

## ***LANGUAGES – IMPLICATIONS***

The primary implication of the existence of languages intended for writing operating systems is surely that we need compilers for them, which is fairly obvious. The implications for the language of the need to write operating systems are more interesting. We can readily see why this is so by contemplating the dustbin metaphor for the operating system; if it has to cope with everything that no one else wants to do, then the language really must be able to do everything.

It was realised a long time ago that there would be problems. Here is an extract from an early paper<sup>IMP17</sup> on the language NPL, later renamed PL/I :

**2. Full access to machine and operating system facilities.** If there is a system which supports NPL, and if there are functions which a programmer can accomplish more easily, or only, by assembly language programming, then NPL has failed in this criterion for this system. Certainly there will be many instances in which NPL will, in fact, fail for different systems. But the presence of this as an explicit goal meant that no facility was discarded because it belonged more properly to assembly or control-card languages.

By these criteria, PL/I did indeed fail for ( we suspect ) all the machines on which it was implemented, but the reason had less to do with the linguistic features of the language than with the limited understanding of what was needed to support an operating system. ( That might be too harsh : later versions of Primos were written in PL/I, and were certainly an advance on the earlier versions, written in Fortran. ) An important implication of the need for these languages is therefore the need for a thorough understanding of operating systems themselves.

More recent attempts to produce languages for writing operating systems have certainly not followed PL/I in aiming for products applicable in all fields; rather, they have concentrated on special-purpose languages incorporating the facilities which seem to be appropriate for the job.

All in all, though, the languages do not in any way require any sort of "lower level" support which has to be implemented in the system. This is just what one would expect, of course, but it's always reassuring to have one's expectations confirmed.

### REFERENCE.

IMP17 : G. Radin, H.P. Rogoway : "Highlights of a new programming language",  
*Comm.ACM.* **8**, 9 ( 1965 ), reprinted in *Programming systems and languages* ( S. Rosen, editor : McGraw-Hill, 1967 ).

---