

Bericht zur internen Praxisphase

Überblick

Meine Praxisphase habe ich intern an der THM verbracht, da mich die freiere Themenwahl mehr gereizt hat. Bei einer Praxisphase in einem Unternehmen wird zumeist ein Thema vorgegeben, was zumindest annähernd wirtschaftlich sinnvoll ist, was die Themenwahl indirekt auf modernere Technologien beschränkt. Ich habe stattdessen meine Praxisphase im Rahmen des EOS32/ECO32-Projekts unter Leitung von Herrn Prof. Dr. Hellwig Geisse verbracht, und hatte so die einmalige Möglichkeit, mich mit den Ursprüngen der modernen Unix/Linux-Welt auseinanderzusetzen, und zwar mit UNIX Version 7 aus dem Jahr 1979, hier in der Form der auf den ECO32-Processor portierten Version EOS32. Konkretes Ziel der Projektphase war sämtliche relevanten Build-Werkzeuge unter EOS32 mittels Native Compiling verfügbar zu machen, um die Übersetzung von C-Quellcode direkt in EOS32 zu ermöglichen.

Organisatorischer Ablauf

Meine Praxisphase fand komplett im Home Office und somit außerhalb der Hochschule statt. Dies ermöglichte mir einerseits ein sehr freies Arbeiten, setzte aber auch andererseits ein hohes Maß an Selbstdisziplin voraus. Unterstützt wurde dieses freie Arbeiten durch regelmäßige wöchentliche Projektbesprechungen mit anderen Studierenden und Herrn Geisse über BigBlueButton. Einerseits war die Möglichkeit eines direkten Austauschs sehr sinnvoll und motivierend, andererseits sorgte die Regelmäßigkeit dafür, dass man stetig am Projekt weitergearbeitet hat, ohne es dank der freien Arbeitseinteilung zu lange aus der Hand zu legen. Besonders gut gefallen hat mir dabei die lockere Atmosphäre unter den Projektmitarbeitenden, aber auch gegenüber Herrn Geisse, der bei Bedarf sehr schnell auf organisatorische und technische Fragen via E-Mail geantwortet hat, oder sogar Einzeltermine in BigBlueButton angeboten hat.

Technische Umsetzung

Ziel des Projektes ist es, die gängigen Entwicklungswerkzeuge zum Übersetzen eines C-Programms unter EOS32 verfügbar zu machen, und zwar in solch einer Art, dass sie sich unter EOS32 selbst übersetzen können (also als „Native Compiling“-Version vorliegen). Dazu gehören insbesondere die folgenden Programme:

- LOC (Little C Compiler, ein Open Source ANSI-C-Compiler)
- AS (Assembler)
- LD (Linker)
- Make (Build-Management-System)

LCC, AS und LD lagen bereits als „Cross Compiling“-Variante vor, sodass die Quelltexte nahezu unverändert übernommen werden konnten.

Bei AS und LD handelt es sich dabei um komplette Neuentwicklungen innerhalb des EOS32-Projekts, da die alten UNIX-Originale auf die Architektur des DEC PDP11 ausgelegt waren, und unter der ECO32-Hardware weitestgehend nutzlos gewesen wären.

Beim LCC waren ein paar Änderungen erforderlich. Obwohl der Quellcode als solcher unter EOS32 ohne weitere Modifikationen zu übersetzen gewesen wäre, verursachte die schiere Größe des Projekts diverse Probleme. So reichte einerseits der Festplattenspeicher nicht aus, um die temporären Objektdaten abulegen, andererseits reichte der Arbeitsspeicher nicht aus, um sämtliche Quelltextdateien gleichzeitig zu verarbeiten. Natürlich hätte es Möglichkeiten gegeben, die Übersetzung dennoch mit der vorhandenen Hardware durchzuführen. Der naheliegendste Weg war jedoch, die Hardwarekapazitäten im ECO32-Simulator einfach zu erhöhen. Danach dauerte der Übersetzungsvorgang zwar immer noch ca. eine Viertelstunde, aber er lief ohne weitere Fehler durch.

Beim Make-Kommando gab es eine Besonderheit. Die „Cross Compiling“-Varianten nutzen das Make-Kommando des Hostsystems, in den allermeisten Fällen also ein modernes GNU Make. Ein modernes GNU-Programm mit all seinen Abhängigkeiten auf EOS32 zu portieren ist jedoch ein sehr gewagtes Unterfangen, weshalb die Entscheidung zu Gunsten einer Portierung des originalen V7-Make gefallen ist. Dieses war jedoch in K&R-C verfasst, sodass ein Übersetzen mit unserem ANSI-C-Compiler vielfältige Probleme bereitet hat. Nachdem alle inhaltlichen und syntaktischen Probleme des V7-Make behoben waren, liess sich auch Make erfolgreich übersetzen.

Zur Nutzung von diesem Make mussten die Makefiles, welche ja ansonsten von einem GNU-Make verarbeitet wurden, leicht angepasst werden, da sich im Laufe der Zeit auch in diesem Bereich Änderungen ergeben haben. Dies führt leider zu verschiedenen Versionen der Makefiles für die „Native Compiling“-Variante und die „Cross Compiling“-Variante.

Fazit und Ausblick

Mein Fazit der Projektphase ist, dass Unix V7 zwar heute keinerlei praktische Relevanz mehr besitzt, viele Ideen und Konzepte aber bis heute in modernen Unix- und Linux-Systemen auftauchen. Mein Verständnis von den grundlegenden Mechanismen innerhalb eines Betriebssystems ist auf jeden Fall an diesem Projekt gewachsen. Auch musste ich feststellen, dass man oftmals am Anfang eines Arbeitsschrittes noch gar nicht so genau weiß, was für Änderungen und Arbeiten überhaupt zu erwarten sind. Im konkreten Fall war

die Schwierigkeit mit den verschiedenen Versionen der Makefiles eine, die ich am Anfang in keinsten Weise erraten konnte.

Eine Weiterführung des Projektes könnte weitere Programme unter EOS32 übersetzbar machen, nicht nur die Programme aus der Build Toolchain.

Denkbar wären diverse Userland-Programme, oder sogar der Kernel.

Sebastian Walter, 3. Dezember 2023