

IBM SYSTEM/360 LAID GROUNDWORK FOR MAINFRAME INNOVATION



With the [IBM® System/360](#) celebrating its 60th birthday this year, the BMC mainframe group was asked if anyone remembered working on this hardware. In a moment of weakness, I admitted that I had, and was asked to blog about what I remembered.

Although I took my first programming course at the age of 15 while still in high school, it was not until I was taking my first programming course as a freshman at the University of Michigan (U of M) that I used a System/360.

In fact, I interacted with two System/360 machines. The computer center used a System/360 Model 20 to run the card readers and printers that we used to submit all our programs for testing, and we'd get the results back on printed output. Our programs were executed on a System/360 Model 67 that was a very unique machine at the time. The 360/67 at the University of Michigan was the first IBM computer to have virtual memory.

The Model 67 was built to the specifications derived from a paper written in 1966 by four authors, [Bruce Arden](#), [Bernard Galler](#), [Frank Westervelt](#) (who were associate directors at U of M's academic computing center), and Tom O'Brian, called "[Program and Addressing Structure in a Time-Sharing Environment](#)". Dr. Galler and Dr. Westervelt were professors who I got to know during my time at U of M. I took my first advanced programming course, as well as the last course for my master's degree, from Dr. Galler. Fun fact: Dr. Galler's son, Glenn, works in the mainframe group here at BMC.

I got my first IT job the summer before finishing my master's degree at Project Management Associates (PMA), a subsidiary of a construction company, Townsend and Bottum, that specialized in building power plants. PMA hired me to do COBOL development on a scheduling system they were developing for the construction industry.

I had one minor challenge with this first job. Although I had learned Basic Assembler Language, FORTRAN, SNOBOL, PIL ([Pittsburgh Interpretive Language](#)), LISP, wrote an operating system, wrote a compiler for a language called GLORY, and used other programming languages at U of M, they did not offer a course in COBOL. I spent first week on my new job reading the manual and learning COBOL.

The COBOL programs I developed ran on the Townsend and Bottum data processing center System/360 Model 30 that had 8 kilobytes (yes, 8K) of physical core memory—no virtual memory. I designed an overlay structure into the program so that it reused the physical memory as the program executed. For example, once I did the initialization processing, I overlaid those in-memory instructions with the instructions to read the data and create the report. If the program produced multiple reports, I needed to design an overlay structure to reuse the application instruction storage from the first report for each subsequent one.

After graduating from U of M, I moved to the Townsend and Bottum parent company and continued to write COBOL programs. Those programs also ran on a Model 30 that used the Disk Operating System (DOS), a predecessor of Virtual Storage Extended (VSE). I remember when they upgraded the Model 30 from 8K to 32K of memory. This greatly reduced the overlay processing requirements in the programs.

A year or so later as Townsend and Bottum was expanding, the company decided to upgrade to a System/360 Model 50. At that time, it also decided it needed its own systems programmer, and I was willing to take the position, so the company sent me to a number of IBM courses to get the knowledge I would need for the job.

My first activity as the new systems programmer was to calculate the electrical power requirements for the new Model 50, the associated bank of eight 2314 disk drives (where each disk pack held 29 MB of data), the tape drives, printer, card reader/punch, and other peripherals so that the computer room could be designed with enough power to run the system.

Initially, the same DOS operating system that was on the Model 30 was used for the new Model 50, but shortly thereafter I installed the [Operating System/360](#) (OS/360) that was designed for the new hardware. The first OS/360 system I generated and installed was Multiprogramming with a Fixed number of Tasks (OS/MFT). Later, I upgraded the operating system to run OS/360 Multiprogramming with a Variable number of Tasks (OS/MVT). Both of these were non-virtual memory systems since the Model 50 did not come with a virtual memory capability.

After a few more years of business growth, the System/360 was too small and Townsend and Bottum moved to a System/370 machine, so I moved off the System/360 platform and on to this larger and faster machine that had virtual memory.

Thanks for indulging me on my trip down memory lane. It's amazing to think how far mainframe systems have come while maintaining their role as the system of record for the global economy. The original System/360 and the innovation of subsequent versions provided the foundation for modern mainframes and their utilization of cutting-edge technology, including artificial intelligence (AI). While it's exciting to see what is on the horizon for future versions of the mainframe, I would not

have the background and knowledge that I possess today if I had not started my career on the IBM System/360.