A Letter from Ritchie

Brian Carpenter
Honorary Academic
The University of Auckland
New Zealand

July 2023

In 1973, I was a systems programmer, in my first job after university, in the Controls Group of the MPS (Proton Synchrotron) Division of CERN, the European particle physics lab in Geneva, Switzerland. I'd been working for two years on software for Imlac PDS-1 display minicomputers used by the synchrotron's operators. The Imlacs were networked in a very simple way to an IBM 1800. We knew that we needed more computers to progressively replace the old manual control system for the rapidly expanding synchrotron complex. We knew they'd be minicomputers, and in 1973 the obvious choice was PDP-11s. In those days, minicomputers were essentially sold unbundled and you had to pick your software as an afterthought. Our bosses were engineers and physicists who, to be frank, didn't understand software.

So, we bought a couple of PDP-11s and began to wonder what to do with them. The plan was clearly to build a network; there was no such thing as Ethernet, so the network was designed as a star. Somebody else was designing the data links that would connect any two PDP-11s. I got the job of figuring out the software details, since I was the local expert after designing the *very simple* Imlac network. The first thing I needed was an operating system.

Enter Mark Krieger. Mark was an ebullient New Yorker who was in our group for a year or so as a Scientific Associate. He helped with the software design of the network we were planning to build. I think it must also have been him who showed me the ACM SIGOPS proceedings with an article about a new operating system for PDP-11s [1]. (The exact text of that article is hard to find, but the final version is available [2].) Mark soon returned to New York and became one of the founders of Whitesmiths Ltd., along with P. J. Plauger and Gabriel Pham. Whitesmiths created Idris, the first Unix knock-off, and the first commercial C compiler.

After reading about Unix, I thought it looked much more interesting than Digital Equipment's rather scrappy offering of PDP-11 operating systems, so I did what you had to do in 1973 – I sent an airmail letter to the authors of the Unix paper, and Dennis Ritchie graciously replied within a few days [3]. Technically, his reply was very interesting, but the bureaucracy (and probably serious money) needed to get hold of Unix was, unfortunately, out of my league at the time (especially with bosses who believed that all software was a small matter of programming).

So I did what any systems programmer would have done in 1973. I continued work on my own small operating system for the PDP-11/10 that was supposed to act as a message switch. It was called MINIMON and was coded in PL-11 [4]. In 1974, I took both MINIMON and PL-11 with me to Massey University in New Zealand, where they were used to build a home-made terminal concentrator. Unix and C finally entered my life about 10 years later.

My letter to Ritchie and Thompson is long gone, and I was astounded when Lars Brinkhoff recently told me that Ritchie's reply had resurfaced via an ancient DECtape. Ritchie was only a few years older than me, so I suppose his helpful letter can be described as collegial, but I felt that it showed exemplary professionalism. Fifty years later, I feel the same.

References

[1] Dennis M. Ritchie and Ken Thompson. 1973. The UNIX time-sharing system. SIGOPS Oper. Syst. Rev. 7, 4 (October 1973), 27. https://doi.org/10.1145/957195.808045

[2] Dennis M. Ritchie and Ken Thompson. 1974. The UNIX time-sharing system. Commun. ACM 17, 7 (July 1974), 365–375. https://doi.org/10.1145/361011.361061

[3] Dennis M. Ritchie, letter to B. E. Carpenter (file written 1973-12-03). To be found at https://www.tuhs.org/Archive/Applications/Dennis_Tapes/ as file dmr_tapes/dmr2/let/lett9. Text attached below.

[4] Russell, Robert D. (1974). T. C. Streater (ed.). PL-11: A Programming Language for the DEC PDP-11 Computer. CERN. https://cds.cern.ch/record/880468/files/CERN-74-24.pdf

Mr. B. E. Carpenter
MPS Division
CERN
1211 Genève 23
Switzerland

Dear Mr. Carpenter:

This is in response to your letter of November 27, reference MPS/CO/BC/je.
The UNIX system, its software, and its documentation are proprietary to the Bell System and are currently being released, under a license from the Western Electric Company, only to educational institutions.
The possibility of extending such licenses to governmental and commercial organizations is being considered, however.
To obtain information about such a license, you may write to

> Mr R. G. Shahpazian
> Room 400
> Western Electric Company
> 222 Broadway
> New York, New York, 10038

You should also send a copy of your letter to

> Mr. S. P. Morgan
> ...

Mr. Shahpazian will acknowledge your letter and will inform you as soon as a decision has been reached.

In answer to your specific questions:

There is currently no user-available method to lock a process in core to meet real-time constraints. This should be very easy to add, however, since the system has internal to it a flag which prevents swapping the process's core image.
Somewhat more work would be required to arrange that physical core occupied by the process lies at the edge of the available space, which is desirable to prevent

fragmentation of the physical address space.

The system will run with only one RK disk, although naturally
both response time and available disk space are better with the
swapping disk space and user file space on
separate devices.
Incidentally, I think the SIGOPS conference version
of the paper was overly optimistic about the
amount of memory needed;
we are unable to support machines with less than
32K words of core.

There is no automatic means of initiating a process in response to
an interrupt.
The approach we would take to this need is to have an existing
process which wakens in response to the interrupt and creates
a new process. Unless the time requirements are very strict
I don't think this is a problem.

Although UNIX for non-segmentation PDP-11's still exists,
we are unable to support it, since its internal structure is
so different from the current system and since
it requires so much handcrafting to adapt it to the
hardware configuration.

It is trivial to introduce a new user program such as a
compiler, since one need merely place the file
containing the program in a particular directory.
New device drivers are in general fairly easy,
since the interface between the system and the drivers
is quite clean. It does require some
knowledge of the way the system operates.

Fortran definitely supports the floating point hardware;
in fact we recommend it if much use is to be made of Fortran.
The compiler generates the same kind of semi-interpretive
"threaded code" as does the DEC Fortran; I suspect it
is somewhat slower than DEC's since Fortran is not heavily
used by us, and we have not felt impelled to optimize
as carefully.
If you are at all adventurous in this area I can advise
using "C", which is the language in which

UNIX itself and most of its software are written.
C is a true compiler which generates excellent code and
supports both single- and double-precision floating-point
arithmetic.

I can give no definite answer as far as cost is concerned.
As I indicated, UNIX is
currently being made available only to educational
institutions; this is on a cost-free basis.
I can't predict any charges that might be
made should licenses be extended
on a broader basis, but I doubt if they would seem excessive.
I'm not sure what you mean by "software support."
Of course all the ordinary software comes with the system:
editor, language processors, utilities, and so forth.
If you refer to continuing support,
the situation is still not very clear.
We hope to be able to send out occasional mailings
describing bugs found and new software, and to
generate tapes of the appropriate source programs.
Still, you will have to take into account
the fact that your only source
of information is a 2-man operation an ocean away.

DMR

Copy to SPM