

The Birth Of PC Software

In The Beginning

How did we program these little computers? The first 8-bit micro-computers available to amateurs had no application software or computer languages. The only way to program them was to insert instructions into memory locations and enable the processor to step from one memory location to the next and to execute the binary instruction held in the combination of memory location bits that comprised a digital word. These instructions were hard programmed into the processor and were different for each type of processor. They were called the processor's instruction set. This type of programming was called machine language programming.

"Programming" was done by setting switches on the computer's front panel to "1" or "0" and then pressing the "Insert" switch. The processor would then step to the next memory location and the process would be repeated. A very laborious process and if you made one error the whole program would not run.

Most people have a mental problem remembering strings of 8-bit digital numbers composed of only "1"s and "0"s, but they can remember combinations of alphabetic letters. These would be keyed into the computer from a keyboard. When a letter or number on the keyboard was pressed, the digital code for that character was sent to the computer and entered into memory. Once the entire sequence of code was keyed in, a previously loaded program called an assembler was run. The assembler converted the entered program into machine language that was understood by the processor. The program entered was called "source code" and the digital code resulting from running the assembler was called "objects code." The entire process was called "Assembly Language Programming."

Holding both the entered program and the assembler program took more memory and computer power than the little computers had. The hobbyists around this problem by using programs called "mini-assemblers" or "monitors" that were loaded-in by running punched paper-tape readers, or hard programmed into read-only memory chips. In addition to loading code, these programs allowed a user to step through memory and examine the contents one memory location at a time. They could also order the processor to load the program, run the program, or stop-on-error. Later audio cassette capability was added to the monitor programs and they became mini-operating systems. Typical of these were Mikbug by Motorola for the M6800, CPU; Software #1 by Processor Technology for the 8080 CPU and Zapple by Technical Design Labs for the Z-80 CPU.

Small memory capacity was not strictly a microcomputer problem; the first mini-computers had the same problem and inability to run the major languages used by large mainframe computers. The computer language FORTRAN (Formula Translator) had become the computer language for science and research while COBOL (Common Business Oriented Language) was the language used for business and commercial applications these were used on large mainframe computers.

When mini-computers grew in capacity they gained the ability to use mainframe languages and were adapted by colleges as teaching tools. However, in most colleges the use of computers was confined to data centers.

In the 1960's Dartmouth College installed a General Electric 225 Computer, one of the first interactive computer systems. This new machine allowed more than one student at a time to use the computer. Dartmouth extended computer use to all of the students, not just those in computer science programs. Students could use computer terminals located outside the data center to enter programs and operate the computer just as if it was located near their terminal.

However, even this interactive computing system had a major flaw. The students had to use FORTAIN, not a simple language to learn and use.

The Birth Of PC Software

Professors John Kemeny and Thomas Kurtz at Dartmouth College solved that problem by writing a simple computer language based upon FORTRAN. They called it BASIC (Beginner's All-Purpose Symbolic Instruction Code). The use of BASIC spread from Dartmouth to other colleges and to businesses. BASIC became one of major contributors to the rapid adoption of computers by business, scientific laboratories and schools. BASIC was extended to permit the development of all kinds of application software and by the time microcomputers appeared, it was even being taught in high schools.

Still the evolving micro-computers did not have the memory capacity to even run any of the dialects of BASIC used on the smallest mini-computers.

The original Dartmouth BASIC, like all programming languages of that time was a compiled language. The source code was entered into the computer memory and then the BASIC compiler was run. This translated the BASIC into machine code which was understood by the computer.

A simpler system needed and an interpretive BASIC was developed. In this system the BASIC source code was entered into the computer and compiled into machine code on a line-by-line basis. This permitted immediate error detection and required less memory capacity. Interpretive BASIC became the preferred language for small mini-computers and eventually for micro-computers.

The break-through for the new micro-computers came from Bob Albrecht and Dennis Allison, two members of The People's Computer Company a California group who published a newsletter and provided mini-computers to students and hobbyists on a per-use rental basis. Albrecht asked his friend Allison to write a stripped down version of BASIC that would run on the Altair, the first hobbyist's micro. The Altair was only sold with a few bytes of memory but the purchaser could buy and add 4 K memory boards. Allison did not have the time to undertake the project himself, but he did write a specification for "Tiny" BASIC. This tiny version would lack the ability to do floating point math, or alpha-numeric strings and would be limited in the number of variables it could work with at one time. Allison published the specification in People's Computer newsletter and invited the subscribers to try to write it. Late in 1975, two Texas hobbyists, Dick Whipple and John Arnold sent in the code for a Tiny BASIC interpreter that could be run on the Altair with a 4K memory card. Bob Albrecht published the BASIC listing in the December 1975 issue of People's Computer Company newsletter.

The result was amazing, versions of Tiny BASIC flooded in from all over the country, each variation getting better and adapted to more microprocessors.

Soon computers based upon the Motorola M6800 and 6502 processors could run Tiny Basic. Most of these versions were given freely to the public domain, but a few versions were copyright and sold for profit

About the same time as Tiny BASIC was being developed, Ed Roberts, owner of MITS, the company that made the Altair, offered to buy a version of BASIC that would run on his Altair computer. He specified that it include most of the extended features of Dartmouth BASIC not available on Tiny BASIC. Two Harvard students Bill Gates and Paul Allen answered the challenge by writing a program that simulated the Altair on Harvard's mainframe computer. They then proceeded to write a version of BASIC that filled Roberts's requirements and sold it to MITS on a per-copy-royalty basis. Leaving Harvard, the partners moved to New Mexico where MITS was located and started their new software business, called Microsoft.

The contract with Ed Roberts looked like a good deal to the Microsoft partners, but there was an almost fatal flaw in it. MITS priced the BASIC software at \$150 per copy for the 4K BASIC, a price that was considered exorbitant by the computer hobbyists. Ed Roberts really did not expect the users to pay this price, but used it as a ploy to sell memory boards. If you purchased a 4K MITS memory board

The Birth Of PC Software

you could get the BASIC for only \$75. If you bought both the memory board and an I/O Board it only cost \$60.

Prices for the Extended BASIC were also reduced when the user purchased additional memory boards. The original Altair came with only enough memory to operate the front panel therefore adding memory boards was a necessity, not an option if you wanted to use the computer at all. Also, the Altair did not come with any Input/Output capability. You had to add a serial I/O board if you wanted to connect the computer to a printer, a punched paper tape reader, or a cassette interface. . To connect a keyboard or a parallel printer you also had to add a parallel interface.

The users soon found that the MITS memory boards did not work very well. They used dynamic memory chips that had to be electrically refreshed or they would "forget" what they were supposed to "remember".

Memory boards made by other manufacturers such as Processor Technology Inc. used static memory chips that did not require refreshing and retained their memory as long as they were powered by the computer.

The owners of Altair or Imsai computers (which were almost identical) bought static memory boards from many companies.

They also resented MITS' tie-in tactics. These hobbyists invented "multi-user BASIC". A group of users would chip-in and buy one copy for \$150 and then make copies for each member of the group. In addition, Altair BASIC had a "trick" so that it would not work on other 8080 computers. The hobbyists found the "Inverted I/O Bit" and fixed it so that the BASIC would run on any other 8080 or Z-80 machine. They called their stolen product "New Jersey BASIC" and freely distributed it all over the country.

As a result, Microsoft did not get the income they expected from their BASIC and Bill Gates became enraged. He wrote a letter to the editor of BYTE Magazine in which he called the computer hobbyists cheats and thieves, who robbed him of his just returns on his software. He further declared that he would write no more software for microcomputers. Can you imagine what this industry would have become without Microsoft? However reason prevailed and Gates found a much more practical way to exploit his and Allen's talents. Microsoft sued MITS to break their contract and won in court. By that time computer users were starting to use floppy disks and required advanced copies of BASIC. Since they were used to Microsoft BASIC that is the one they bought. So the theft of Altair BASIC assured it of the widest distribution and started Microsoft's domination of the language. However Gates never forgot the lesson of Altair and never entered into an exclusive contract again.

Other versions of BASIC were developed for use with different computers. Apple BASIC for the 6502-based Apple II was developed by Steve Wozniack and TRS-80 BASIC was developed for the Radio Shack TRS-80 Computers. There were soon dialects of BASIC for every microcomputer on the market South West Technical Products amusingly priced their BASIC starting at \$4 to lure hobbyists to their line of M6800 computers.

All of the versions of BASIC were derived from the original Dartmouth BASIC and a user who knew any version could learn to program in any other version with a minimum of study.

The BASIC programs were partially hard-coded into read-only-memory chips built into the computer and partially loaded by audio cassettes. After the BASIC language was running, the software application could be loaded from another cassette. Any data resulting from the programs could only be saving into another cassette, or it would be lost once the computer was turned off... This was not a

The Birth Of PC Software

satisfactory situation and limited the practical use of the personal computers. Another solution had to be found.

Floppy Disks

The floppy disk drive was invented by IBM as a device for storage and loading of short programs called "boot-straps" used to initialize mainframe computers. The disks were circular pieces of recording film sandwiched between two pieces of cardboard. The disk assembly was inserted into the drive and two revolving hubs clamped the film disk through openings in the cardboard and spun it. The read/write heads contacted the film through slits in the cover and moved rapidly across the film under control of a servo positioner. Using the floppy disks required a program which produced a pattern of concentric tracks which were further divided into sectors.

There are various methods of positioning and controlling the movement of the read/write heads, the most popular is known as IBM-formatting. This uses a single hole punched through the disk to allow a light beam to shine through which causes a photoelectric cell to develop a pulse called the index pulse. The logic circuits used this index to determine the position of the heads with respect to the tracks and sectors.

The first floppy disks were 8-inches in diameter and single-sided. They held about 100K bytes of data. Later double-sided disks were developed and as technology advanced 5 1/4-inch and 3-inch disks were invented that help almost 10 times as much data, The floppy disk became very popular for use with mini-computers as an inexpensive and convenient method of storing programs and data.

When microcomputers were invented, it became a very important task to adopt them for use with floppy disks.

The Birth of the CP/M Operating System

Intel hired Gary Kildall to develop a version of the PL/M language to be used as a systems development language on their Intellec-8 Development System, At the time, the CPU chips themselves barely existed and Intel was just then starting to design a computer system that used the 8080. The plan was for Gary to use the 8080 emulator Intel had running on their big PDP-10 minicomputer, but he preferred to work directly on the 8080 itself, in part because by working on his own machine at home he could avoid the 50 mile drive to Intel to work every day. The only 8080-based computer Intel had available was called "Intellec-8", but it didn't have any software or disk storage attached to it. So Kildall obtained a used test floppy drive from Shugart Associates, attached it to the Intellec-8 with a controller designed by his friend John Torode, and wrote a primitive operating system for it, He called CP/M. for "Control Program, Microcomputer."

CP/M was developed on Intel's 8080 emulator under DEC's TOPS-10 operating system, so naturally many parts of CP/M were inspired by it, including the eight character filenames with a three-character extension that every MS-DOS/Windows 3.X user still lives with today. By the end of 1975, Kildall at last had CP/M version 1.0 ready and had started on PL/M, but by then Intel was no longer interested in the systems development language. Gary offered CP/M to them, but the company saw no potential in it and declined to market it, allowing Kildall to keep the system.

The IMSAI, company which built a very successful 8080-based system, had just started marketing their own floppy disk system, and desperately needed an operating system. . They heard about Kildall and CP/M, and contacted him to license it. One fly in the ointment was that CP/M was written to only use "standard" IBM-compatible soft-sectored floppy disks. To avoid IBM license fees, IMSAI used a different, incompatible type floppy, called "hard sectored... Gary Kildall solved the problem by changing the operating system to make it more flexible. What he did was to separate the parts of CP/M version 1 that addressed the specific format of the diskettes, and placed them in a separate module he called the BIOS, for Basic Input/Output System. Thus, the system could easily be adapted

The Birth Of PC Software

to new hardware without having to rewrite or even revise the complex heart of the software, this modular treatment led to the establishment of CP/M as the first cross-platform operating system.

Other new hardware startups need not write their own operating systems, they could buy CP/M and adapt it themselves to their unique hardware. Further, because CP/M operated the same way on every 8080-based, or Z80 computer, other software developers were also relieved of the necessity of adapting their software products to each new machine -- they could write to be compatible with CP/M, and modify the BIOS to work with different input /output systems. One company Lifeboat Associates made a big business out of providing application software versions that ran under CP/M in almost any type of 8080 or Z-80 computer.

It was CP/M's adaptability that gave it appeal and launched it on the road to success, but any operating system that had that characteristic might have succeeded in a similar way, given the right timing and some luck. But CP/M was more than just lucky -- it was good. It packed a surprising amount of power in a tiny package, and did so in a simple, clean logical way. Many of its critics bemoaned its sometimes cryptic commands (rightly) and also its lack of powerful features. But it must be remembered that CP/M was designed in an age when it was a rare, high-end computer owner that could afford the thousands of dollars it took to fill up the whole 64K of the 8080's address space. The entire operating system took only 8K of the computer's memory, and would run in a mere 16K of total memory with room left over for any of its system development utilities to run. More features would have swelled the system to the point where decently featured applications would have had no room to execute.

It was the applications that moved this operating system out of the realm of the computer enthusiasts and into the hands of "real users" (people who don't care if their computers are powered by hamsters, so long as they run their necessary applications reliably). The first real "killer app" for CP/M was probably WordStar, an excellent word processing program that became very widely used. The first microcomputer database application, dBASE II. Was very popular many. Visi Calc, the first spreadsheet, originally written for Apple II was also converted to CP/M, many other applications and utilities eventually made CP/M a business tool for a wide range of ordinary people.

By 1978, CP/M 2.2 had been ported to nearly every 8080 and Z80 based microcomputer built. In the end, more than 500,000 computers would be sold with CP/M as their operating system. By 1980 it completely dominated the 8-bit microcomputer world, except for Apple computers and Radio Shack computers. Digital Research came out with the CP/M 3.X family which became their primary product before the advent of 16-bit personal computers.

Death of CP/M

By 1981, the apogee of the CP/M orbit, a new generation of Intel microprocessors was on the horizon -- the 8086 and 8088 16-bit chips, which could address an incredible 1 megabyte of memory. This seemed at the time more than anyone could ever figure out a use for, so Digital Research focused much of their attention on producing CP/M 3.0 for the dominant 8080/Z80 platform. There were plans of course to port CP/M to the new 16-bit chips with a version called CP/M-86, but it was not a priority at the time.

But IBM planned on entering the microcomputer market in a serious way in 1981, and the computer giant had its own priorities and a great deal of money and marketing power to back them up. IBM chose the new 8088 chip as the heart of their new PC, and of course they needed an operating system to run on it.

Instead of writing their own, as they generally did with a new machine, time and the demands of the market led them to Digital Research's door to license the ubiquitous CP/M for the new IBM-PC.

The Birth Of PC Software

As the now-famous story goes, Gary Kildall was not there to open that door when IBM came calling -- an avid amateur pilot; he was flying his private plane on a business trip to the Bay area. His wife and business partner, confronted with IBM's imposing code of secrecy and non-disclosure agreements, refused to sign even enough for talks to begin. Rebuffed by what they considered arrogance, the IBMers went elsewhere.

According to a Computer Shopper interview with Kildall this story is not true. He did speak with IBM about doing a 16-bit version of CP/M for them, but because of previous commitments he refused to meet their time schedule for release of the IBM PC.

IBM returned to Microsoft and told them of Kendall's time problems and asked if Bill Gates could help. There was a small company in Seattle that had developed an operating system for the 16-bit, Intel 8086. The 8088 chip was a sub set of the 8086 family and could run on the same software with minor alterations. Microsoft bought the system from Seattle Computer and modified it for the IBM PC. They called it MSDOS and licensed it to IBM on a non-exclusive basis.

Kendall's company Digital Research finally came up with a 16-bit version of CPM which they called CP/M-86 and IBM did offer it as an alternate operating system on the IBM PC. However the price for the MSDOS system for the IBM PC was \$60 while the price for the CP/M-86 System was set at \$260 under that price structure very few users selected CP/M-86 as their operating system.

CP/M continued to improve as an 8-bit operating system as long as 8-bit computers remained on the market, but eventually it passed from popularity as 8-bit machines went off the market.