Software Gems: The Computer History Museum Historical Source Code Series

By the time personal computers based on microprocessors began to emerge in the mid-1970s, programmers had been writing operating systems – the software that manages the computer hardware and provides commonly used services for application programs – for about twenty years. Big mainframe computers had operating systems that were huge and complicated, created from hundreds of thousands of lines of code. But other operating systems, designed to fit in the small memory of minicomputers, were tiny. That was the kind that the PCs could use.

Computer Scientist Gary Kildall created just such an operating system in 1974 for a small computer called the "Intellec-8" that Intel had designed to showcase their new microprocessors. Called "CP/M", it was unlike most other operating systems for small computers because it was written in PL/M, a portable higher-level language that he had designed earlier, rather than in the assembly-language of a particular computer. That meant that CP/M could be ported to run on many different personal computers. And if the applications were written in PL/M, they could be ported as well.

(http://www.computerhistory.org/atchm/wp-content/uploads/2014/08/Kildall_1977.jpg)

Gary Kildall at the first West Coast Computer Faire in the San Francisco Civic Auditorium in 1977
[CHM Object ID: 500004174 © Tom Munnecke/Hulton Archive/Getty Images]

The early availability of CP/M, combined with its portability, made it a runaway success. Kildall started a company called Digital Research, Inc. (DRI) in Pacific Grove, California to develop and market CP/M, and for years it was the dominant operating system for personal microcomputers. It was eventually overtaken by Microsoft's MS-DOS for the IBM PC in the 1980s, but before then it was responsible for a vibrant ecosystem of application programs that contributed to the rapid proliferation of the early personal computers.

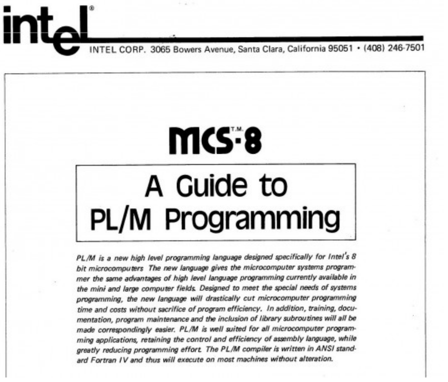
To mark the 40th anniversary of the prototype demonstration in Kildall's backyard tool shed in Pacific Grove in the fall of 1974, the Computer History Museum is pleased to make available, for non-commercial use, the source code of several of the early releases of CP/M.

CP/M's creator

In the early 1970s, Gary Kildall was an instructor in computer science at the Naval Postgraduate School in Monterey, California. He was also a part-time contractor for Intel, and he began using an Intel development system to create software for that young company's new microprocessors.

In 1974 Kildall wrote a program that allowed microprocessor designers to replace slow paper tape storage with the new fast 8" floppy disks that were becoming commercially available. Initially called Control Program/Monitor, later renamed Control Program for Microcomputers ("CP/M"), his program proved to have value far beyond fulfilling the requirements of his part-time consulting gig for Intel. When it was fully configured as a commercial operating system in 1976, CP/M elevated a hobbyist activity into a mainstream personal and small business computer tool, and in doing so it laid an important foundation for the personal computer revolution.

Early versions of CP/M were written entirely in PL/M ("Programming Language for Microcomputers"), a high-level systems programming language he developed for Intel in 1972 for its microprocessor development systems [1]. Primarily a subset of the mainframe computer language PL/I, it also incorporated ideas from ALGOL and XPL, and included an integrated macro processor. In an article presented at the 1975 National Computer Conference [2], Kildall called CP/M "modest in structure and scope", but "allowing access to all machine functions, without becoming completely dependent upon a particular CPU organization."


INTEL CORP. 3065 Bowers Avenue, Santa Clara, California 95051 • (408) 246-7501

MCS-8 T.M.

A Guide to PL/M Programming

PL/M is a new high level programming language designed specifically for Intel's 8 bit microcomputers The new language gives the microcomputer systems programmer the same advantages of high level language programming currently available in the mini and large computer fields. Designed to meet the special needs of systems programming, the new language will drastically cut microcomputer programming time and costs without sacrifice of program efficiency. In addition, training, documentation, program maintenance and the inclusion of library subroutines will all be made correspondingly easier. PL/M is well suited for all microcomputer programming applications, retaining the control and efficiency of assembly language, while greatly reducing programming effort. The PL/M compiler is written in ANSI standard Fortran IV and thus will execute on most machines without alteration.

(http://www.computerhistory.o
/atchm/wp-content/uploads
/2014/09
/PLM_title_page_top.jpg)

"A Guide to PL/M Programming," Intel Corporation, September 1973

The story of CP/M

Starting in the late-1970s, one of the premier publications for the incipient personal computer software industry was the unusually-named *Dr. Dobb's Journal of Computer Calisthenics & Orthodontia* (http://www.drdobbs.com%20). In the January 1980 issue, Gary Kildall published a personal memoir [3] about the creation and growth of CP/M. These sections quote liberally from that article.

The first generation of software for microprocessor-based systems was developed using mainframe timesharing systems. Kildall dreamt of the day that the microprocessors themselves could be used to develop their own software.

> *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 … 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.*

He realized that a key requirement was better storage, and that there was something appropriate on the horizon.

> *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. … (http://www.computerhistory.org/atchm/wp-content/uploads/2014/09/8-inch-floppy.jpg) 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…*

> *At that time, a smallish company called Shugart Associates … a few miles up the road from Intel … 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.*

Not one to let the lack of hardware be an impediment, Kildall proceeded to develop CP/M using a timesharing simulator.

> *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.*

The first boot

Assistance to build the hardware interface to the floppy disk did materialize, however, allowing CP/M to come to life.

> *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 an [IBM] 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.*
> *Shortly thereafter, in the fall of 1974, John Torode [a fellow University of Washington alumnus] 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 wire wraps, 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.*

Commercialization

The proliferation of CP/M depending on it being adopted by PC manufacturers. Kildall's hardware savior was the first adopter.

> *John Torode redesigned and refined our original controller and produced his first complete 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 their Octopus timesharing network.*

It quickly became clear that in order to make adaptation to other computers easy, the hardware-dependent portions had to be segregated into a separate, easily-changed part.

> *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.*

It was a successful strategy. DRI licensed CP/M to IMSAI for a one-time fee of $25,000, and others soon followed.

(http://www.computerhistory.org/atchm/wp-content/uploads/2014/09/Imsai_DSC1239as.jpg)

IMSAI System, CP/M User Manual (1977)
CHM Catalog # 102679413

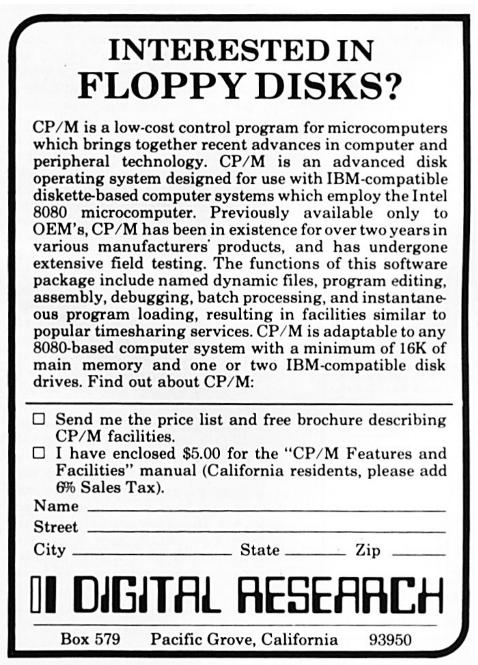
Creating a market for independent software vendors

Before CP/M, computer manufacturers designed their operating systems to work only with their own hardware and peripheral equipment. An IBM operating system would only work with IBM computers; a Burroughs operating system with Burroughs computers, etc. And applications had to be written for the specific operating system of each computer. Such "closed systems" made it difficult or impossible to mix and match the best pieces of equipment and software applications programs from different manufacturers.

Kildall's addition of the BIOS allowed not just IMSAI systems but all Intel 8080 and compatible microprocessor-based computers from other manufacturers to run same the operating system on any new hardware with trivial modifications that could be accomplished by a programmer in a few hours. This capability stimulated the rise of an independent software industry by expanding the potential size of the market for each product. A single program could run without modification on computers from multiple suppliers.

Growth

Together with his wife Dorothy McEwen, Kildall incorporated Digital Research, Inc. in Pacific Grove in 1976 to market CP/M version 1.3, which incorporated the BIOS code. Through mail order advertisements in *Byte* magazine and other publications, the word got out that there was a manufacturer-independent operating system available.



(http://www.computerhistory.org/atchm/wp-content/uploads/2014/09/DRI-advert-1976.jpg)

DRI advertisement for CP/M, *Byte* magazine, December 1976

There was little competition. CP/M was first sold by mail-order to hobbyists for $75 [4]. It later became a standard OS offered by many pioneering personal computer manufacturers, such as Altair, Amstrad, Kaypro, and Osborne. By 1980, about 200 different computer models were running CP/M.

CP/M was initially limited to personal computers based on the Intel 8080 microprocessor and its derivatives. But CP/M and the application programs written for it provided such value that it was reasonable to do a "brain transplant" and install an additional microprocessor into computers that didn't have an 8080 or compatible chip. Microsoft, for example, built a Zilog Z80-based "SoftCard" for the otherwise incompatible Apple II because they were a supplier of language compilers and interpreters for CP/M. It also enabled the Apple II to run popular CP/M applications such as dBASE and WordStar.

DRI released many versions of CP/M, of which the largest seller was version 2.2 in 1979. They also developed multiuser (MP/M), multi-tasking (Concurrent CP/M) and networking (CP/Net) versions. By 1982, the company enjoyed annualized sales in excess of $20 million, employed more than 200 people, and claimed that "more than a million people" were using systems based on its technology.

Losing the Edge

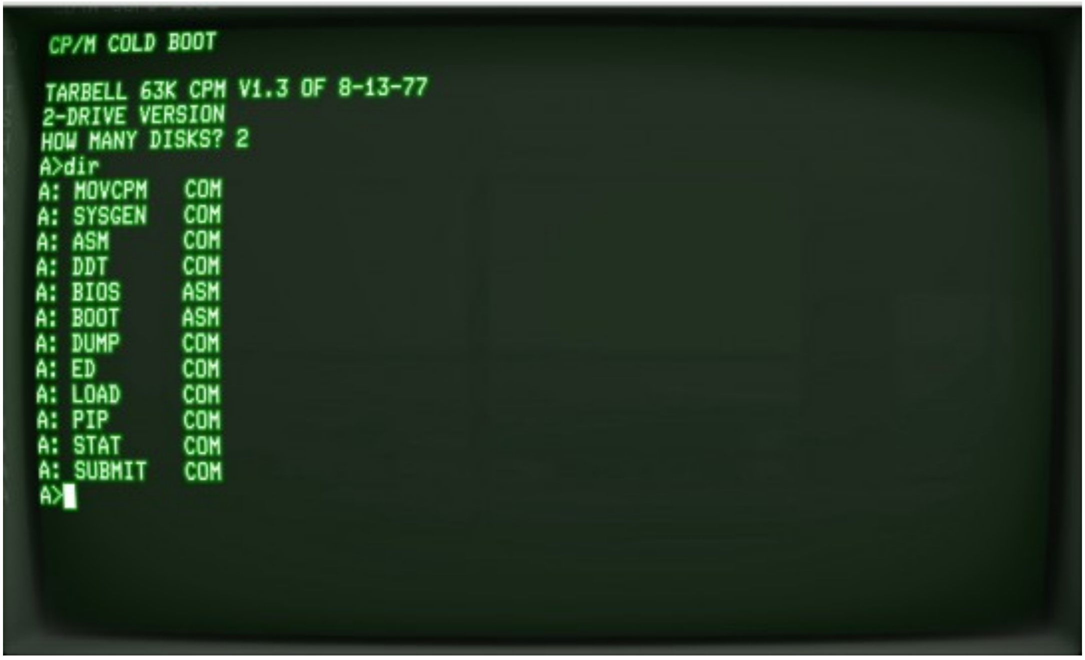
In 1980, Tim Paterson at Seattle Computer Products used DRI's 1976 *CP/M Interface Guide* and other information to guide the development of QDOS, a new operating system with the look and feel of CP/M but different internals, which he was writing for the new Intel 8086 16-bit microprocessor. Microsoft later acquired QDOS and licensed it to IBM as PC DOS, and to other PC-compatible manufacturers as MS-DOS.

Because of the popularity of the IBM PC platform, by the mid-1980s' CP/M had lost its market dominance to MS-DOS. For more information about the MS-DOS story, see another in the Computer History Museum Historical Source Code series: "Microsoft MS-DOS early source code (http://www.computerhistory.org/atchm/microsoft-ms-dos-early-source-code/)"[5].

What CP/M was, and what it wasn't

CP/M and the other operating systems for small personal computers, including MS-DOS (http://www.computerhistory.org/atchm/microsoft-ms-dos-early-source-code/) [5] and Apple DOS (http://www.computerhistory.org/atchm/apple-ii-dos-source-code/) [6], were very different from the operating systems we know today, or the ones that were on the large mainframes in the 1970s. A PC-based operating system was basically just a file management system, a rudimentary text command processor, and a way to load application programs. CP/M came with five built-in commands (for saving, erasing, renaming, listing, and typing files), and nine additional "transient" commands. There was no graphical user interface, no loadable device drivers for new peripherals, and no memory management system.

The "Introduction to CP/M Features and Facilities" manual initially published by DRI in 1976 [7] describes its key features:



(http://www.computerhistory.org/atchm/wp-content/uploads/2014/08/cpm13_screenshot.jpg)

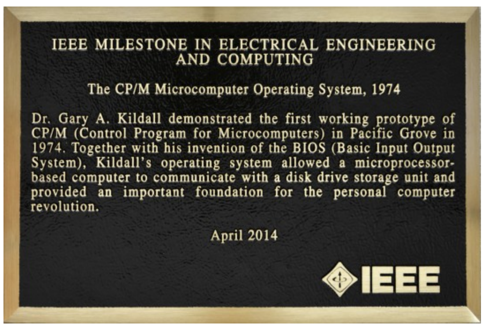
Screen shot of CP/M version 1.3; courtesy of Udo Munk

> *CP/M also supports a powerful context editor, Intel compatible assembler, and debugger subsystems. Optional software includes a powerful Intel-compatible macro assembler, symbolic debugger, along with various high-level languages. When coupled with CP/M's console command processor, the resulting facilities equal or excel similar large computer facilities.*
>
> *CP/M is logically divided into several distinct parts:*
> *BIOS Basic I/O System (hardware dependent)*
> *BDOS Basic Disk Operating System*
> *CCP Console Command Processor*
> *TPA ansient Program Area*
>
> *The BIOS provides the primitive operations necessary to interface standard peripherals (teletype, CRT, Paper Tape Reader/Punch, and user-defined peripherals), and can be tailored by the user for any particular hardware environment by "patching" this portion of CP/M. The BDOS provides disk management by controlling one or more disk drives containing independent file directories. The BDOS implements disk allocation strategies which provide fully dynamic file construction while minimizing head movement across the disk during access. Any particular file may contain any number of records, not exceeding the size of any single disk (240 records of 128 bytes each). In a standard CP/M system, each disk can contain up to 64 distinct files…*
>
> *The CCP provides symbolic interface between the user's console and the remainder of the CP/M system. The CCP reads the console device and processes commands which include listing the file directory, printing the contents of files, and controlling the operation of transient programs, such as assemblers, editors, and debuggers. …*
>
> *The last segment of CP/M is the area called the Transient Program Area (TPA). The TPA holds programs which are loaded from the disk under command of the CCP. During program editing, for example, the TPA holds the CP/M text editor machine code and data areas. Similarly, programs created under CP/M can be checked out by loading and executing these programs in the TPA.*
>
> *CP/M is a monitor control program for microcomputer system development which uses IBM-compatible flexible disks for back-up storage. Using a computer mainframe [sic] based on Intel's 8080 microcomputer, CP/M provides a general environment for program construction, storage, and editing, along with assembly and program check-out facilities… The CP/M monitor provides rapid access to programs through a comprehensive file management package. The file subsystem supports a named file structure, allowing dynamic allocation of file space as well as sequential and random file access. Using this file system, a large number of distinct programs can be stored in both source and machine-executable form.*

The original implementation of CP/M used only Kildall's high-level PL/M language. The later versions of CP/M that had a separate BIOS used a mix of assembly-language and PL/M.

Recognition for CP/M

Although CP/M disappeared as a mainstream operating system, echos of it still survive. As recently as early 2014, whenever the IBM OS 4690 operating system booted up in point-of-sale terminals around the world, it displayed the notice "Copyright (C) 1976 Digital Research".



(http://www.computerhistory.org/atchm/wp-content/uploads/2014/08/cpm_plaque.jpg)

CP/M was indisputably an important part of the early personal computer revolution. In April 2014 the Institute of Electrical and Electronic Engineers recognized the development of CP/M as an IEEE Milestone in Electrical Engineering and Computing by installing a bronze plaque outside the former DRI headquarters at 801 Lighthouse Avenue, Pacific Grove.

The IEEE Milestone “The CP/M Microcomputer Operating System, 1974” details (http://www.ieeeghn.org/wiki/index.php/Milestones:The_CP/M_Microcomputer_Operating_System,_1974).
Video of the dedication (https://www.youtube.com/watch?v=HO6IPpL0y8g).
A Computer History Museum blog about the anniversary (http://www.computerhistory.org/atchm/gary-kildall-40th-anniversary-of-the-birth-of-the-pc-operating-system/).

The source code

We are releasing scanned printer listings and/or machine-readable source code for four early versions of CP/M dating from 1975 to 1979. Some versions are incomplete, but please don't ask us for what is missing because we are releasing everything we have.

This material is provided for non-commercial use only. All the files are combined into one ZIP file with four directories representing the four versions.

1. Version 1.1(?), from 1975

This is the earliest source code for CP/M we have, dating from before there were official version numbers. It was used at Lawrence Livermore Labs for their Octopus [8] network system. Another variant was produced for terminal-maker Omron.

CP/M was originally written in PL/M and compiled with Intel's FORTRAN-based cross-development tools running on a timeshared mainframe computer. The object code was in an Intel hex tape format that couldn't be linked with other objects, which meant that the BIOS and the BDOS had to be in a single source code module. This version also had no fixed entry point for operating system calls; application programs had to call the BDOS entry point using the “magic address” of where it compiled to.

Built-in commands had a long form (“DIRECT” for directory listing) and a short form (“DIR”). But they were primitive. If you typed “DIR”, it showed nothing; you needed to type “DIR *.*” to show all files. Udo Munk, a dedicated CP/M enthusiast, observed, “One also can't do DIR B:*:*. One first has to switch to B: and then run commands there. This shows very well how this all was improved over the time. It must have been a very dynamic process of constantly trying and moving forward.”

For this early version we have original PL/M source code files for the BIOS and BDOS, for the Console Command Processor, and for the Transient Command Loader. These were recovered by Udo Munk from the CP/M Users Group disk #5. Other files in this directory are later reconstructions by Udo; see the readme.txt file for more details.

2. Version 1.3, from 1976

In response to the need to supply another version of CP/M to microcomputer maker IMSAI, Kildall realized that he needed to start making it easier to modify for different computers. In version 1.3 the BIOS and BDOS were separated and separately compiled. The BIOS was rewritten in assembly-language so that it has easy access to hardware I/O registers. There was now a permanent entry point at location 0005 for BDOS calls. This is, in essence, the first public release of CP/M.

The authentic source code we have for this version, courtesy of Steven A. Ness, is a scan of a printer listing that was released by Digital Research in 1976. Steve describes how he acquired the manual from Gary Kildall:

> “
>
> *On my first visits, I used machines in Gary's garage/toolshed (at 781 Bayview Avenue). Very soon after, Gary and Dorothy moved Digital Research to the top floor of 716 Lighthouse Avenue, and I worked there until I got my own machine, probably in May or June of 1977. I typically spent several days in PG [Pacific Grove] before returning to Palo Alto, so I often slept in the tiny overlook at the top of the building. Gary Kildall gave me a hard copy of CP/M 1.3 source during that period (i.e., between 2/77 and 6/77). I remember a shelf of identical copies in the office; each customer who bought CP/M source received one, in addition to machine-readable source on floppies." [9]*
> *Bob Swartz, founder of Mark Williams Company, hired me in February 1977 to write a BASIC interpreter (eventually called XYBASIC) for 8080s. At that time, it was possible to get S-100 bus 8080 hardware, but there was a waiting list for 8" floppy drives. Bob already had a CP/M machine and knew Gary Kildall and John Torode, and he arranged for me to use Gary's hardware initially. I commuted from my apartment in Palo Alto to Pacific Grove for several months.*

There is a gap of several lines on page 78 of the listing. The missing code has been reconstructed by Larry Greene from a later version and is included as a text file in the directory.

Update on August 2, 2016: Bob Zeidman has created text files from the image scans of the code, and they are in the Zeidman_code subdirectory. Thanks, Bob!

For version 1.3 we also have an amazing 48-page reverse-engineered source code listing that we think was written by Wayne Gramlich and modified by Leo Kenen. Even though CP/M was partially written in PL/M, this carefully constructed hand-annotated disassembly of the object code was made without access to the original source code. The author analyzed the code and created his own original comments. In some places he noted that he wasn't sure whether the object code was compiled from PL/M or assembled from 8080 assembler code! The printed listing is dated 1980, but the disassembly was probably done earlier. If anyone knows more about this tour-de-force, please let us know.

Manuals related to this release are:

- An Introduction to CP/M Features and Facilities, Digital Research, 1976 (http://bitsavers.org/pdf/digitalResearch/cpm/1.3/CPM_1.3_An_Introduction_to_CPM_Features_and_Facilities_1976.pdf)
- CP/M INTERFACE GUIDE, Digital Research, 1976 (http://bitsavers.org/pdf/imsai/dos-a/05_CPM_Interface_Guide_1976.pdf)
- CP/M (DOS-A) FLOPPY DISK OPERATING SYSTEM MANUAL, IMSAI, 1977 (http://bitsavers.org/pdf/imsai/dos-a/00_IMSAI_Floppy_Disk_Operating_System_DOS-A_Mar77.pdf)
- IMSAI CP/M SYSTEM USER'S GUIDE Version 1.31, Rev. 2, IMSAI, 1977 (http://bitsavers.org/pdf/imsai/dos-a/01_81-0000020rE_IMSAI_CPM_System_Users_Guide_Version_1.31_Rev_2_Mar77.pdf)
- CP/M System Alteration Guide for IMSAI CP/M Version 1.31, IMSAI, 1977 (http://bitsavers.org/pdf/imsai/dos-a/06_81-0000056rC_CPM_System_Alteration_Guide_for_IMSAI_CPM_Version_1.31_1977.pdf)

3. Version 1.4, from 1978

In this version the BDOS has been divided into two parts: a main PL/M body, and a "BDOS Interface Module" written in assembly language.

This is the first version that didn't require a mainframe computer to build. Compilation and assembly was done on Intel development systems running ISIS, which had a PL/M compiler and a macro assembler.

For this version we only have the BDOS source code, in its two parts.

Manuals related to this release are:

- AN INTRODUCTION TO CP/M FEATURES AND FACILITIES, Digital Research, 1978 (http://bitsavers.org/pdf/digitalResearch/cpm/1.4/CPM_1.4_Introduction_to_Features_and_Facilities_Feb78.pdf)
- CP/M INTERFACE GUIDE, Digital Research, 1978 (http://bitsavers.org/pdf/digitalResearch/cpm/1.4/CPM_1.4_Interface_Guide_1978.pdf)
- CP/M SYSTEM ALTERATION GUIDE, Digital Research, 1978 (http://bitsavers.org/pdf/digitalResearch/cpm/1.4/CPM_1.4_System_Alteration_Guide_Jan78.pdf)
- CP/M DYNAMIC DEBUGGING TOOL (DDT) USER'S GUIDE, Digital Research, 1978 (http://bitsavers.org/pdf/digitalResearch/cpm/1.4/CPM_1.4_Dynamic_Debugging_Tool_Users_Guide_1978.pdf)
- CP/M ASSEMBLER (ASM) USER'S GUIDE, Digital Research, 1978 (http://bitsavers.org/pdf/digitalResearch/cpm/1.4/CPM_1.4_Assembler_Users_Guide_1978.pdf)
- ED: A CONTEXT EDITOR FOR THE CP/M DISK SYSTEM USER'S MANUAL, Digital Research, 1978 (http://bitsavers.org/pdf/digitalResearch/cpm/1.4/CPM_1.4_ED_Users_Manual_1978.pdf)

4. Version 2.0, from 1979

Version 2.0 considerably expanded and generalized access to disks. There could be up to 16 logical drives with 8 MB each, as specified by a "disk parameter block". Files could be spread over up to 512 extents. There was also a new concept of users with "user numbers", whose files are kept separated.

The source code of this version, thanks to former Digital Research VP of Engineering Tom Rolander, is a scan of the source code listing released by the company in 1979.

We also have machine-readable source code for most of the files. We suspect they have been retyped from the listing, so they may have errors, but spot checks are encouraging about their accuracy.

Manuals related to this release are:

- CP/M 2.0 USER'S GUIDE FOR CP/M 1.4 OWNERS, Digital Research, 1979 (http://bitsavers.org/pdf/digitalResearch/cpm/2.0/CPM_2_0_Users_Guide_for_CPM_1_4_Users_1979.pdf)
- CP/M 2.0 INTERFACE GUIDE, Digital Research, 1979 (http://bitsavers.org/pdf/digitalResearch/cpm/2.0/CPM_2_0_Interface_Guide_1979.pdf)
- CP/M 2.0 ALTERATION GUIDE, Digital Research, 1979 (http://bitsavers.org/pdf/digitalResearch/cpm/2.0/CPM_2_0_System_Alteration_Guide_1979.pdf)

Acknowledgements

Thanks to Steven Ness, Tom Rolander, and Udo Munk for donating source code, and to Al Kossow, Dag Spicer, Udo Munk, and Len Shustek for contributing to this blog. Thanks also to the many other CP/M enthusiasts and former employees and consultants of DRI who inspired this source code release project, including Larry Greene, Herb Johnson, Emmanuel Roche, and John Wharton.

Bibliography

[1] "A Guide to PL/M Programming", Intel Corporation, September 1973 (http://bitsavers.org/pdf/intel/MCS8/MCS-8_PLMprogGuide1973.pdf)
[2] "Microcomputer software design – A checkpoint", by Gary A. Kildall, Proceedings of the 1975 National Computer Conference, pps. 99-106 (http://www.computer.org/csdl/proceedings/afips/1975/5083/00/50830099.pdf)
[3] "The History of CP/M. The Evolution Of An Industry: One Person's Viewpoint," Gary A. Kildall, *Dr. Dobb's Journal of Computer Calisthenics*, January 1980, No. 41, Vol. 5 Issue 1, pp. 6-7.
[4] "From Airlines Reservations to Sonic the Hedgehog: A History of the Software Industry", Martin Campbell-Kelly, MIT Press, 2003
[5] "Microsoft MS-DOS early source code", Computer History Museum (http://www.computerhistory.org/atchm/microsoft-ms-dos-early-source-code/)
[6] "Apple II DOS source code", Computer History Museum (http://www.computerhistory.org/atchm/apple-ii-dos-source-code/)
[7] "An Introduction to CP/M Features and Facilities", Digital Research (http://bitsavers.org/pdf/digitalResearch/cpm/1.4/CPM_1.4_Introduction_to_Features_and_Facilities_Feb78.pdf)
[8] "Principles of Design in the Octopus Computer Network", John G. Fletcher, ACM '75 Proceedings of the 1975 annual conference, pps.325-328 (http://dl.acm.org/citation.cfm?id=810357)
[9] "CP/M Source", Stephen A. Ness (http://nesssoftware.com/www/cpm_source.php)