Using KDF9 Fortran to implement APT IV in 1965-6

Miles Ellis

English Electric KDF9

- Scientific transistor-based multi-programmed scientific computer introduced around 1961
- Maximum of 32K 48 bit words of magnetic core memory
- Stack (nest) based CPU (four sets of nests)
- Paper tape input
- Magnetic tape backing store
- Usercode or Algol programming

KDF9 Egdon OS

- Developed by AWRE around 1964-5
- Punched card input
- Magnetic disc backing store as well as tapes
- A major feature was independent compilation of procedures to RLB
- A FORTRAN compiler was available called EGTRAN

EGTRAN

- Based on FORTRAN II
- Extensions to provide
  - Variable dimension arrays
  - Recursive subroutines
  - etc
- Well-defined and easy to use interface with Usercode procedures

APT
(Automatically Programmed Tools)

- Created by Doug Ross at MIT in 1959
- Used to produce control tapes for numerically controlled machine tools (2-5 axis)
- High level programming language
  - Define geometry of machined part
  - Define required machining process
  - Specify machine-tool specific aspects

APT III

- In widespread use by mid 1960s
- Maintained and developed by IIT Research Institute under a multi-member sponsorship arrangement
- Written in FORTRAN II
- Developed on an IBM 7090 computer
- Difficult to port to a different type of computer

APT IV

- Announced in 1965
- Written in FORTRAN IV
- Intended to be easily implemented on different types of computers

APT IV

- Announced in 1965
- Written in FORTRAN IV
- Intended to be easily implemented on different types of computers

- EELM joined ALRP in 1965 and pledged to produce the first commercial implementation of APT IV

The APT IV Processor

- The APT IV processor was intended for implementation using either an interpretive or assembly approach based on the supplied Fortran ‘translator’ and an implementation-dependent post-translator
- In effect this translation phase was a compiler for a program written in the APT language using a ‘production’ method

The APT IV Processor (2)

- Output from the translation phase was a ‘main’ program which was then executed in conjunction with a large subroutine library to produce a CL File
- The CL File was processed by a further (IITRI supplied) Fortran ‘CL Tape Editor’ program to produce the CL Tape
- This was used by the Post-Processor to produce the machine tool control tape

KDF9 Implementation

- Work began in late summer 1965
- We decided to use an assembly approach requiring only a handful of Usercode procedures
- A team of three worked on the project and fully processed their first part-program on 31st December 1965!

Major problems

- Conversion from IBM 7090 Fortran IV to Egtran
- KDF9 programs were limited to ca. 25K instructions in memory at one time – in an 8K (word) section of memory
  - The compiled APT Subroutine library included an unsegmentable section well in excess of 25K instructions
  - We wrote our own 'binary operating system'

KDF9 Implementation (2)

- 3 man months were spent converting Fortran IV procedures to Egtran
- 3 man months were spent writing and testing the post-translator (in Egtran)
- 2 man weeks were spent writing and testing Usercode routines
- 4.5 man months were spent devising and writing the ‘binary operating system’

KDF9 Implementation (3)

- The experience gained during this implementation led to IITRI requesting EELM to second one of their team to them to help improve the base system
  - This was the first time any of the ALRP members had been allowed to help in APT development
- ME spent August to December 1966 in Chicago working at IITRI on APT IV!

Major Successes

- We did produce the first commercial implementation in the world of APT IV
- We also proposed some changes to the base system which enabled a complete Fortran interpreter to be produced by IITRI
  - This meant that a computer-independent system could be delivered that would compile an APT part-program without the need to write any implementation-dependent code

Conclusion

- APT IV was a huge program for its time
- The KDF9 implementation taught us many lessons about writing portable software
- I, personally, learned a very great deal about Fortran

And Finally …..

- APT and Fortran provided the impetus – both separately and together – for most of the major events in my subsequent career!
  - My videotape Fortran 77 course launched my activities in both video and textbooks
  - My research into user-adaptable numerical control language processors led to my Candidate of Sciences and PhD degrees
  - I spent fifteen years working on Fortran Standards!