

Date: Mon Nov 26 05:45:54 1992
Subject: Re: A little Pop history
From: Ray Dunn
Volume-ID: 921126.03

This has turned into a long posting, for which I apologize, but hopefully some will be interested in the reminiscences of an old hacker and the early history.

In ref'd article, axs@cs.bham.ac.uk (Aaron Sloman) writes: >Pop10 was implemented for the PDP10 in Edinburgh by Julian >Davies and was used there and in various other places including >Toronto I think, for several years in the mid 70s.

I have no recollection of Pop10, but POP-2 (please notice the capitalisation and the hyphen) for the PDP10 (later known as the DECsystem 10) was implemented by myself and Malcolm Atkinson (now of Cambridge) in 1970 or 1971.

Didn't Julian implement an early version of POP-2 based Prologue?

Malcolm worked from Cambridge and I worked from Edinburgh into the PDP10 of TimeSharing Ltd in London, via 10cps model 33 teletypes in our homes - is this the earliest case of teleworking in the UK? (-:-).

After 3 months of design and heavy coding (and enormous telephone bills) we met one Easter Weekend in London and compiled the system for the first time. By the end of the weekend we had 2+2 => working!

The initial implementation of POP-2 was "Multi-POP", in the Department of Machine Intelligence and Perception in Edinburgh University. It was done on a Elliott Automation 4120 (later ICL 4130) by Robin Popplestone, Dave Pullin and myself, and was an eight user multi-access system based totally on POP-2 (which was the control language of the system also).

This came live in 1967 (possibly late 1966). It's now said that we later realized that we were the first online multi-access system to come live in the UK, several weeks ahead of the Cambridge system, but I've no proof of that nor do I remember when the Cambridge system went on-line!

When you consider the research output of the DMIP in the late sixties and early seventies, it is difficult to believe that all the work was done on a machine with 64K 24-bit words, 2 microsecond add time, through model 33 10cps teletypes.

A PDP7 with a model 340 (?) vector graphics display and later a Honeywell 316 were linked to the 4130 through a "high-speed" data link (into the ATU of the 4130 for devotees).

The PDP7 was used by John Oldfield and his group for early CAD system research, and the H316 was the control processor for FREDDY, the hand-eye robot system that formed the focus of much of the department's research. We would have used a PDP11 instead of the 316, but DEC announced that machine a matter of days after we'd ordered the Honeywell (shame - the order was not retractable)! I could probably still key in the boot code on the front-panel hand-keys of the 316 if I had to!

After the technical success on the 4100, POP-2 was implemented on the following systems in addition to the PDP10:

ICL System-4 by John Barnes,

ICL1900 under GEORGE3 by John Scott & Frazer Dingwall (now with Digital),

CDC6000 under SCOPE by Ken McCallum (Strathclyde) and Laurence Shafe,

MODULAR-1 again by Frazer Dingwall,

and (believe it or not) also on the

IBM 360/370 under TSO by Rod Steel

(so much for the complaints about the Intel architecture for POP implementations!)

The foundation of these implementations was a machine independent compiler written in POP-2 itself.

Much of this later compiler work was funded by Conversational Software Ltd, a spin-off company from the Department under the Chairmanship of Donald Michie.

I was the Technical Manager of CSL, and the Managing Director was Peter Burt who is probably the only person connected to POP-2 to achieve perpetual fame - he is now the Treasurer and Chief General Manager of the Bank of Scotland - his is the signature on the Scottish pound notes and fivers etc!!

Robert Rae (now with AIAI in Edinburgh) was one of many who later slaved at enhancing the 4130 and DECsystem 10 implementations.

There were two main POP-2 reference manuals published in addition to articles in the Machine Intelligence series:

The "Blue Book": "POP-2 Papers" by R.M. Burstall, J.S. Collins and R.J.

Popplestone, Edinburgh University Press, 1968.

and

The "Silver Book": "Programming in POP-2" by R.M. Burstall, J.S. Collins

and R.J. Popplestone, Edinburgh University Press, 1971.

As Rod Burstall and Robin Popplestone wrote in "Programming in POP-2":

"POP-2 is a new computer language. Conceptual affinities can be traced to:

1. John McCarthy's LISP (1962), from which it takes ideas for handling non-numerical objects of computation (lists).
2. Christopher Strachey's CPL (1963) and Peter Landin's ISWIM (1966), which foreshadow the aim of making a programming language into a notation with full mathematical generality, akin to algebra.
3. Cliff Shaw's JOSS (1964), which it resembles in its 'conversational' facilities.

[Note: Those who baptized this group with its first irrelevant flame war over the lack of a BASIC newsgroup, should be interested to know that JOSS is also the grand-daddy of BASIC]

4. Robin Popplestone's POP-1 (1968) of which POP-2 represents a rationalized and greatly extended development.

[Note: 1968 is the date of publication of Robin's paper on POP-1 in

Machine Intelligence 2. POP-1 and its precursor COWSEL had been around for several years previously. The first POP-2 reference manual appeared in the same book, and POP-2 had already been running for at least a year - was Robin stuck for a paper to present at the conference? (:-)

"Machine Intelligence 2", edited by E. Dale and D. Michie, Oliver & Boyd, Edinburgh, 1968]

"These ingredients have produced a powerful but compact language for non-numerical programming. POP-2 was designed for implementation on a medium-sized machine with a modest investment in system programming.

[Ha - it was easy for THEM to say that! (:-)].

Because the language had to be stripped down to the level of the basic mathematical principles of programming, it is unrestrictive and open-ended.

The main distinctive features of POP-2 are:

1. The syntax is very simple but the programmer has some freedom to extend it.
2. The programmer can create a wide variety of data structures: words, arrays, strings, lists, and records. A 'garbage collector' automatically controls storage for him.
3. Functions can be used in the same manner as in mathematics or logic, for example, as arguments or results of other functions, with no unfortunate restrictions on free variables.
4. The novel device of 'partial application' allows one to fix the value of one or more parameters to the function. This has a surprising multiplicity of uses, for example, to generalize the notion of an array to non-numerical subscripts and to disguise the distinction between stored values and computed values.
5. Another technique, 'dynamic lists' enables a physical device like a

paper-tape reader to be treated as if it were an ordinary list.

6. The programmer can call for immediate execution of statements at any time, giving facilities for conversational use and rapid debugging of complex programs.

7. The facility for immediate execution together with the variety of data structures available makes POP-2 suitable for use as the control language of a time-sharing system, enabling the user to effect filing, editing, compilation and execution.

8. In the context of the widespread shortage of system programmers, a crucial feature is the open-endedness of the language. Work normally done in machine code by highly skilled system programmers can be done in POP-2 itself.

9. POP-2 is compact and easy to implement. On the ICL 4100, for example, the whole system for compiling, running, and time sharing occupies only 22K of core (24-bit words). The effort needed to construct the complete system was less than 5 man-years. A machine independent POP-2 in POP-2 compiler has been written.

Unquote.

Later, in the reference manual section, the main features of the language are detailed with their associated predecessors:

"The following main features are provided. Roughly analogous features of some other programming languages are mentioned in brackets as a guide:

(a) Variables (cf. ALGOL but no types at compile time)
(b) Constants (cf. ALGOL numeric string constants, LISP atoms and list constants)

(c) Expressions and statements (cf. ALGOL)

(d) Assignment (cf. ALGOL, also CPL left-hand functions)

(e) Conditionals, jumps and labels (cf. ALGOL)

(f) Functions (cf. ALGOL procedures but no call by name, cf. CPL and ISWIM for full manipulation of functions)

(g) Arrays (cf. ALGOL; also CPL for full manipulation of arrays)

(h) Records (cf. COBOL, PL/1, Wirth-Hoare ALGOL records, CPL nodes)

(i) Words (cf LISP atoms)

(j) Lists (cf. LISP, IPL-V)

(k) Macros

(l) Use of compiler during running (cf. LISP, TRAC)

(m) Immediate execution (cf. JOSS, TRAC).

Unquote

It's interesting to note that Rod Burstall didn't initially believe that the open stack should be a concept that the user was overtly aware of (that clearly goes against the principle that the language is a mathematical notation), and although he mentions the stack when he has to, no emphasis on the stack is given in either of the manuals.

I had a very different view, and explained the stack mechanism in some detail in the user manuals I wrote for the Multi-POP and PDP10 versions.

Finally, I think it's appropriate to repeat the acknowledgement that appears in the introduction to Programming In POP-2, so that the remainder of the early pioneers' names are written into this record:

"The suggestions and criticism of many members of the Department of Machine Intelligence and Perception at Edinburgh University, especially Mr Ray Dunn, and of others outside it, especially Mr Michael Healy and Mr Michael Woodger and those who have implemented POP-2 systems, are gratefully acknowledged. Technical consultations with Dr Michael Foster and Dr David Park about storage control were invaluable. Thanks are due to Mrs Pat Ambler who has helped in the editing of this book, to Mr Bruce Anderson who has provided answers to the exercises in the primer and to Miss Eleanor Kerse who typed the new manuscript through numerous drafts.

"POP-2 has been implemented on the ICL 4100 by Messrs Robin Popplestone, Ray Dunn, and David Pullin, on the ICL 1900 by Mr John Scott, and on the ICL System-4 and IBM 360 by Mr John Barnes and Mr Rod Steel (using a machine independent version of the compiler written in POP-2). The library of programs has been built up by many hands and organized by Mr Ray Dunn and Mr Robert Owen. The work has been part of a machine intelligence project, directed by Professor Donald Michie, whose guidance and encouragement have been invaluable, and supported by Edinburgh University, the Science Research Council, and the Medical Research Council.

"We would like to thank Edinburgh University Press for their painstaking work on a very technical book, especially Dr Helen Muirhead, whose care and patience never failed us.

R. M. Burstall (Editor)

Unquote

--

Ray Dunn at home		Beaconsfield, Quebec		Phone: (514) 630 3749
ray@philmtl.philips.ca		ray@cam.org		uunet!sobeco!philmtl!ray

Laurence Shafe (who worked on the CDC6000 version of POP-2 with Ken McCallum) can be found at <http://www.shafe.co.uk> and is still in the computer industry. He founded Intelligent Environments in 1985 and floated the company in 1996, some say the first Internet company to float in the UK. He is now working for the M&A advisor The Catalyst Group and studying for a History of Art degree.