

## The IBM 360/67 and CP/CMS

Tom Van Vleck

IBM's 360/67 was a 360/65 with virtual memory operations added to the CPU. The machine supported paging and segmentation, 4096 segments each with 256 4Kbyte pages, and had other modifications that made it possible to simulate multiple virtual 360s efficiently. 🌐 Melinda Varian, in her outstanding memoir *VM and the VM Community*, tells how IBM didn't respond to the MIT Project MAC request for a next-generation timesharing machine in 1964. She tells how, too late, Gerry Blauuw of IBM made a proposal to Project MAC for a machine with some virtual memory features; and how the IBM Cambridge Scientific Center, located in Tech Square on the third floor, created the 44MPS and **CP/CMS** in the mid-60s.

When MIT and then Bell Labs chose GE machines for their next generation time-sharing systems, and the University of Michigan showed interest in Multics, corporate IBM woke up to the need for time-sharing and responded with the 360/67. IBM's concern over the "snowball effect" led them to announce plans to build a large-scale timesharing system, TSS/360, as described in an article by Judy O'Neill in the 1995 *Annals of the History of Computing*. So the 360/67 was the machine of choice for IBM's CP/CMS and TSS, as well as for the 🌐 Michigan Terminal System, MTS, which was begun about the same time. The 360/67 was announced in August 1965, and 360/67 serial #2 was delivered to the University of Michigan in January 1967.

The architecture of the CP/CMS system was quite different from other time-sharing systems of the day, which provided processor multiplexing, file system, and user command processing as an integrated system. Such systems provided each user with a "virtual computer," more powerful than the real computer the system ran on. CP-67, on the other hand, provided each user with a virtual IBM 360. CP made one real 360/67 look like multiple virtual 360s: it multiplexed the CPU and main storage, and statically divided the physical disk into small virtual disks called P-disks. CP also managed terminals and users, and let a user log in to a virtual machine controlled from the terminal. You could run OS/360 on such a virtual machine, and many users did in order to gain access to large (virtual) memory spaces. You could also run a single-user system that thought it was doing I/O to a console device: this was CMS. The strong architectural separation between CP-67 and its guest operating systems was a thing of beauty: CP concerned itself with user isolation, including paging memory and virtualizing device I/O, while 360 operating systems, including CMS variants, could be developed safely in isolation from each other.

Both TSS and Multics had problems delivering an effective system on schedule, while CP/CMS became available as an IBM Type III program and gained a following at a dozen or so installations in the late sixties. (A Type III program was "customer contributed" and IBM didn't support them or promise they'd keep working.) Dave Tuttle writes: "The CP-67/CMS system 'snuck' into the Type III library by collusion between the CSC and MIT Lincoln Labs; the public release was officially submitted by Lincoln Labs because IBM was no longer allowed to make it freely available."

One of these sites was the **MIT Urban Systems Laboratory (USL)**, an interdisciplinary research lab led by civil engineers and school of management professors. I took over administration of USL's CP/CMS system in April, 1969, from Betsy Schumacker when USL gave up their 360/67 (I think it was serial #10) to MIT's Information Processing Center. I worked with a team of system programmers who were as green as I was (we all had blue reference cards, mine might be around here somewhere) trying to satisfy the civil engineers etc who would have much rather have control of their own machine, except they couldn't afford it. I remember I wrote a program (in FORTRAN) to show how the cylinders of the physical disk were allocated to users' P-disks. Craig Franklin wrote a billing program that processed the CP-67 accounting cards.

CMS, in those early days, was not as elegant and rock-solid as CP. There were bugs in the file system, and although the virtual 360 had both user and system states, CMS didn't keep its file system tables protected from user programs: an errant application could destroy the entire contents of the user's P-disk. And no matter how often users were warned that they were responsible for their own backups, few bothered.

One day CP/67 crashed 27 times, and the users were lining up outside my door with torches & pitchforks. In those days we had real paper listings of CP/CMS and the programmers had them all spread out on a couple of tables, trying to analyze the dumps. I was not a 360 assembly code wizard myself but I did sit in on the debugging sessions in the "war room" on the fourth floor of building 39, asking dumb questions, fetching soft drinks, trying to encourage debugging progress. The crash was happening sometime after some kind of communications event, but the dumps were garbage. So I was looking at the code, and noticed something: the 2741 code and the Teletype code were slightly different. I forget the exact difference -- this was in 1970 or so -- but the missing instruction in the Teletype code could cause CP to try to move zero bytes of data into a buffer, and the opcode used would move 256 bytes if given a count of zero. When this happened, the long move scrambled the stack. The bug only occurred if a user program sent a Teletype an output line more than 80 characters long, which never happened with real TTYs since upstream code was inserting CRLFs. But on that day, a user at Harvard School of Public Health had connected a plotter to a TTY line and was sending graphics to it, and every time he did, the whole system crashed. (It is a tribute to the CP/CMS recovery system that we could get 27 crashes in in a single day; recovery was fast and automatic, on the order of 4-5 minutes. Multics was also crashing quite often at that time, but each crash took an hour to recover because we salvaged the entire file system. This unfavorable comparison was one reason that the Multics team began development of the New Storage System.)

There was a sort of informal race between the TSS/360 and Multics groups in the 1966-69 time frame, to see who would ship first and whose system would be better. TSS shipped first, and some people think it was a Pyrrhic victory, that the need to support customers early and remain compatible with the early first release retarded their later progress. The initial releases of both TSS and Multics had a reputation for slow performance and lots of frustrating bugs and loose ends. CP/CMS, meanwhile, spread rapidly; the O'Neill paper claims 66 TSS sites by 1972, but Lynn Wheeler believes that many of these were CP/CMS sites, counted for TSS by IBM doublethink (by 1972, TSS had been decommitted for 4 years). By this time, both National CSS and Interactive Data Corporation were providing commercial CP/CMS timesharing, while there were no Multics or TSS service bureaus.

CP-67 didn't use the segmentation features of the 360/67; In fact, segmentation was not a feature of the original Blaauw proposal, but was derived from a 1966 *JACM* paper by Arden, Galler, O'Brien, and Westervelt of Michigan. Both TSS and MTS did use segmentation, but not as pervasively as Multics did on the GE 645. The 645's descriptors contained the access permission bits, whereas access in the System/360 was controlled by a per-page storage key. The privileged instruction SSK (Set Storage Key), not available on all models, could set the key. User jobs each had their own key and could only write pages with their own key; the 360/67 added fetch protection. This was simpler than the 645 segmentation mechanism but precluded sharing of memory between jobs.

CP/CMS was eventually adopted by IBM as a real product, **VM/370**, and spread to hundreds of sites. The 360/67 itself was withdrawn as a computer only on 6/15/77. IBM decommitted TSS/360 in 1968, un-decommitted it in 1969, and again killed off development in May 1971 (although in 1979, on an airplane, I sat next to a TSS user from Bell Labs Indian Hill). Honeywell starved Multics for years and finally killed it in 1985. MTS evolved and prospered, and was used until the RPI site shut down in 1999.

Dve Tuttle writes:

Your comment that "CP/CMS was eventually adopted by IBM as a real product, VM/370, and spread to hundreds of sites." is an understandable mistake, perhaps, but nonetheless a mistake. VM/370-CP was a complete redesign and new implementation, based on IBM product development project requirements, phase reviews, test criteria, etc. Because of budget and time constraints we carried over significant portions of CMS - more than we wanted to, in fact - but even there we had to "toe the line" of product quality. There were some notable internal challenges to overcome, but that's a different kind of history for another time, perhaps.

I had the impression that VM development was killed by IBM in 1983, but John Durkin informs me that many sites still run it, that IBM is still charging for it and holding technical conferences every year.

11/10/00: Got mail from Stan Willis, who tells me that IBM introduced the newest version of VM in October of 2000. It runs on their new z/Architecture. z/VM succeeds VM/ESA, the version for System/390 computers.

04/14/09: Got mail from Ken Leonard, who says:

I was a programmer at IBM "TSS Marketing and Support" group from about 1974 to 1978. For about two years before that, I was doing "driver" programming as an IBM employee at NASA Lewis Research Center (Cleveland, OH). The un-de-commit in 1969 was the result of what we called "Westchester I", a pitchforks-and-torches meeting of certain very large users of TSS/360 with IBM\_top\_management. There was another un-de-commit, "Westchester II" in maybe 1974.

Our job in the TSS M&S group was to build the "Bridge" to migrate TSS/360 to run on S/370-158 and -168 as -SP, -AP and -MP installations. The Bridge was later officially released as TSS/370 and it survived at least until 1980, maybe a couple years longer. It ran on S/370-135, -145, -165 and -168. The first S/370-168-MP to actually run\_as\_a multiprocessor was at GM Research and ran only TSS/370.

As of 1978, the TSS customers I can recall off the top of my head were Bell Labs Indian Hill, GM Research, KFA (in Germany), DKFZ (in Germany), Carnegie-Mellon University, NASA Lewis RC and NASA Ames RC. There were a total of 11 or 12 (I think) that could be publicly named and apparently a handful that (still) cannot be publicly named.


By the time I left IBM in '78, one of our several 370/1x8-MP installations could pump out 125% to 150% more terminal-driven engineering work than MVS/TSO could get out of any two 370/1x8 processors in any configuration.

The ultimate demise of TSS, and the "true S/370" or "FS" system architecture is an absolutely byzantine tale of IBM internal politics that I lack the background exposure to say more about.

---

 Pictures of University of Newcastle 360/67 from 1969.

Bob Creasy wrote  "The Origin of the VM/370 Time-sharing System" in the *IBM Journal of Research and Development*, Vol. 25, No. 5, September 1981. This article describes the roots of CP/CMS in CTSS.

Norm Hardy has an  informative page about IBM's virtual machine software.

---

*Thanks to Melinda Varian, Lynn Wheeler, and Dave Tuttle for CP/CMS reminiscences, Mike Alexander for information on MTS and the /67, John Durkin and Richard Willis for information on the current state of VM, Ken Leonard for info on TSS, and Jeff Ogden for info on MTS. And in memory of Bob Creasy.*

updated 12/15/10

Copyright (c) 1995-2010 by Tom Van Vleck