

A LIFE WITH COMPUTERS

J. Richard Swenson

I programmed my first computer during a summer job in 1957 at Electric Boat Company in Groton, Connecticut. I was between my sophomore and junior years at MIT where I had learned some numerical analysis of systems of ODEs. I was hired and put to work. Yes, but on what computer?

The numerical analysis division at EB had a very smart Dutchman whose name I can't remember. He had lashed together two IBM 650s that were being used to do a lot of numerical computations. These were decimal machines with 2000 word drums. The word size was 10 digits as I remember.

We generally coded in SOAP, Symbolic Optimum Assembly Program, and then tinkered with the code to further improve the placement of instructions on the drum so that they would be 'right there' when the drum had rotated during the execution of a previous instruction.

I succeeded in getting numerical solutions for the equations of motion for Skipjack class submarines, an early nuclear sub intended for continuous operation underwater. I learned a lot more numerical analysis as well as the joy of working in assembly language.

Returning to MIT, I had my first chance to program an IBM 704. Believe it or not, I made a proposal, which, if it had been accepted, might have been the precursor for CAT. I worked in a lab where we performed X-ray treatment on cancer patients using a van de Graaf generator. The radiation was administered on the basis of x-ray photographs of the patient. We would locate bone, fatty tissue, etc. in the photographs, plan the variations in dosage depending on depth and intervening material, and manually adjust the beam strength in an attempt to get the most radiation on the tumor while the patient rotated under the stationary beam.

My proposal was to build a model of the patient's cross section using the photographs, and then let the beam be adjusted automatically via a computer to accommodate bone, fatty tissue, and vital organs. My proposal wasn't accepted. *C'est la vie.*

I returned to Electric Boat in 1958 after my junior year for a slightly extended period of time as I needed money and had other problems due to illness. I did statistical calculations on a small 704 using Fortran; not Fortran II or Fortran IV, just Fortran!

In September 1959, I started at the Courant Institute of Mathematical Sciences (CIMS) at NYU as a graduate student and was immediately put to work on a REALLY BIG 7094. 32K of 36 bit words, 7 index registers, and a 'special' feature - significance arithmetic. NYU also had some UNIVAC machines, and I regret never having saved one of the multi-pound metal tapes that these machines used.

There was lots of discussion about the idea of rounding errors in long, large scale calculations at that time. Jim Wilkinson led this discussion in England. IBM modified the NYU machine so that a special kind of 'unnormalized' arithmetic, called significance arithmetic, rather than standard normalized arithmetic could be used. Often calculations would be performed using both flavors of arithmetic. The goal was to gain a better understanding of the error propagation in calculations.

To help with the use of this feature, I built special versions of some of the standard Fortran functions, e.g., the trig functions, to incorporate the significance arithmetic. I also built tools to help perform normalized calculations using 'random' normalizations. The results of the latter were published in the *Comm. of the ACM* in a joint paper with Tom Hull of Toronto. Understanding rounding errors in calculations was a big topic.

I also did some computations in number theory. I wrote some cleverly optimized assembler code to compute the number of lattice points in a circle. We were trying to see if there was any numerical evidence for various estimates of the error term. The code was so efficient that we extended other published results by 4 or 5 orders of magnitude. The results were inconclusive and were published by Herb Keller and me in the journal *Mathematics of Computation*. We also showed that some earlier results produced in a Stanford thesis were simply wrong. Ah well.

I loved assembler language.

Then we got a REALLY BIG 1401 to support our 7094. To accomplish this, I built a program in SPS, an assembler language, that permitted the 1401 to read cards and write tapes for input to the 7094, AND simultaneously read output tapes from the 7094 to print results and punch cards. We were able to keep every device on the 1401 working. I still have that code somewhere in a box.

All of this occurred under the direction of Max Goldstein who died some years ago. Please get someone to write about him. Among many other things, Max worked at Los Alamos on the Manhattan project.

In 1967 I attended a NATO summer school in Denmark and had a chance to meet many of the computing stars of the time including Don Knuth who had just finished volume 3 of his magnum opus. I still have these books. I also heard Bernie Galler deliver the most humorous response to someone who criticized Bernie's pronunciation of British English.

CIMS moved to Control Data computers in 1966 as part of the AEC move to CDC super computers. We acquired serial 4 of the CDC 6400. As I recall; serial 1 was kept by CDC; serial 2 went to Brookhaven, I think; serial 3 wasn't delivered but became a spare parts device; and we got serial 4.

I wasn't a big fan until I started to learn how to program a pipelined computer. What a hoot. I built a number of matrix subroutines that performed an arithmetic operation every machine cycle thus masking all of the housekeeping that occurs in loops. But this was my swan song as I soon left NYU to go to the University of Toronto as a manager.

Two major events in my life occurred there. The first was that I became a devotee of relational data base theory and was fortunate enough to co-author papers on relational theory with Denis Tsichritzis, Phil Bernstein, and H-A Schmid that were given at the 1975 SIGMOD conference.

The other was that under the tutelage of John Wilson, I became enamored of APL. As a result of John's schooling and the cooperation of a friend, again whose name I have forgotten, we built an interactive system to manage space at the University using APL that made dumb terminals behave like interactive terminals. This system showed me just how important having active user participation was as we constantly (with APL this was easy) adjusted the system to the needs of the young women who operated it, none of whom had any interest in computers. The system ran for a few years after I left the University completely unattended.

A truly major influence on me while I was at U of T resulted from Jim Horning being in the Department of Computer Science. As far as I can tell, Jim knew everybody and was instrumental in bringing them to the University. Thus Dijkstra, Tony Hoare, Jerry Weinberg, Wirth, and many others visited us regularly. What a joy to be able to interact with these people. I even have a copy of Dijkstra's thesis that he autographed!

There were many interesting events in my career at U of T, such as installing 9600 bps modems in the early 70s to facilitate remote input-output to the University's computing systems. These were System 360s, a model 50 soon upgraded to a 65 and another model 65. They were installed under my direction as Planning Manager and we were early adopters of the University of Waterloo 'fast' compilers for FORTRAN and other languages that were used by an enormous number of students.

Even though I left the world of programming completely, I occasionally had recalls. In 2000, I was fortunate to find some old PDP 11 manuals that helped a project that I was managing figure out the floating point representation of 1970 era Digital Equipment machines. And in 2001 I was a guinea pig in the IEEE effort to develop a 'software engineering' certification. I took the test and was among those who passed and therefore became a "Certified Software Development Professional". Though my employer paid for me to take the test, they took no interest in this accomplishment. I have since wondered about the success of this program. I never renewed my Certification as I couldn't find anyone who even knew that it existed. *Sic transit gloria.*

I have now retired, but I maintain an avid interest in computing. My general feeling is that little of real significance has changed in the human aspects of the profession, though of course, the engineering changes have been huge. People just take longer to change, I guess.

As part of some later efforts, I often gave a short talk on 'the major events of computing.' Here is my view of these events:

Turing's work and the idea of the Turing machine

Compilers, e.g., FORTRAN and COBOL

Computing complexity (P vs NP, and all the various complexity classes)

Object oriented theory (SIMULA was invented in 1962 and was the basis for this idea)

Relational data base theory (most current dbs are not relational even if they say they are)

Graphic interfaces

Continuing circuit miniaturization

And, perhaps the new field of quantum computing could be added, but we'll see.

*J Richard Swenson
Unit C4
515 Newport Way NE
Issaquah, WA 98027
425.369.2146*

Source: <http://www.acm.org/ubiquity>