

# THE EVOLUTION OF AN INDUSTRY: ONE PERSON'S VIEWPOINT

## The History of CP/M

**BY GARY A. KILDALL**

Digital Research  
P.O. Box 579  
Pacific Grove, California 93950

1973. . .

I was sitting quietly at my desk when Masatoshi Shima hurried into my office at Intel and asked me to follow him to his laboratory down the hall. In the middle of his work bench, among the typical snaggle of jumpers, oscilloscopes and multi-meters, sat a binocular microscope with spider-leg probes, all of which were subjecting a minute piece of silicon to help-less investigation. I peered through the microscope at the enlarged regular patterns with particular interest. As a consultant, my job was to design and develop certain software tools for Intel. One was Interp/80, a program which simulated Intel's newly evolved 8080 microprocessor to be used by Intel customers on timesharing systems. As I searched for something recognizable, I hoped my simulation resembled the operation of Shima's first 8080 chip which had finally come to life.

My proposal to Intel had been simple: I would provide them with a language, called PL/M, to replace serious systems programming in assembly language. The compiler would first be written in FORTRAN for operation on timesharing computers and "cross compile" to the eight-bit processors. Next, we would write a PL/M compiler in PL/M and "bootstrap" from the timesharing computer to a resident compiler operating on Intel's new Intellec-8 development system. The first part was complete. PL/M cross compilers and Interp simulators were implemented for the now-best-forgotten 8008, as well as the 8080. Programs had been written and tested by Intel's software group, consisting of myself and two other people, and we were ready for the real machine. Things were going well: the resident compiler would be the next step.

Unfortunately, nearly all small computer systems in 1973 used paper tape as the backup storage device, with the ubiquitous model 33 Teletype serving as the nerve-shattering I/O device. It was readily apparent that resident development systems could not compete with timesharing services when considering throughput, resources, and services. Still the notion of a personal computer for software development interested everyone.

I became intrigued with a new device, called a floppy disk, which, though designed by IBM to replace punched cards, appeared to have much greater potential. The device was ideal: over 3,000 times the data rate of a Teletype, each \$7 diskette could randomly access the equivalent of 2000 feet of paper tape. Best of all, the drive was priced at a low \$500. Due to a slight problem of undercapitalization, I found this incredibly low price still a bit high. At that time, a smallish company called Shugart Associates was in operation a few miles up the road from Intel. Dave Scott, then marketing manager at Shugart Associates, donated one of their 10,000-hour test drives to the cause, complete with worn-out bearings and a bearing repair kit. It was only later, as I sat in my office at home, staring at the naked disk drive, that I realized I had no cabinet, no cables, no power supplies, no controller, and most distressing of all, no hardware design experience. To make matters worse, no controllers were commercially available, even if I could afford one.

After several abortive attempts at constructing an interface to my Intellec-8, it became readily apparent that my efforts would be better directed toward the software aspects.

Between projects I put together the first CP/M file system, designed to support a resident PL/M compiler. The timesharing version of PL/M, along with the Interp simulator, allowed me to develop and checkout the various file operations to the level of primitive disk I/O. A simulation is, after all, just a simulation, and the inability to make that 10,000 hour drive work for just one more hour was frustrating.

Shortly thereafter, in the fall of 1974, John Torode became interested in the project. I offered as much moral support as possible while John worked through the aberrations of the IBM standard to complete one of my aborted controllers. Our first controller was a beautiful rat's nest of wirewraps, boards and cables (well, at least it was beautiful to us!) which, by good fortune, often performed seeks, reads, and writes just as requested. For agonizing minutes, we loaded the CP/M machine code through the paper tape reader into the Intellec-8 memory. To our amazement, the disk system went through its initialization and printed the CP/M prompt at the Teletype.

Anyone who has brought -up CP/M on a homebuilt computer has felt this moment of elation. A myriad of connections are properly closed; bits are flying at lightning speeds over busses and through circuits and program logic, to produce a single prompt. In comparison to our paper tape devices, we had the power of a S/370 at our fingertips. A few nervous tests confirmed that all was working properly, so we retired for the evening to take on the simpler task of emptying a jug of not-so-good red wine while reconstructing battles, and speculating on the future of our new software tool.

In the months that followed, CP/M evolved rather slowly. Intel was experiencing enormous growth and all software development was halted while new management structures were instituted. Intel expressed no interest in CP/M, nor in continuing any resident compiler work. Nearly two years passed before Intel again took interest in resident software tools, with their introduction of the ISIS operating system and later, the resident PL/M -80 compiler.

Meanwhile, John Torode redesigned and refined our original controller and produced his first complete computer system, marketed under his company name, Digital Systems (which later became Digital Microsystems). The first commercial licensing of CP/M took place in 1975 with contracts between Digital Systems and Omron of America for use in their intelligent terminal, and with Lawrence Livermore Laboratories where CP/M was used to monitor programs in the Octopus network. Little attention was paid to CP/M for about a year. In my spare time, I worked to improve overall facilities, and added an editor, assembler, and debugger which were predecessors of the current ED, ASM, and DDT programs. By this time, CP/M had been adapted for four different controllers.

In 1976, Glenn Ewing approached me with a problem: Imsai, Incorporated, for whom Glenn consulted, had shipped a large number of disk subsystems with a promise that an operating system would follow. I was somewhat reluctant to

adapt CP/M to yet another controller, and thus the notion of a separated Basic I/O System (BIOS) evolved. In principle, the hardware dependent portions of CP/M were concentrated in the BIOS, thus allowing Glenn, or anyone else, to adapt CP/M to the Imsai equipment. Imsai was subsequently licensed to distribute CP/M version 1.3 which eventually evolved into an operating system called IMDOS.

By coincidence, Jim Warren and I were both consulting at Signetics Corporation during this time. Jim was then the editor of *DDJ*, and pushed for sale of CP/M to the general public. There was, at the time, a pervading paranoia among software vendors who felt that any and all loose software would be immediately "ripped-off" by this immoral group of computer junkies. Jim's faith in the industry, however, led me to introduce the CP/M 1.3 system for sale on the open market at \$70 per copy. In the months that followed, the nature of the computer hobbyist became apparent. In most cases he was, like myself, in the computer industry and merely wanted a personal computer for his own endeavors. CP/M gradually gained popularity through a "grassroots" effect and, to the amazement of the skeptics, the rip-off factor was practically nil. A new company called Digital Research was formed to support CP/M, develop new products, and provide administrative functions.

It's been nearly three years since CP/M's initial introduction, with several revisions and improvements. Although floppy disks maintain their popularity, CP/M 2.0 is now offered to manage larger capacity hard disks which are becoming more readily available. Customer needs and demands have also led to the recent introduction of MP/M, a CP/M compatible multiterminal multiprogramming system for more sophisticated applications.

More important, however, is that CP/M provides a manufacturer independent basis for an evolving software market. We all know that software is expensive to develop and support, with numbers quoted in the hundreds of thousands of dollars over the product lifetime. In a classical computer marketplace, these costs are amortized over a few installations, resulting in seemingly outrageous prices. Active CP/M users, however, number in the tens of thousands and can be reached through any number of popular magazines. Thus, marketeers reduce their prices substantially to interest a much larger customer base. Software is sold profitably as an independent commodity by a large number of responsible companies, and the benefits to the consumer are clear. Competition forces low prices and quality control, with selection among a wide variety of software products. Currently, CP/M compatible products range from word processing programs through business systems to a variety of language processors for BASIC, FORTRAN, COBOL, PASCAL, and others. All are priced in the \$100 to \$700 range. The future is, without doubt, optimistic for producers and customers alike.