Forth in the U.S.S.R.

S.N. Baranoff

Leningrad Institute for Informatics and Automation
USSR Academy of Sciences
14 liniya 39, Leningrad, 199178, U.S.S.R.

Abstract

Although the Forth programming language was first introduced to the USSR beginning in the early 80's, several researchers had already developed systems based-upon threaded interpreters, address and text interpreters, and continuous toggling between compiling and interpreting states. The August 1980 *Byte* magazine issue on Forth and the FIG and Laxen-Perry F83 models were the basis for subsequent work. Since then, Forth has been written for a variety of computers including ones based-upon the IBM 360, DEC PDP-11 and Intel 8080 processors. It has been used for a variety of applications incudling symbolic computing, compiler writing, embedded applications and video games.

– *(Lawrence P. Forsley)*

Information about Forth first reached me in 1982, but long before that many teams in this country had looked for principles that now form an image of the Forth language and its way of programming, some starting from theoretical issues, others from their practical needs. As a result, by 1983 there existed such systems as DSSP (Dialog System for Structured Programming, Moscow State University **[1]** by N.P. Brussentsoff et al.), COMFORT (Leningrad Polytechnical Institute, Leningrad **[2]** by V. P. Kotlyaroff et al., "Work Mess" System (Computing Center of the Siberian Branch of the USSR Academy of Sciences, Novosibirsk **[3,4]** by A. A. Baehrs et al.) and others that independently developed the ideas of threaded code, address and text interpreters, partial parameterization, continuous toggling between compiling and interpreting, strong notion enveloping, etc.

My personal experience at that time was with a team at the Leningrad State University developing software for the K580 microprocessor (a Soviet analog to the Intel 8080) and so-called intelligent terminals that were based on it. The terminals (designated ES-7961) consisted of the microprocessor and 32K (now 64K) of storage, and were connected through a control unit (ES-7970, also with a K580 microprocessor and 4K of storage) to an ES mainframe (an IBM/360 compatible). We only had an ES-1033 mainframe with 512K of core storage running under OS/360 MVT 6.1 at our disposal; the target hardware was being developed in parallel with our work. So we first wrote a rather decent cross-system **[5]** in PL/I running on the ES mainframe and intended to develop the target software for the K580 with its assembler and a specially designed high-level macro-assembler. (As it later turned out, this macro-assembler, named ASSOL/M, was quite similar to the PL/360 language developed by N. Wirth).

Upon finishing that work and gaining much experience in developing and debugging our target software for the K580, we came across the August 1980 special Forth issue of *Byte* magazine which contained a listing for a built-in Forth assembler for the Intel 8080. (Unfortunately, the photocopies of several articles from that issue of *Byte* passed through many hands and were finally lost. Now I can't identify the author. A regular *Byte* subscription was obtained by some Soviet libraries only after 1984-1985, and only after a long struggle.)

The most striking thing for us was that the assembler used only a single page (about 40 lines) of source code; it was a great challenge to our own PL/I program that needed 180K of core storage to run. Unfortunately, (or perhaps fortunately?) the articles had such poor explanations of the Forth approach and Forth words that we had to think over the listings, trying to understand how they worked. When we understood, we decided to make another system based on what we had discovered of the principles of threaded code, `<BUILDS-DOES>` (now `CREATE-DOES>` in Forth-83), reverse Polish notation with its stack machine, and so on. As I was personally mostly interested in mainframe programming, I started my own FORTH-ES system **[6]**, having in mind to turn the mainframe into a gigantic personal computer. (At the beginning of the 1980's the ES was the most available computer environment for many people.) At the same time my university colleagues went on with the K580 line **[7]**.

When we later got a FIG-Forth description **[8]**, it turned out that nearly everything was designed similarly, the main difference being the search order and the treatment of context switching within a colon definition. In 1984, we got a copy of the Forth-83 standard **[9]** and quickly adjusted our implementations accordingly. Thus our small university team became a center for Forth propaganda within the Soviet programming community. We distributed our systems freely, delivered many talks at conferences, and established a sort of Forth "lobby".

Along with the self-developed implementations, the FIG-Forth system was spreading on SM-4 (Soviet PDP-11 compatible) minicomputers, especially after the appearance of a variety of native small computers with the same instruction set. People working with this computer type showed great interest in Forth and developed their own implementations, starting from FIG-Forth (thanks to the detailed comments in the source assembler code, it was an exciting and quite fruitful job). I would like to mention here our Estonian colleagues in Tartu (M. Tombak, J. Peyal, and V. Soo of the Tartu State University) and in Tallinn (L. Voehandu and A. Vooglaid of the Tallinn Polytechnical Institute and A. G. Astanovsky and V. N. Lomunoff of the Institute of Cybernetics of the ESSR Academy of Sciences).

As the public interest in Forth increased, a special subgroup on Forth was established in 1985 within the Working Group on Programming Technology for Microprocessor Hardware. The group is a volunteer organization under the State Committee on Computing and Informatics that organizes exchange of information between its members, thus forming the "public opinion" on programming technology issues. It holds annual conferences in various places in the U.S.S.R. that attract people from industry, high-schools, etc., and it publishes ad hoc proceedings. The aim of the group is to turn the current computer science achievements into public property by studying and synthesizing the experience gained from submitted reports and papers.

I now have the honor of being the Chairman of the Forth subgroup and see my task as propagating the Forth style of programming. Today the programming community of the U.S.S.R. is well enough informed of the existence of that approach. We now need particular systems, and real results showing considerable gains in programmers' productivity and the quality of their products. As was summed up at our Annual Working Group Conference in 1987, "The time for just saying that Forth is a good new thing has gone. The Forth enthusiasts should show systems already made up according to Forth technology if our programming industry is to be interested in turning this way." The areas for using Forth in the U.S.S.R. are now mostly in making compilers, built in applications, and video games. Special interest is paid to designing a Forth computer on the given element base **[10]**.

As for the literature on Forth in Russian, at present there is only one book **[11]** and several articles scattered in different volumes, most of which have been mentioned here. It is planned that *Starting Forth*, by Leo Brodie will appear in translation in 1989, and his *Thinking Forth* is under discussion (both at the Mir Publishing House, Moscow). Due to specific conditions in this

country, programming information on Forth circulates mostly in the form of narratives of its adepts and as computer printed material on particular systems. (I was both surprised and pleased to learn that my book of Forth **[11]** was typed in at ES terminals by Forth enthusiasts of the Rostov State University, Rostov-upon-Don, and is now well known at many computer centers with ES mainframes.) I have found it encouraging to teach Forth at high schools.

Finally I would like to enumerate the Forth systems developed by Soviet programmers and that are in use within the U.S.S.R. They are grouped according to their host computer types.

The PDP-11 Line (SM-4, SM-3, DVK, BK-0010, Elektronika-60, and many others)

1. FORTH-SM — in use since 1985 **[12]**, implements Forth-83, the dictionary comprises over 350 words, size, 10.5K. It may be run under RSX-11, RT-11, or stand-alone. Indirect threaded code is used. The system supports interaction with OS files, a structured assembler with labels, screen and line text-editors, and a target compiler.
2. FORTH-TARTU by R. Vainaste and A. Juurik — in use since 1983 under RSX-11 and since 1985 under Unix; the dictionary comprises over 270 words; size is 8K; it is an extension of FIG-Forth. Direct threaded code is used. The system is supplied with an assembler and a screen text-editor. It is used in research **[13]** and for teaching.
3. COMFORT by V. P. Kotlyaroff et al. of the Leningrad Polytechnical Institute **[14]**. A quite original system that uses both direct and indirect threaded code and strong parameter checking. The system is called by its authors "an opened system on threaded code" and implements an interesting technical approach based on big libraries of small program modules **[15]** for real-time and control applications.

The Intel 8080 Line (A great variety of computer devices with the K580 microprocessor, among them ES-7970 and SM-1800 under CP/M and stand alone, K1-10, and school computers Yamaha and Korvet.)

1. FORTH-K580 in use since 1983 **[7,16]**. It implements Forth83 with direct threaded code. It is supplied with packages for floating-point arithmetic, a Dialog Forth manual, an assembler, screen editor, and target compiler for ROM and RAM combinations. The nucleus comprises 300 words with a size of 9K. The system has been a basis for extensive research, including a full PASCAL compiler in 22K **[17]**, special means for databases and experiments in logic programming, expert systems, and the object-oriented approach **[18,19]**.

The Personal Computer Line (A variety of devices that may be treated as personal computers.)

1. FOS (Forth Operating System) has been in use since 1986 **[20]**. It is a rather developed operating system for the 16-bit ISKRA-226 computer, comprising over 400 words and being an extension of FIG-Forth with direct threaded code. It supports a parallel task scheduler, file and database manager, 128K address space, floating-point arithmetic and elementary mathematical functions, and window options for its screen editor.
2. FORTH-M6000 has been in use since 1985 **[21]** for the M6000 and SM-2M computers (HP compatibles) under DOS RTE-2; it is a derivative of the FORTH-K580 implementation.
3. Astro-FORTH by I.R. Agamirzyan and G.M. Shuvaloff of the Institute for Theoretical Astronomy, USSR Academy of Sciences, Leningrad. It has been in use since 1987 for the ES-1840 (IBM PC compatibles under MS-DOS 3.0+). It implements FORTH-83 with indirect threaded code and is greatly extended for a multiwindow user's interface, assembler for the 8086–8087, multitasking, a variety of graphical modes, a sophisticated Forth decompiler, floating-point arithmetic, menu handling, and helping systems both in

Russian and English. It has an MS-DOS file interface, background sound generation, a music compiler, and a target compiler.

The Mainframe Line

1. FORTH-ES **[6]** in use since 1984 for ES Ryad 1 (IBM/360 compatibles) under OS/360 and ES Ryad 2 (IBM/370 compatibles) under CMS. It implements Forth-83 with indirect threaded code. The nucleus comprises 350 words with a size of 13K. It is supplied with packages for assembler, floating-point arithmetic, helper and manual, editor, a window interface, and a target compiler.
2. FORTH-BESM-6 by I. R. Agamizyan. It has been in use since 1984 for the original Soviet BESM-6 mainframes under OS DISPAK; it is a derivative of FORTH-ES. The nucleus comprises 500 words with a size of 24K bytes (the computer is word-oriented, each address pointing to a word of 6 bytes). The system uses subroutine threaded code, which allows direct assembler insertions within colon definitions and calls to colon definitions from within assembler codes.
3. FORTH-ELBRUS by A. E. Solovyev. It has been in use since 1986 for the Elbrus mainframe (which has an architecture similar to that of a Burroughs). It works under OS Elbrus. The nucleus comprises 200 words, implements Forth-83, and supports interaction with procedures written in Elbrus' high-level assembler language El-76.

References

[1] N. P. Brussentzoff and A. M. Schaumann, eds. *Dialog Microprocessor Systems*. Moscow University press, Moscow, 1986. 148 pp. In Russian.

[2] Kotlyaroff, V. P. "A Technology for Developing Software for Built-in Microcomputers and the Supporting Tools." *Microprocessor Means and Systems*. 1987. #5, pp. 29-35. In Russian.

[3] Baehrs, A. A., V. G. Polyakoff, and S. B. Rudneff. "The Main Principles for Developing a Programming System for Microprocessors." *Software For Informatics Problems*. Novosibirsk: Comp. Centre Sib. Branch USSR Acad. Sci., 1982. pp. 5-25. In Russian.

[4] Rudnev, S. B., and V. A. Chetvernin. "Portable Work Mess as a Nucleus for Personal Computer Software." *Personal Computers in Informatics Problems*. Novosibirsk: Comp. Centre Sib. Branch USSR Acad. Si., 1984. pp. 85-72.

[5] Baranoff, S. N. et al. "The K580/LGU Cross-system." *Control Systems and Machines*. 1983. #3, pp. 19-21. In Russian.

[6] Baranoff, S. N. *The FORTH-ES System*. Len. Int. Inf. Autom. USSR Acad. Sci., Leningrad. 1986. 48 pp. (Preprint #1). In Russian.

[7] Baranoff, S. N., V. A. Kirillin, and N. R. Nozdrunoff. "An Implementation of the Forth Language at the Display Station ESv7970." *Programming Microprocessor Devices*. Inst. Cyber. ESSR Acad. Sci. Tallinn. 1984. pp. 41-49. In Russian.

[8] Ragsdale, W. F., ed. *FIG-Forth Installation Manual*. Forth Interest Group, San Carlos CA. 1980. 61 pp.

[9] *FORTH-83 Standard*. Forth Standards Team. Mountain View Press. 1983. 82 pp.

[10] Astanovsky, A. G., and V. N. Lomunoff. "A Threaded Code Processor." *Interpretative Programming Systems and their Hardware Support*. Inst. Cyber. ESSR Acad. Sci., Tallinn. 1988. pp. 24-31. In Russian.

[11] Baranoff, S. N. and N. R. Nosdrunoff. "The Forth Language and its Implementations." *Mashinostroyeniye*, Leningrad. 1988. 159 pp. In Russian.

[12] Katzeff, S. B. and I. A. Shendrikoff. "FORTH-SM — an Implementation of the Forth Language for Computers with the SMv3/4 Instruction Set." *Interpretive Programming Systems and their Hardware Support.* Inst. Cyber. ESSR Acad. Sci., Tallinn. 1988. pp 13-18. In Russian.

[13] Tombak, M. O., J. P. Peyal, and V. K. Soo. "Usage of an Extendable Language in a Compiler Design System." *Proc. Comp. Centre Tartu Univ.* vol. 52, 1985. pp. 39-54. In Russian.

[14] Kotlyaroff, V. P. and A. E. Pitko. "An Instrumental System Supporting a Fragment-modular Programming Technology for Built-in Microcomputers." *Instrumental Support for Programming.* Len. Inst. Inf. Autom. USSR Acad. Sci., Leningrad. 1988. pp. 88-104. In Russian.

[15] Kotlyaroff, V. P. "A Fragment-modular Programming Technology for Microcomputers within Control Applications." *Programming Technology Problems.* Len. Inst. Inf. Autom. USSR Acad. Sci., Leningrad. 1988. pp. 74-105. In Russian.

[16] Klubovich, A. A. "Target Compilation at Microprocessor K580." *Interpretative Programming Systems and their Hardware Support.* Inst. Cyber. ESSR Acad. Sci., Tallinn. 1988. pp 40-43. In Russian.

[17] Kirillin, V. A. "An Instrumental System for Developing Formal Language Means for Microprocessor Hardware." Ph.D. Thesis, Len. State Univ., Leningrad. 1985. 90 pp. In Russian.

[18] Sacharov, A. L. "D/M: An Interactive Language for the Modular Construction of Programs." Univ. Hamburg, Fachbereich Informatik FBI-HH-B-112. 1985. 24 pp.

[19] Pliss, O. A. "MEDIUM — An Object-oriented System." *Instrumental Means for Programming Support.* Len. Inst. Inf. Autom. USSR Acad. Sci., Leningrad. 1988. pp. 108-123. In Russian.

[20] Boldyreff, A. Yu., N. N. Veretyenova, and H. V. Lezin. "FOS — An Interactive Programming System for ISKRA-226 Computers." *Inst. Soc. Econ. Probl.* USSR Acad. Sci., Leningrad. 1986. (Preprint). In Russian.

[21] Patrysheff, B. N. "A Forth System for the ASVT Line." *Interpretive Programming Systems and their Hardware Support.* Inst. Cyber. ESSR Acad. Sci., Tallinn. 1988. pp. 19-23. In Russian.