

News from Nan

COMPUTER HISTORY

MY HIGH-TECH ADVENTURE: CHAPTER 04, IBM SERVICE BUREAU

SEPTEMBER 30, 2018 | DICK

Dick wrote this post, which was part of a book, My high-tech adventure, to record his memories of working in the computer industry, starting in about 1960. It was first published in 2012.



Going to work at Service Bureau Corporation

I enter the computer business

My first job after I got my MA in Math in 1965 was to be a “scientific application programmer” at SBC (IBM’s Service Bureau Corporation) in Los Angeles at their Scientific Center. SBC also had other scientific (and commercial) offices in other cities. SBC was a wholly owned subsidiary of IBM that had been created to allow IBM to satisfy the terms of an antitrust consent decree between it and the U.S. Department of Justice. At that time, Control Data Corporation (CDC) was a major competitor for IBM, and CDC had claimed that IBM was “bundling” computers and programming (software) to give it an unfair advantage, so by the terms of the consent decree, IBM had to split the programming of computers off into its own company.

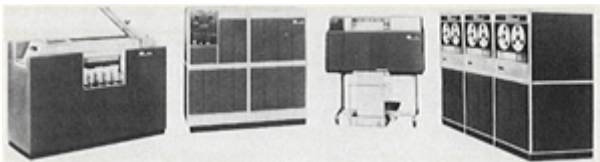
When I first got there, SBC sent me to a six-week class on the IBM 1401 and the IBM 7094 computers. These were the two machines they had in their machine room. My instructor was Harry King, who knew both machines as well as anybody could, and was a fantastic teacher. In the class, Harry taught us the complete instruction set for both machines, plus the FORTRAN language and the FMS and IBSYS operating system environments. We did daily programming exercises that were key-punched by the students and run on computers at SBC. These classes were very intense, and I enjoyed them a lot. I now had no doubt that computing was where I wanted to work. I was the top student in the class.

Before going further, I'll describe the hardware and software of the day.

IBM 1401 hardware and software

This machine was about the size of a large freezer. IBM sold thousands of them and made a lot of money from it. The SBC 1401 had between 4K and 16K of memory. It was a variable word-length machine, which made programming it tricky. The words were delineated by things called "word marks."

At SBC the 1401 was mainly used for reading card decks onto tape and for printing tape output from batch jobs run on the 7094. You could do simple commercial applications on the 1401, but SBC didn't use theirs for that. SBC ran a little program called the "4K Monitor" on the 1401 that provided the functions of card-to-tape, tape-to-print, card duplication, etc. The most common language for writing programs for the 1401 was called Autocoder, which I learned. If you tried to do anything very compute intensive, the machine would slow to a crawl. By modern standards, it was a very slow machine. I remember having a bug in one of my 1401 Autocoder programs that overlaid a work mark and then printed an entire box of blank paper before the operator noticed what was happening. That was really embarrassing!



IBM 1401 computer

SBC had a 1401 and 7094 II in one large room with a glass wall across the front. SBC's main business was writing and running scientific computer applications, and renting 7094 computer time. SBC made most of its revenue renting machine time. The 7094 rented for \$600/hour. Machine time was measured by stamping a timecard at the beginning and end of each job on a time clock. I was part of a 15-person programming staff. The operations staff was about the same size. Our branch office was about a mile from Los Angeles International Airport. We could watch the airplanes taking off and landing out our office windows.

Our staff (including me) used to wear suits and ties at all times. Our main 7094 operator was a particularly fancy dresser and wore a large diamond pinky ring. Nobody ever came to work wearing casual clothes, even late at night or on weekends.

When I first got to SBC, my office was in a building a few blocks away from the machines. When we wrote programs, they got keypunched by our keypunch operators and then the decks were taken by courier to the main building to be run on the machines. If we were lucky, we might get 2-3 “turnarounds” per day. We would have to wait hours just to find out we had made a typo! We got really good at visually checking our card decks for errors the machine might hit. When I was new at SBC, I had to write program flow charts before I even key-punched the program. The flow charts were checked by others. After I had been there a while, they quit asking me to do flow charts.

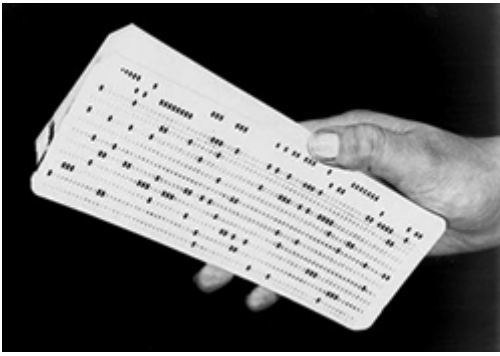
Programs were all stored as 80-column card decks. Once a program was debugged, it was kept for us in a file drawer by Jean, our program librarian, who stored all the programs in long file drawers. There was a catalog of the programs kept in a three-ring binder.

Card decks sometimes got dropped. In order to facilitate getting the cards back in order, we either sequenced the deck in the right-hand columns of the cards or we drew a diagonal line across the top of the deck with a marker pen.

“Jobs” were card decks with operating system control cards at the front followed by the Fortran source code and finally any input parameters. The cards were read on the 1401 and copied card-to-tape to produce a “job tape.”

When it was time to run your job, it was taken to the 7094 and the OS was loaded from a tape drive with your job on another tape drive. Output from your job was written to yet another tape drive. After the job finished, the output tape was taken back to the 1401, which used a tape-to-print utility to produce the printed output. You got back either output from your program or a core dump, which was memory printed in octal. The core dump meant your program had a bug and had crashed the operating system on the 7094. We would stand by the 7094 printer console when we had an important job to run and could tell quickly whether the job was running normally or had ended in a dump. As a courtesy, the operator would phone us in our office so we could attend critical job runs in the machine room.

The following figure shows an 80-column IBM card.



80 column punched card

My first failed project

Soon after I got to SBC, I was assigned to help a senior programmer debug and run a Fortran program that did grade reporting for several local Catholic high schools. The senior programmer got the code to the point where it compiled and ran, and then he disappeared for several months to attend the IBM SRI (Systems Research Institute) in New York City. After he had left we got the first live data from a local school and found out on the first run that things were not working right. I was the only programmer available to work on fixing the code, and I was a brand-new employee. After a few days I found out that almost every single subroutine in the code was full of bugs, and I had to rewrite and debug almost everything. Many long nights later we finally were able to give the customers correct results, but it was much too late to be useful. By the next semester we could produce results on time, but the cost to SBC was much more than we were able to charge the customers. In those days using 7094 for such a task was truly like using a sledge hammer to drive a tack. The contracts were not renewed after that first year.

For me the good thing was that I was able to show SBC that I had talent for writing and debugging software. I got promoted several times because of this “failed” project. From this I first learned that a “bad” situation might not always be bad for me personally!

Dr. Chuck Fillerup

After this failed project I worked under the direction of a mentor, Dr. Charles (Chuck) Fillerup. Chuck had been in the computer business for several years before I met him. He was one of the early pioneers. His specialty was numerical analysis and linear programming. Chuck used to talk a lot about how much “fun” he had programming the IBM 650. This machine had been introduced by IBM in 1955. It stored data and programs on a revolving magnetic drum. Programming it was very time critical. Under Dr. Fillerup I worked on several projects, including:

- CRISP: a program that calculated the critical speed of a rotating shaft. A critical speed is one where the shaft will tend to vibrate.
- A program that calculated loading on an air slider bearing (like read/write heads found in disk drives). This project was done for National Cash Register (NCR), then a significant computer company.

- Seat equivalence optimization for the Los Angeles Civic Light Opera using mixed integer linear programming. The Light Opera was opening a new theater and wanted season ticket holders in the new theater to have seats as good as what they had in the old theater.



Dr Charles Fillerup

I got a few letters from Chuck as late as 1972. He tried to convince me to go to work for Computer Usage Corporation in LA, where he had moved himself. His letters were usually from Milan, Italy, where he was working on a contract. I found out just recently that Chuck died in September 2002 (age 73) after having spent his last 7 years as a part-time member of the faculty at Cal State Long Beach. He had also been teaching Navy personnel in the Persian Gulf aboard ships. His students all said he was quite a guy. One thing he told me that I still remember: "If you stay some place long enough, they always take you for granted."

Chuck did not like being taken for granted.

Managers

My first manager at SBC was Tony Landi. He suffered through the failed grade reporting project with me. Years later I met Tony again in Kingston, NY, where he was a manager in IBM.

Our SBC office manager was Bob Maldonado. Bob taught me a good lesson I still remember: I asked him for permission to work on a small skunk-works project while I was between customer jobs. Bob told me, "Don't ask for permission unless you need to! You might get told no, and then what do you do?"

My last manager at SBC was Ken Tranbarger, who came to SBC from Litton. Ken taught me to write GPSS simulations in only a few days after SBC got a simulation contract from Litton. I wrote a simulation model of something similar to the AWACS airplanes of modern warfare. One strange thing about this project was that Litton had us changing constants in the model until the simulation got the right answer.

I thought the model was supposed to show whether the proposed system would work. Litton just wanted to get the contract!

The coming of System/360

After I had been at SBC for about a year, they took delivery of a small IBM System/360 model 30 computer. The IBM System/360 was an entire family of machines that had the same instruction set. The model 30 was one of the smaller models that SBC wanted to use to replace their 1401s. This machine had 64K of memory and a few tape drives attached to it. The model 30 ran a primitive operating system called BPS (Basic Programming Support). We used BPS, because IBM did not yet have OS/360 running. There was a period of time after it was first delivered when I got to operate the model 30 all by myself. This was my first “personal computer.”

In 1966, SBC replaced their 7094 with an IBM 360 model 40 coupled to an IBM 360 model 65. The model 40 did all the front-end work of reading in cards and printing output from the 65. The 65 did all the real computing. The software consisted of an early release of OS/360 and a software system called ASP (Attached Support Processor).

Attached to the model 65 were several 2311 disk drives (holding 7.25 megabytes each!). These were the first disk drives I had ever seen and I remember wondering how to use them instead of tape. The 65 had firmware that allowed it to emulate (theoretically) a 7094. We soon found that many of our customer 7094 programs did not run properly, and SBC very quickly lost most of their machine rental customers to other companies that still had real 7094s. We also found out that both OS/360 and ASP were very unstable. Today we would probably call this an alpha level system! For example, every time the card reader experienced an error reading a card, the whole ASP system would crash. This happened quite a lot and a system restart took at least 15 minutes. The other thing we found was that the 40 could not really keep up with the 65 when it tried to generate print spool output. This caused the 65 to sit idle a good fraction of the time.

I left SBC in 1967 and dragged my wife and family off to New Jersey to work at Bell Laboratories. Mainly I did this because I felt bad that my mathematics background was not being used enough. Also, things got a little slow at SBC and I got bored. I found out later that most of the people at my SBC office left not long after I did.

The following figure shows an IBM System/360 model 67 computer. The model 65 looked just like this.



IBM System/360 model 67

◀ 2018 ◀ AUTOCODER PROGRAMMING LANGUAGE ◀ CHUCK FILLERUP ◀ FLOW CHART ◀ FMS ◀ HARRY KING
◀ IBM 1401 ◀ IBM 2311 ◀ IBM 360 MODEL 65 ◀ IBM 7094 ◀ IBM ASP ◀ IBM BPS ◀ IBM SERVICE BUREAU
◀ IBM SYSTEM/360 MODEL 30 ◀ IBSYS ◀ MYHITECH ◀ SYSTEM/360