Guest Editorial[1]

Creative endeavour varies greatly in the amount of overhead (i.e. money, manpower and organization) associated with a project which calls for a given amount of creative work. At one extreme is the activity of an aircraft designer, at the other that of a poet. The art of programming currently falls much closer to the former than the latter. I believe, however, that this situation is likely to change considerably in the next decade. The following remarks state (perhaps overstate) the reasons for this belief.

Programming, after all, is a rather abstract activity which can in principle proceed without a great many interactions with the real world. Furthermore, because of the reflexive nature of the computer it is possible to amplify the effectiveness of a skilled practitioner enough to overwhelm any difficulties of sheer magnitude. Unfortunately, the practical consequences of these facts are blunted by two other conditions. One, which is not likely to change much, is that many programs must function as part of a larger system through interfaces which are often complex and usually ill-defined. The other is the extremely high overhead associated with most kinds of programming.

This overhead has three major components:

(1) The hardware and software system needed to develop and run a program.
(2) The administrative apparatus required to operate this system, allocate its resources and finance it.
(3) The constraints imposed by this apparatus on who writes programs, what he works on and what he can do with the resulting product.

To appreciate its consequences, compare a programmer's life style with a writer's, or the distribution system for software with the publishing business.

New hardware technology strikes at the roots of this jungle by making it possible to produce a system roughly comparable to a 360165 in computing power for a manufacturing cost of perhaps $500 (in quantities of 100,000), divided more or less equally among:

(1) A 16-bit, 100 ns/instruction processor.
(2) 32 kilobytes of main storage.
(3) 1 megabyte of swap storage with 1 ms latency.
(4) 2-4 tape cartridges with 2 megabyte capacity, 20 kilobyte/second transfer rate and 10 second average latency.
(5) A 2000 baud modem which transmits over the dial-up telephone network.
(6) An 875 line raster-scan CRT which can display a double-spaced typewriter page of text with nearly typewriter quality.

Such a device can probably be built in five years, certainly in ten; the pacing item will be the swap storage. It has three important properties: a price which anyone can afford, reliability at least as good as a television set and enough power (processing and storage) to support large programs.

Some progress in software technology is also discernible. Here I am less willing to give dates,

---

[1] This editorial appeared in *Software— Practice and Experience* **2**, 1972, pp 195-196. This version was created by scanning the original and doing OCR; it may have errors.

but I think it plausible that we will someday have:

(1) Both the facilities (languages, translators, operating systems) and the doctrine to write programs which are truly modular (or structured, if you prefer).

(2) A set of rules for coding with the property that programs which obey them can be mechanically verified, and are less than twice as long as equivalent programs which do not.

In this Utopian future we will have reliable software components, and we will not have conversion problems in moving programs from one environment to another.

When (if) these predictions come true, the economics of programming will be very different. For one thing, the entry cost will be very low, in the same sense that the entry cost for writing is low today. As a result, millions of people will write non-trivial programs, and hundreds of thousands will try to sell them. Of course, the market will be much larger and very much more diverse than it is now, just as paper is more widespread and is used in many more ways than are adding machines. Almost everyone who uses a pencil will use a computer, and although most people will not do any serious programming, almost everyone will be a potential customer for serious programs of some kind Furthermore, such a mass market will require mass distribution. Analogues of bookstores, news-stands and magazine subscriptions seem plausible, as well as the kind of mail-order and home improvement marketing patterns we have now.

I expect that most readers will find the picture I have been painting rather phantasmagoric, especially since I have not provided the kind of corroborative details which can be culled from the pages of *Electronics* or the *Communications of the ACM*. Unfortunately, a more convincing argument would need a book, so I must stand on what is written here and what you may find between the lines.

BUTLER W. LAMPSON
Xerox Corporation,
Palo Alto Research Centre,
California,
U.S.A.