage nach der richtigen Beurteilung von Systemen. Nicht nur die Lösung selbst, sondern vielmehr die Analyse des Gesamtsystems muss für die Realisierung einer deterministischen Aufgabe angestrebt werden.

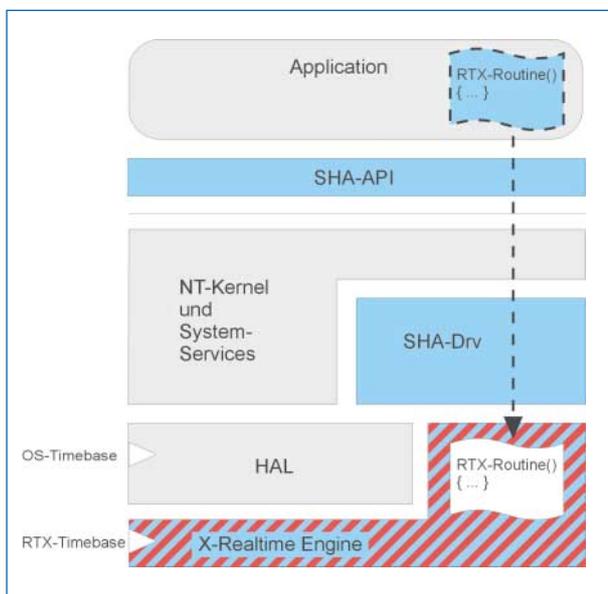
Die Forderung nach einem deterministischen Zeitverhalten bei Kommunikation und Datenaufbereitung umfasst nahezu den gesamten Schnittstellen-Bereich. Determinismus wird vor allem bei Feldbussystemen, serieller Kommunikation und Ethernet-Verbindungen gefordert. Hierbei muss jedoch zwischen zwei Ebenen des Determinismus unterschieden werden:

Realtime-Level 0

Es muss sichergestellt werden, dass bei der Kommunikation keine Daten verloren gehen können.

Realtime-Level 1

Es muss sichergestellt werden, dass die Datenaufbereitung (Filterung) in einem deterministischen Raster durchgeführt wird. Um



SHAX-Realtime-Engine

Grafik: Sybera



das Echtzeitverhalten nicht zu beeinflussen, muss der Datenaustausch zwischen deterministischen Tasks und nicht-deterministischen Tasks zeitlich entkoppelt werden – z. B. über Ring-Puffer. Ein Echtzeit-Betriebssystem muss zudem einen Notbetrieb gewährleisten, um im Fehlerfall die Systemstabilität aufrecht zu erhalten.

Grundsätzlich unterscheidet man bei der Echtzeit-Verarbeitung zwischen zwei Verfahren:

Umlaufende Echtzeit

Unter der umlaufenden Echtzeit versteht man ein deterministisches Multitasking, bei dem zwei unterschiedliche Mechanismen zur Anwendung kommen.

- ▷ Beim Non-Preemptive Realtime Multitasking werden Tasks periodisch durch einen Scheduler (Verteiler) in einem deterministischen Zeitraster aufgerufen. Bei dieser Methode richtet sich die Periodendauer des Verteilers nach der Ablaufzeit aller aktiven Tasks.
- ▷ Beim Preemptive Realtime Multitasking hingegen werden Tasks periodisch in einem deterministischen Zeitraster durch einen Scheduler unterbrochen. Die Periodendauer des Schedulers ist hierbei von der Anzahl der aktiven Tasks unabhängig.

Ereignisgesteuerte Echtzeit

Bei der ereignisgesteuerten Echtzeit wird eine Task innerhalb eines deterministischen Zeitfensters aufgerufen, wenn das Ereignis eintritt (z. B. Interrupt). Die Latenzzeit bis zur Abarbeitung des Ereignisses ist hierbei von Software- und Hardware-Verarbeitung abhängig (z. B. Prioritäten-Verteilung der Interrupts).

Sybera hat mit SHA (Sybera Hardware Access) eine Realtime-Engine entwickelt, die eine asynchrone Entkopplung (basierend auf zwei getrennten Taktquellen) der Echtzeit-Erweiterung zum bestehenden Betriebssystem ohne zusätzliche Hardware umsetzt und somit ein besonders geringes Jitter-Verhalten aufweist. Während bei herkömmlichen Echtzeit-Subsystemen die Latenzzeit mit zunehmender CPU-Belastung starken Schwankungen unterliegt, bleibt die Realtime-Engine von SHA auch bei extremer CPU-Belastung zeitstabil.

Vor dem Einsatz einer Echtzeit-Erweiterung empfiehlt sich die Analyse des Zeitverhaltens der Multitasking-Anwendung. So sollten neben einer Echtzeit-Erweiterung auch entsprechende Tools für die Echtzeit-Systemanalyse zur Anwendung kommen. Daher gibt es zu der Echtzeiterweiterung SHA mit der Software Kernscope ein Werkzeug zur Echtzeit-Systemanalyse. So lässt sich

z. B. der zeitliche Ablauf vom Eintreffen eines physikalischen Ereignisses (Interrupt) bis zur Abarbeitung per Software in Kombination mit dem damit verbundenen Multitasking-Verhalten analysieren.

Um die Komplexität solcher Systeme zu verringern und den Einsatz zu erleichtern wird Sybera in Zukunft eine Plattform mit Echtzeit-Cores anbieten. Ein Echtzeit-Core stellt hierbei eine erweiterte Abstraktionsebene (erweiterte Schnittstelle) für eine bestimmte Aufgabe dar. So sind unter anderem Echtzeit-Cores für RS232/RS485, Ethernet, CANbus, Profibus und weitere Schnittstellen zusammen mit entsprechender Hardware geplant. Ein weiterer Vorteil ergibt sich aus der direkten Umsetzung des Realtime Level 0, so dass der Einsatz deutlich günstiger passiver Peripherie (z. B. passive CANbus-Adapter) für deterministische Aufgaben ohne Datenverluste möglich wird.

www.sybera.de

Sybera **329**

Jürgen Rall arbeitet bei Sybera in Holzgerlingen