

Wie FORTH entstand

Forth wurde von Charles H. Moore 1969 entwickelt. FORTH weist eine Reihe von Eigentümlichkeiten auf, die es stark von herkömmlichen Programmiersprachen unterscheidet. FORTH stellt i. A. nicht nur eine Entwicklungsumgebung, sondern auch ein Betriebssystem dar.

Diese Eigentümlichkeiten lassen sich gut aus der Entstehungsgeschichte erklären. Moore hatte zur Steuerung des Teleskops einer Sternwarte einen Rechner ohne Software gekauft. Er hatte sich vorgenommen, alle Komponenten selbst zu programmieren, die für eine komfortable Programmierung und den Betrieb des Rechners notwendig sind. Hierzu gehören ein Betriebssystem, eine Hochsprache und eine Entwicklungsumgebung. All diese Komponenten wurden innerhalb eines einzelnen Programms verwirklicht – dem Forth-System.

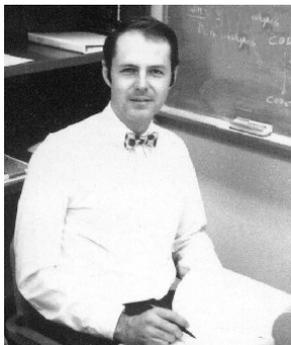


Abbildung 1

Moore erzählte später selbst: *Ich entwickelte FORTH im Laufe mehrerer Jahre als eine Schnittstelle zwischen mir und den Computern, die ich programmierte. Die traditionellen Sprachen lieferten nicht die Leistungsfähigkeit, Einfachheit oder Flexibilität, die ich wollte. Ich missachtete viele geltende Weisheiten, um exakt die Fähigkeiten einzubauen, die ein produktiver Programmierer benötigt. Die allerwichtigste davon ist die Möglichkeit, neue Eigenschaften hinzuzufügen, die später einmal notwendig werden. Als ich zum ersten Mal die Ideen, die ich entwickelt hatte, zu einer Einheit zusammenfasste, arbeitete ich auf einer IBM 1130, einem Computer der "dritten Generation". Das Ergebnis schien mir so leistungsfähig, dass ich es für eine "Sprache der vierten Computergeneration" hielt. Ich würde sie FOURTH genannt haben, aber die 1130 erlaubte nur eine Kennung mit 5 Buchstaben. So wurde aus FOURTH FORTH, immerhin ein nettes Wortspiel. (forth = vorwärts)*

(Zitiert nach L. Brodie: FORTH)

Wie MikroForth entstand

Eines Tages kam mein Sohn zu mir - er war gerade 14 Jahre alt - und fragte mich, wie man einen Compiler herstelle. Er würde gerne einen selbst programmieren. Nun hatte ich einmal gelesen, wie man Compiler baut; aber das war schon viele Jahre her und das meiste hatte ich wieder vergessen. Allerdings hatte ich sehr wohl noch in Erinnerung, dass der Compilerbau schon etwas komplexer ist und kaum etwas für einen 14-Jährigen. Und das sagte ich ihm dann auch.

Aber er ließ nicht locker. Einige Monate später - ich arbeitete gerade an einem Konzept für eine Mikrocontroller-Fortbildung - kamen wir auf die Idee, einen Compiler für den Mikrocontroller Attiny 2313 zu programmieren. Als Sprache wählten wir FORTH, nicht zuletzt wegen der einfachen Grundstruktur.

Unser FORTH-Compiler sollte allerdings nicht auf dem Mikrocontroller selbst laufen, sondern auf einem PC: Dieser sollte den FORTH-Code in Maschinencode umsetzen, welcher dann auf dem Attiny hochgeladen werden sollte. Der Compiler selbst ist relativ einfach, er greift auf eine Datenbank zurück, in der sich eine Sammlung von A-Wörtern (Befehlsfolgen in Maschinencode) und F-Wörtern (Befehlsfolgen in FORTH-Code) befindet. Da es uns hauptsächlich um das Prinzip ging, haben wir auch nicht alle gängigen FORTH-Wörter (Befehle) implementiert, sondern nur einen kleinen Bruchteil.

Es war äußerst lehrreich, derartige Programmschnipsel zu schreiben, und ich kann nur jedem raten, dies einmal selbst zu versuchen. Mikro-FORTH macht dies möglich; denn es ist ein offenes System: Die Datenbank kann nach eigenen Vorstellungen beliebig verändert und erweitert werden. Dies entspricht ja gerade auch der Moore'schen Leitvorstellung.

Faszinierend für uns war insbesondere folgender Umstand: Als wir mit dem Projekt begannen, kannten wir nur einige wesentliche Eigenschaften von FORTH. Später - als das Projekt fast fertig war - haben wir einmal recherchiert, wie FORTH im Original beim Compilieren vorgeht. Und siehe da - wir fanden einige unserer Ideen wieder.

Was die Betrachtung von FORTH leistet

- Verständnis für einfachen Compiler
- Verständnis für Assembler und Maschinencode
- Verständnis und Übung im Umgang mit Stapeln
- Verständnis mit der Übergabe von Parametern
- last but not least: effizienten Maschinencode