WHYCPM.WS4

- "Compatibility: The key to CP/M"
  Michael Lehman
  "Digital Research News", Vol.2, No.3, September 1982, p.4

(Retyped by Emmanuel ROCHE.)

The key to the success of CP/M as a widely accepted operating system is compatibility. This compatibility ranges from hardware compatibility, data interchange, and flexible operating system options, to a vast array of compatible software.

The CP/M operating system's primary feature is its ability to support the concept of "Object Code Transportability". This means that a program which works on one CP/M system will work on another CP/M system, even if the CPU, disk, terminal, and printer are all different. One system could have an 8080, a mini-floppy, a teletype, and a matrix printer, while another could have a Z-80, a hard disk, an intelligent terminal, and a letter-quality printer. The user software interacts with these devices via CP/M which hides the differences, so that each application program perceives the system to have the same basic capabilities.

CP/M also supports compatibility by defining a standard for data interchange. Application programs can operate on a data file, and that file can be moved to another system. Because of the standard CP/M format, the other system may contain a different type of CPU (e.g., 8080/Z-80 disk moved to an 8086/8088), or a single system could contain two CPUs simultaneously, and a single data file could be manipulated by programs running on either CPU.

In addition to hardware and data compatibility, the CP/M family of operating systems ranges from single-user, single-tasking (CP/M), through networked (CP/NET) and single-user multi-tasking (Concurrent CP/M) to multi-user, multi-tasking (MP/M). Applications programs which adhere to the CP/M standard interface can run on all of these compatible operating systems. This provides the user with an upgrade path as requirements and hardware grow. No other microcomputer operating system family provides such a broad range of operating systems while still maintaining object code transportability.

Just as CP/M provides object code transportability, Digital Research provides source code transportability using our line of quality languages. Programs written in a Digital Research language run virtually unmodified across the entire range of CPUs on which CP/M is running (8080, 8086/8088, and 68000).

This provides the software writer the opportunity to write an application program once, and distribute it to a wide audience of computer owners with different hardware configurations, and different CPUs. The program maintenance and improvement costs are significantly lower, which means that the software writers can provide improved functionality to all of their users simultaneously.

In addition to compatibility between machines containing different CPUs, Digital Research provides productivity tools which provide compatibility between programs written in different languages and between programmers.

Access Manager provides a common data file format between programs written in different languages, just as CP/M provides a common disk format between different configurations and CPUs. Programmers now can write applications which create indexed data files, and manipulate the file in whatever language is convenient.

No longer is a user required to use the language the software was written in, but may use the language the user is most comfortable with. This greatly increases the flexibility of an application program, for both the user and the software writer.

Display Manager provides a portable interface to display devices. Screen-oriented programs can now be written which need not be manually reconfigured for each terminal device. A software writer can use such features as highlighting and underscoring and, if the user's terminal does not have these features, Display Manager will automatically adapt to the user's hardware. This will increase the availability of software which uses display devices efficiently. It also will provide a method for owners of less capable terminals to use the software without customization.

For the systems-level programmer, the Symbolic Instruction Debugger (SID) is a tool which is compatible with Digital Research languages. The linkers which accompany the language products can produce a SYM file containing information regarding the location of variables and subroutines in the compiled and linked program. The user then can set breakpoints, display and modify memory using symbolic names. This provides the programmer with the ability to debug without remembering where the linker located each item in the computer's memory.

Also for the systems programmer is XLT-86, which will convert assembly language programs for the 8080 into their equivalent program for the 8086/8088. Unlike other converters, XLT-86 actually compiles the 8080 assembly language into 8086/8088 assembly language, keeping track of register and flag usage, so that the resulting code is a true optimized translation, and not just a transliteration.

In summary, Digital Research provides a family of compatible operating systems, compatible programming languages, and compatible productivity tools which allow users and software writers to take advantage of a wide range of hardware, including computers with diffrent CPUs. Continue to look to Digital Research for compatible solutions to the software challenge.

EOF