volksFORTH Version 3.90 Manual (work in progress)

Table of Contents

- volksFORTH Version 3.90 Manual (work in progress)
- Prolog
- Interpreter and Compiler
- Stack
- Assembler
- Vocabulary Concept
- FORTH-Files
- Why do we make this system freely available?
- Why can you program in VolksForth83?
- Notes of the lecturer


The authors have made every effort in this manual for a complete and accurate representation. The information contained in the manual, however, serves only as a product and should not be interpreted as guaranteed characteristics in the legal sense. Any claims for damages against the authors for whatever legal reason are excluded. It is not guaranteed that the procedure given free of third party rights.

We thank the entire FORTH community, particularly Charles Moore, Michael Perry and Henry Laxen

Prolog

volksFORTH83 is a language which is unusual in several respects. For FORTH itself is not only a language but a programming system, which is in principle, boundless. One of the main features of the FORTH programming system is its modularity. This modularity is ensured by the smallest unit of a FORTH system, the WORD.

In these terms a FORTH procedure, routine, program, definition and command are synonymous with all the needed words. FORTH is therefore, like any other natural language also made up of words.

These FORTH-words can be seen as already compiled modules, it being always extended a core of several hundred words through their own words. These core are in a FORTH83 standard set and ensure that standard programs can be run without changes on the respective FORTH system. It is unusual that the program text of the core is a FORTH program itself, in contrast to other programming languages, where it is machine language based. This core is produced by a special FORTH program, the meta-compiler, the executable Forth system.

How do you add your own words to the runnable kernel? The creation of new words is introduced in FORTH with COLON ":" and semicolon ":;" complete.

? expected: in the execution of a name and assigns that name to all subsequent words. ? completed and this assignment of words in the name of the new word and is ready for calls under that name.
Interpreters and Compilers

A classic FORTH system is always both an interpreter and a compiler. After Einschaltmeldung or pressing the button waiting for the FORTH interpreter with the FORTH-typical "ok" for your input. You can write one or more command words in a cell. volksFORTH starts after pressing the return key with his work in helping processed by the series after each command in the command line. These command words are delimited by spaces. The delimiter (Limiter) for FORTH procedures is, therefore, the space, which also has the syntax of the language FORTH would be described.

The compiler in a FORTH system is thus part of the interpreter interface. There is therefore no compilation process to create the code, as in other compiled languages, but the interpreter is secured with new words as necessary to solve the problems the user programs.

Even ":" (COLON), and ";" (semicolon) are compiled words, which turn off the compiler for the system on and off. Since even the words control the compiler, "normal"-FORTH words are missing from the usual FORTH compiler options in other languages or compiler switches. The FORTH compiler is controlled by FORTH-words.

The call of a FORTH-word is by its name without an explicit CALL or GOSUB. This leads to the FORTH-typical appearance of the word definitions:

```
  : <name>
    <word1> <word2> <word3>  ... ;
```

The standard system response in FORTH is the famous "ok". There is no requirement characters like 'A>' for DOS or ']' in good APPLE 11 is not it! This can lead to. that after a successful action, the screen completely blank, true to the motto:

*No News Is Always Good News!*

And - unusually - FORTH used the so-called Postfix Notation (RPN) is similar to HP calculators, which in some circles are very popular. This means FORTH always expects only the arguments, the action ... Instead of

```
3 + 2 and (5 + 5) * 10
```

It means

```
2 3 + and 5 5 + 10 *
```

Since the expressions are evaluated from left to right, there are no brackets in FORTH.

Stack

Equally unusual is that FORTH performs only actions explicitly requested: the result of your calculations will remain in a special area of memory, the stack, right up there with an output command (usually "will.") issued on then screen or printer. As the words of the FORTH-subroutines and functions meet other programming languages, they also need the ability to receive data to process and store the result. This function takes the STACK. In FORTH parameters for procedures are often stored in variables, but mostly passed on the stack.

Assembler

Within a FORTH environment can be instantly programmed into the machine language of the processor without having to leave the interpreter must. Assembler definitions are the equivalent programs FORTH FORTH-words.
Vocabulary Concept#

The state-FORTH has an advanced vocabulary structure that was proposed by W. Ragsdale. This vocabulary-concept allows the classification of the FORTH-words in logical groups.

This allows you to switch on when needed and necessary commands disconnecting after use. In addition, the vocabularies allow the use of the same name for different words, without getting into a name conflict. An approach in action ähnliche catfish offers the UNIT-concept or MODULA PASCAL compiler or the packages in Java.

FORTH-Files#

FORTH often uses special files for its programs. This is a historical basis and the legacy of a tent when Porth very often took over functions of the operating system. Since there were only FORTH systems, the mass completely without even a DOS operating system or intervening management and file structures for your own use. These files are so-called block files and consist of a series of large blocks of 1024 bytes. Such a block, which is often called SCREEN is the basis of the source text editing in FORTH. However, with the volks4TH normal files can be edited in the format of the native operating system (MS-DOS, Atari TOS, Apple DOS, AMS-DOS, CP/M there ...), so-called "stream flow”.

In general, any language is behind a certain concept, only with knowledge of this concept is possible to use a language effectively. The language concept of FORTH is described in the book "In FORTH think ’ (Thinking Forth) by Leo Brodie (Hanser Verlag).

A first impression of volksFORTH83 and our pride in this prologue is intended to provide. volksFORTH83 is an "open source” system, with its performance, the question arises:

Why do we make this system freely available ?#

The spread, which has found the language FORTH was significantly linked to the existence of figFORTH. Also figFORTH is an open-source program (previously this was called public domain, but today the term is open-source more accurate), ie it must be inclusive of the source text passed and copied. Nevertheless were unfortunately various providers to easily adapt the figFORTH on different computers can pay very expensive.

Established in 1979, published figFORTH is no longer the case currently, as caused by the further spread of Forth an abundance of elegant approaches, some of Forth83 in the standard and the ANSI Forth Standard have been included. It was then of Laxen and Perry wrote the F83 and disseminated as a public Domaln. This free-FORTH '83 standard for MS-DOS, with its numerous utilities quite complex and is not available with manual.

We have developed a new Forth for different computers. The result is the volksFORTH83, one of the best Forth systems, there is. The state-FORTH is to build in the tradition of the above systems, in particular the F83, and promote the spread of the language FORTH.

state-FORTH FORTH was ultra under the name initially written for the C64. After publication of the Atari ST computer series, we decided to develop it for a volksFORTH83. The first shipped version 3.7 was that concerned editor and mass storage, still heavily based on the C64. It contained, however, have an improved tracer, the GEM library and other tools for the ST. The next step consisted in the integration of the operating system files. Source texts could now be processed from the desktop and other utilities. The third adaptation of the state-FORTH was designed for the CP/M machine (8080 processors), which is specifically for Schneider CPC also supports the graphics capability. Then the state-FORTH for widespread computer of the IBM PC series was adapted.
In the 90 years, computers have been set with many megabytes of main memory and disk space to Stabndard, and graphical operating systems like Windows or MacOS through.

But state-FORTH is now still interesting for computers with limited system resources, be it on old home computers from the 80s or PDAs and mobile phones.

People Forth is in version 3.90 available for the following computer systems:

- 6502 CPU
  - Commodore C64
  - Commodore C16
  - Commodore Plus4
  - Atari XL / XE
  - Apple I
  - Apple II
- Z80 CPU
  - CP / M
  - Amstrad / Schneider CPC under AMS-DOS
  - Amstrad NC100
  - Research Sinclair Z88
- 8088 CPU (Intel / AMD)
  - MS-DOS (in a DOS-BOX under Windows, Linux, OS / 2, MacOS)
- 68000 CPU
  - Atari ST

Why can you program in VolksForth83?#

The volksFORTH83 is an extremely powerful and compact tool. By resistant runtime library, compiler, Editor and debugger, the tiring ECLG-cycles ("Edit, Compile, Link and Go") is unnecessary. The code module is developed for the module, compiled and tested. The integrated debugger is the perfect test environment for Forth words. There are no huge hexdump or assembler listings that have little resemblance to the source text. Another important aspect is the multi-tasking. As you divide a program into individual, independent modules, or words, one should also available in single, independent processes can be divided. This is not possible in most languages. This has volksFORTH83 a simple yet powerful multi-tasker.

Finally, has the volksFORTH83 still a wealth of details that do not have the other FORTH systems:

- The vectors used in many places and so-called deferred words, allow an easy transformation of the system for different device configurations.
- It has a heap of "nameless" or words to code that is needed only temporarily. The block mechanism is so fast that it makes sense for large amounts of data processing, which are present in files that can be used.
- The system includes Tracer, Decompiler, Multi Tasker, assembler, editor, printer interface ...
  The volksFORTH83 produced, compared with other FORTH systems, relatively fast code that is slower than that of other compiled languages.

This guide is designed to support the yet to be completed volksFORTH83. The FORTH Society, a nonprofit organization, offers the platform. It gives the club FORTH magazine "FOURTH DIMENSION" out. Forth The Company may be accessed through the website http://www.forth-ev.de.

Notes of the lecturer#

This guide is intended to volksFORTH83 both as a reference and as a textbook for FORTH (especially state-FORTH). Therefore, it is not, look like the other ethnic FORTH manuals, a collection
of vocabulary. Instead, with detailed descriptions and programming examples in many chapters the possibilities of the FORTH system is explained. The chapters are supplemented by each word descriptions of the commands that occur in (Glossary). To distinguish between description, FORTH-words, program inputs and expenditures will be working with different character types:


On with Chapter 2.