



Diplomanden- und Doktorandenseminar
des Instituts für Informatik

Die Adaptierung des retargierbaren Little-C-Compiler zur Erzeugung von Binärcode für die Protection Enhanced Risc Maschine (PERM)

Dipl.-Inf. Sascha Lützel, FH Nordhausen

Der Vortrag beschäftigt sich mit der Anpassung retargierbarer Compiler an neue Prozessormodelle. Insbesondere wird das am Beispiel des LCC (Little C-Compiler) und dem PERM Prozessormodell gezeigt. Es wird dabei auf die Arbeitsweise des LCC eingegangen und auf mögliche Schwierigkeiten bei einer Portierung hingewiesen. Es wird ein kurzer Überblick über Alternativen zum LCC gegeben.

Die PERM ist die Protection Enhanced Risc Machine, die im Institut für Informatik entwickelt wurde. Sie ist ein 32bit RISC-Prozessor, der möglichst orthogonal und streng methodisch im Befehlssatz aufgebaut ist und mit wenigen wirksamen Prinzipien auskommt. Der Mikroprozessor ist als Experimentalplattform für neue Prozessor Erweiterungen gedacht. Im Vordergrund der Entwicklung standen die Optimierung auf Anwendungen im RTOS (Real Time Operating Systems) Bereich und die Eignung für die Programmiersprache C. Der Kerncode soll durchgängig in einem Takt laufen und eine hohe Vereinfachung bringen.

Der LCC ist ein von David R. Hanson und Christopher W. Fraser entwickelter retargierbarer ANSI-C Crosscompiler. Welcher in dem Buch „A Retargetable C Compiler: Design and Implementation“, für die Version 3.5 beschrieben ist. Der LCC liegt in der aktuellsten Version (4.2) vor, das Code Generation Interface ist mit der im Buch beschriebenen Version nicht kompatibel. Doch existiert ein, wenn auch unvollständiges Paper von Fraser und Hanson, was das Code Generation Interface der Version 4.2 beschreibt.

Bei retargierbaren Compilern gibt es verschiedene Ansätze, um sie anpassbar zu gestalten. Der LCC geht dabei nach dem Prinzip der Maschinenbeschreibungsdfilei vor, welche einen erzeugten Zwischencode in den Zielcode übersetzt. Der Gnu-C-Compiler verfolgt ein anderes Prinzip, und bewältigt die Anpassung über eine eigene Programmiersprache, die sogenannte „Register Transfer Language“.

Donnerstag den 29. Juni 2006
13.00 Uhr in Raum 106