THE UCSD P-SYSTEM MUSEUM

IN THE WEST WING OF THE JEFFERSON COMPUTER MUSEUM

TERAK MUSEUM - UCSD PASCAL MUSEUM - USUS LIBRARY
ANCIENT ALPHABETIC ART - LIBRARY - ALTAIR AND IMSAI EMULATORS
REVIVING CASSETTE DATA - DISK UTILITIES - COMPUTER RESCUE
WHAT'S WRONG WITH THIS PICTURE

News flash! A UCSD Pascal Reunion took place October 22, 2004...

What is the UCSD P-System?
It is a portable operating system that was popular in the early days of personal computers, in the late 1970s and early 1980s.

Like today's Java, it was based on a "virtual machine" with a standard set of low-level, machine-language-like "p-code" instructions that were emulated on different hardware, including the 6502, the 8080, the Z-80, and the PDP-11. In this way, a Pascal compiler that emitted p-code executables could produce a program that could be run under the P-System on an Apple II, a Xerox 820, or a DEC PDP-11.

The most popular language for the P-System was UCSD Pascal. In fact, the P-System operating system itself was written in UCSD Pascal, making the entire operating system relatively easy to port between platforms.

By writing a p-code interpreter in the platform's native assembly language, and a few minimal hooks to operating system functions for the file system and interacting with the user, you could move a p-code executable from another system and run it on the new platform. In this way, the p-code generated on one computer could be used to bootstrap the port of the P-System to another computer.

Wirth, Jensen and Pascal
Niklaus Wirth first developed the Pascal language around 1969, with the first version implemented on the CDC 6000 in 1970. By 1983, it was an ISO standard language.

Prof. Niklaus Wirth

"Europeans call him by name, but Americans call him by value."

I asked Prof. Wirth if he still had the source code to any of these early Pascal compilers. At first he said "no". Later he found an old CDC 6400 7-track mag tape with the source, but we haven't found a way to read it yet...

Prof. Wirth believes it was UCSD Pascal that truly brought Pascal into its own. In his 1985 Turing Award Lecture, "From Programming Language Design To Computer Construction", as published in Communications of the ACM, February 1985, V28 N2, he said "But Pascal gained truly widespread recognition only after Ken Bowles in San Diego recognized that the P-system could well be implemented on the novel microcomputers. His efforts to develop a suitable environment with integrated compiler, filer, editor, and debugger caused a breakthrough: Pascal became available to thousands of new computer users who were not burdened with acquired habits or stifled by the urge to stay compatible with software of the past."

The UCSD Pascal compiler descends from Wirth's early compilers, through a compiler as "P2" as adapted by Urs Ammann for his doctoral thesis. The UCSD adaptation was done under the guidance of UCSD professor Kenneth Bowles and other personnel including Stephen Franklin and Alfred Bork. If you have source to early Pascal compilers, I'd love to see it.

This CDC 6400 version of Pascal was not a byte-code interpreter. It was a one-pass compiler that relied on the CDC's linking loader to resolve jumps. CDC linking was performed at program load time.

Circa 1973 Pascal compiler source code
In July 1997, Huw Davies of the La Trobe University in Melbourne, Australia sent me the source code to Jensen & Wirth's Pascal assembler and interpreter, "pasint.pas" circa March 15, 1973, and the "modified step 5" compiler "pcomp.pas" circa May 20, 1974. Shift-click here to get a 39K Zip file of it.

Davies said, "While waiting for the paint in my office to dry I've mounted a few of my old DECsystem-10 tapes on one of our VMS systems and restored the original source code for both this and Pascal-S (part of my honours project so many years ago :-) I suspect that these programs arrived at La Trobe on a tape from either ETH directly or from DECUS (this is well before we had an external network link, probably either 1974 or 1975). My honours project involved a survey of then current compiling techniques and a complete documentation of the Pascal-S compiler (basically the P2 compiler with some features removed, but basically the same code). I can't recall whether I got Pascal-S on the same tape or not - it's been a while..."

Pascal-S was a byte-code interpreter, aimed at faster compilation and no linking step in order to minimize CPU time (which had a dollar value back then) for running student programs.

The source code to UCSD P-System and UCSD Pascal
I'd like to get permission from UCSD to redistribute the source code and executables for the archaic versions of the UCSD P-System version II and UCSD Pascal. They have granted a redistribution right to a group of TI-99 revivalists, so I'm hopeful.

I'm also pessimistic because the people at UCSD licensing have dollar signs in their eyes, and they believe that they should extract cash from anyone who wants the rights to something the University developed, even if it is so old as to be useless. Fortunately, one of the P-System's creators believes that these archaic versions should be clearly freed to the public domain, so it might happen someday.

All of this is complicated by the fact that a company in the United Kingdom, Cabot Software, continues to sell newer versions of the P-System under non-exclusive license to UCSD, although they are the sole licensee at this time. They're promoting it as an alternative to Java in set-top boxes and consumer appliances.

The source to the Terak version of UCSD Pascal and P-System version I.3 (August 1977) was uploaded to CompuServe's PDP-11 Forum in July 1984 by one of the earlier UCSD licensees, Eli Willner of Pecan Software, and remains there to this day - certainly one of the oldest files still on CompuServe. According to Willner, "Since this code was developed with government funding, it is public domain, and the University has so acknowledged. If this is all you're interested in, then, as I understand, no license is required at all."

The source to at least two versions (I.3 and I.5) were passed around freely in the days of the Terak. Before commercial interest in it grew, UCSD itself copied the source code disks for a small handling fee. I have a copy of the source to version I.5 (September 1978).

The UCSD P-System was developed at the Institute for Information Studies at the University of California - San Diego, under the direction of Kenneth Bowles.

A Terak And UCSD P-System Emulator
Although many Terak programs made use of Terak-specific assembly language calls, almost all programs called these functions through a standard set of pre-written library functions that could be easily emulated by function, which should mean that we wouldn't have to do a low-level hardware emulation of the Terak in order to run almost all programs. At worst, a better emulation would integrate a PDP-11 emulator for the unusual custom in-line assembly language calls.

In my spare time, I have begun an ANSI C implementation of a P-System disassembler and interpreter. In theory, combined with similar emulations of library functions for other platforms such as the PC or Apple II, this interpreter could run P-System executables from many systems. I hope to make it extensible and configurable, and to run under Windows and X Windows. Progress has been aided by several native disassemblers on the Terak.

If anyone can direct me to a good reference to the P-codes used in the UCSD P-System version I.3, I.5 or even II, I'd appreciate it. One such reference might be a copy of the Western Digital Microengine, a hardware implementation (a CPU, actually) that understood the p-codes of the P-System.

A working UCSD P-System Emulator!
Here is a link to a UCSD P-code interpreter executable for DOS written in Turbo Pascal by Laurie Boshell <laurie@pnc.com.au>, a fellow in Australia who also hopes to tinker with UCSD P-system tools. If you transfer your .COD files to a PC file, it will run them with a certain degree of success. Shift-click here to get a Zip file of it.

My tools for reading RT-11 and UCSD P-System disks
See the Disk Utilities exhibit for a description of my methods for transferring floppy disk images to other computers for archiving and restoration. Source code is included.

Other UCSD Pascal links and tidbits

UC-Irvine
There's a lot of information about the UCSD P-System at the University of California at Irvine, among the archives of computer systems formerly in use there.

The P-System and Java
People mention the UCSD P-System when they recount the predecessors of portable, interpreted virtual machines like Java. In theory, a P-System executable program was portable between machines with different processors because the program was not expressed in machine-dependent assembly code but instead the "p-code" machine language of a virtual computer. Java uses the same approach to gain platform independence.

The P-System and the IBM PC
Most Teraks ran the UCSD P-System operating system. The UCSD P-System was one of two other operating system available for the original IBM PC in 1981 - the other operating systems being PC-DOS a.k.a. MS-DOS 1.0 and CP/M-86. PC-DOS was priced must lower than these alternatives, and the rest is history.

The P-System and the Apple II
It was UCSD Pascal II that mutated into Apple Pascal and UCSD Pascal IV along separate paths of development. In 1978, UCSD students Richard Gleaves and Mark Allen used UCSD Pascal to develop the 6502 interpreter which became the basis for Apple Pascal.

The P-System and NCR
It NCR once made a computer based on the AMD 2901/2910 bit-slice processor that ran UCSD Pascal, in an engine developed by Joel McCormack, a UCSD alumni.

Hardware to run P-codes
Apart from being a representation of an idealized but non-existent assembly language, one of the concepts behind the P-code system was that it was closely matched to the kind of data that is emitted by a compiler.

Western Digital, best known today for making hard disk controllers, also developed the Microengine, a hardware implementation (a CPU, actually) that understood the p-codes of the P-System.

Advanced Computer Design developed an operating system for the Western Digital Microengine called AOS, or the Advanced Operating System, that included multi-threading and the equivalent of DLLs, etc. long before many other personal operating systems.

Pascal and C
Pascal and C developed contemporaneously in the early 1970s, but by most measures, C became the dominant programming language for applications for personal computers and workstations.

An interesting historical document describing this turning tide is Brian W. Kernighan's "Why Pascal is Not My Favorite Programming Language" , Computing Science Technical Report 100, AT&T Bell Laboratories, from April 1981.

Borland Turbo Pascal

Although it is not the P-System, it brought Pascal to the masses and played a major role in the decline of UCSD Pascal. Its integrated development environment was novel and useful, and its native x86 code ran faster than interpreted Pascals. Now Borland is giving away copies of Turbo Pascal at its Museum.

UCSD Trivia
For while, UCSD P-System programmer Roger Sumner had the personalized California license plate "77FILES", after the limit of the number of files on a UCSD P-System floppy.

At one point in the late 1970s, NASA released a news report that claimed that some of the data returning from the Voyager spacecraft was lost due to this 77 file limit.

The original UCSD P-System team has been meeting every year for almost twenty years, on the drawbridge at Disneyland at noon on the day after Labor Day.

It is possible to write data onto the 0th block of a UCSD P-System eight-inch floppy. The quiz system used this block to transfer pass/fail information on the quiz floppies.

The P-Machine documented in the I.5 manual is actually the I.3 P-machine.

No electronic copies of the UCSD P-System manuals still exist. Only paper copies exist. Something happened to the one floppy that held the last copy.

The startup logo on the Terak version of UCSD Pascal of the "Triton" is modeled after the 1977 UCSD student parking sticker.

Thanks to Keith Allan Shillington for most of these tidbits. Other people involved in the development of UCSD P-System include Dr. Kenneth Bowles, Mark Overgaard, Roger Sumner, John Van Zandt, Richard Kaufmann, Steven Thomson, Peter Lawrence, Joel McCormack, Bill Atkinson, Julie Erwin, Gillian Ackland, Lucia Bennett (now Yandell), Robert Hofkin, Shawn Fanning, Barry Demchak, C. Richard Grunsky, Bruce Sherman, George Symons, Karen Fraser, Mark Allen, Richard Gleaves and Nancy Lanning.

More UCSD Pascal history
I was also contacted by Richard Gleaves regarding UCSD Pascal history. He said:

> "Many thanks for your work in documenting the history of UCSD Pascal. The only thing I have left from back then is an 8" Terak floppy disk (brand = Dysan; disk label = "Rich"; contents = unknown). I hope it's not too forward of me to humbly request that my name (and especially Mark Allen's, too) be added to the list of original UCSD P(ascal developers.
>
> Mark and I spent the summer of 1978 working on-campus at UCSD writing the 6502 interpreter that later became the basis for Apple Pascal. (We were paid UCSD's standard student "junior coder" wage of $5.50 per hour.) Trivia detail: the interp was developed (and thus the first 6502-based Pascal system booted) on a Rockwell box.
>
> At the end of the summer Bill Atkinson started showing up in the lab, and he worked closely with Mark to get the thing going on the Apple II. Later Bill offered Mark and me jobs up in Cupertino, but we both turned him down because we both wanted to stay in San Diego...

> Later, I worked with Barry Demchak (bdemchak@tpsoft.com) at ACD to co-develop the AOS (Advanced Operating System) variant of UCSD Pascal. Later, I and some of the other UCSD types got into the even more Java-like Modula-2, but C ate the marketplace.
>
> It's neat to see Java recycling all the old ideas (including one big one that never quite made it into the P-system: when Barry and Mark and I were at SofTech Microsystems, and the IBM PC had just come out with DOS as the defacto standard, we tried long and hard to convince the SofTech management that the right way to go was to make the P-system a portable programming environment that ran on the DOS operating system and directly accessed the DOS file system... or any other host op sys (such as UNIX). But they wouldn't buy it -- they thought there was more money in positioning the P-system as a competing operating system."

USUS Mysteries
I'd like to find out what happened to the Pascal user group known as USUS, the UCSD p-System System Users Society. For many years, they maintained a library of Pascal floppy disks containing source code and programs. There's no trace of them on the net, and no one seems to know what happened to them. If you know, please send me an e-mail. I'd like to consider keeping a copy of their archive alive for this web page. Update 12/99: There's a new USUS Software Library exhibit in the Jefferson Computer Museum.

If you're mystified...
Or otherwise searching for UCSD Pascal information because you've been asked to research how your company can switch from some out-dated and archaic P-System database to a contemporary computer, I'm not sure you've come to the right place. However, a lot of people land on this web page for this reason.

To answer the most common question about rescuing the old data from these programs, if you have the Pascal source code, you have a chance of porting it to a newer Pascal. With the source code, you have a description of the data stored in the files. A competent programmer should be able to write a utility to translate the old data to a newer neutral format that could be imported into a newer database or application. Without the source, you'll need to decode the data files. Again, a programmer might be able to reconstruct the data, given a detailed description of what you know is stored in the files.

There's even a freeware program known as 'p2c'. It is available in C source code form and is easily recompiled for Unix and Linux systems. With this, you can automatically translate UCSD Pascal (and other Pascal variants) directly to C code. This isn't a perfect solution due to language- and machine-specific functions such as screen graphics, certain types of I/O, etc. but it will translate the majority of the code to C. With this, you can more easily perform the translation of old data files or old code into something more easily adapted for today's computers.

Inventory of Pascal stuff
Here is a list of the UCSD Pascal software and documentation that I have:

Software

UCSD p-System IV.0 r1, IBM version, circa 1982. 5 1/4" disks, "STARTUP," "PASCAL," "EXTRAS," "UTILITY," "SYSTEM 4," "SYSTEM 2". *

Pecan UCSD Pascal, DOS-hosted Pascal Power System, 3 1/2" disk 1/2 is LAP/PSYS1.PAS0, IV.2.2 R[1.1]ba, disk 2/2 is LAP/PSYS2-0. Disk 3, "8086 Assembler LAP/ASM-0 I.V.2.2 R[1.0]B *

Heath Z-80 UCSD p-System, six 5 1/4" disks: "SYSTEM 1," "SYSTEM 2," "ZINT," "PASCAL," "UTILS," "Native code generator NZ84BP". *

Documentation

UCSD Pascal User Manual, inside "UCSD Development System," Pecan Software, 594 pages, circa June 1986. For MS-DOS and Amiga.

UCSD Pascal User Manual, inside "The Pecan Power System," Pecan Software, 197 pages, copyright 1986. For MS-DOS and Amiga. *

The UCSD Pascal Handbook, by Randy Clark and Stephen Koehler, Prentice-Hall, 356 pages, copyright 1982. *

The Pascal Handbook, Jacques Tiberghien, Sybex, 473 pages, copyright 1981.

The BYTE Book of Pascal, Blaise Liffick, ed., BYTE Publications, 333 pages, copyright 1979.

[LIST INCOMPLETE, see also Terak info...]

* Graciously donated by Earl F. Glynn and Louis-Luc Le Guerrier

---

Return to the main page of the Jefferson Computer Museum.

www.threedee.com - www.goJefferson.com - www.foust.org - www.saltglaze.com