Entwicklung einer echtzeitfähigen CLI-Laufzeitumgebung für den Einsatz in der Automatisierungstechnik

Alexej Schepeljanski, Martin Däumler und Matthias Werner

TU Chemnitz, Fakultät für Informatik
09107 Chemnitz
{osh|mdae|mwerner}@cs.tu-chemnitz.de


1 Einleitung

1.1 Motivation


In dieser Veröffentlichung wird der Einsatz von C# auf einer SPS betrachtet. C# ist eine CLI-Sprache (CLI: Common Language Infrastructure) und in
ECMA-335 standardisiert [13]. Die prominenteste Implementierung dieses Standards ist das Microsoft .NET Framework [10].

Um eine in C# programmierte Anwendung auszuführen, wird der C#-Code zunächst in Common Intermediate Language (CIL) übersetzt. Das nun generierte CLI-Assembly benötigt zur Ausführung eine CLI-kompatible Umgebung. Auch die meisten Bibliotheken liegen in CIL-Form vor, so dass die Nutzung einer CLI-VM nicht vermieden werden kann.

1.2 Problemstellung


In dieser Veröffentlichung wird ein Ansatz vorgestellt, der den, durch JIT-Compilierung eingeführten, zeitlichen Nichtdeterminismus reduzieren kann.

1.3 Rahmenbedingungen

Das hier vorgestellte Konzept entstand im Rahmen eines Kooperationsprojektes mit dem Unternehmen SYS TEC electronic [14], das zum Ziel hat, eine neue Hochleistungs-SPS zu entwickeln. In diesem Projekt waren folgende Anforderungen gegeben:

A.1 Die SPS soll mindestens in C# programmierbar sein, wobei die Erweiterbarkeit für andere CLI-Sprachen und IEC-61131-3-Sprachen möglich sein soll.

A.2 Der Umstieg von den bestehenden SPS-Systemen soll einfach sein und in einer kurzen Eingewöhnungsphase gelingen.

A.3 Es sollen strikte Echtzeitanforderungen realisiert werden können, wobei die Zykluszeiten von 400µs erreicht werden sollen.¹

A.4 Als Hardware-Plattform ist mindestens eine auf 32-Bit x86-CPUs basierende Architektur vorgesehen. Die Erweiterung auf andere Architekturen (insbesondere ARM) soll möglich sein.


¹ Die SPS soll Ethernet POWERLINKs [4] einsetzen, für die diese Zykluszeit typisch ist.