

SCIENCE AND ENGINEERING RESEARCH COUNCIL
RUTHERFORD APPLETON LABORATORY

COMPUTING DIVISION

D I S T R I B U T E D C O M P U T I N G N O T E 5 8 3

ICL/PERQ/SERC COLLABORATION PAPERS

PASCAL IMPLEMENTATION

Issued by
R W Witty
24 February 1982

DISTRIBUTION: ICL R Chamberlain
 P C L Shave
 H S Woodgate
 SERC F R A Hopgood
 R E Thomas
 R W Witty
 L O Ford
 J M Loveluck

Attached are copies of two papers, one from ICL and the other from SERC, in response to actions 1.2 and 2.7 of the meeting of 1st February 1982. These papers define the ICL and SERC preferred versions of Pascal.

The Common Base PASCAL Question

1. Introduction

Pascal is a slightly variable commodity. It is important that precise a definition be given of 'common base' Pascal. Appended is a discussion paper and some comments. There are two questions which must be addressed:

1. What is 'standard' Pascal?
2. Should the 'common base' Pascal included extensions to the 'standard'?

2. Standard Pascal

Standard Pascal is, to oversimplify, either Jensen and Wirth Pascal or ISO Pascal. The two are not too far apart.

3. Extended Pascal

The main reasons for extending Pascal are to enhance its use for systems work and large programs. Extensions tend to be

- a. string manipulation features
- b. improved I/O facilities
- c. modularity, encapsulation
- d. separate compilation
- e. multi tasking, parallelism, monitors
- f. mixed language (eg C, Fortran) compatibility
- g. operating system interface (eg Unix system call).

Current extended Pascals include, for example:

- | | |
|-----------------------|----------------|
| a. UCSD Pascal | d. Pascal Plus |
| b. Concurrent Pascal. | e. Pascal-M |
| c. PERQ Pascal. | f. Pascal-VU |

4. Recommendation

The Software Technology Panel recommends that ISO Pascal be adopted as the 'standard'. Use of extensions should be discouraged.

The Software Technology Panel recommends that Pascal Plus be put forward as the candidate for 'common base' Pascal as it includes ISO Pascal as a subset. (An implementation of Pascal Plus for the PERQ is already under way - delivery summer 1982).

SOFTWARE TECHNOLOGY INITIATIVE

A Position Paper on Common Base Pascal

Clearly the Common Base must include compilers and related tools for Pascal programming, but in choosing these, consideration must be given to the variations in Pascal that now exist.

Current implementations of Pascal may be divided into three categories:

1) "Standard Implementations"

Until recently the only effective definition of Pascal was that given by Jensen and Wirth's User Manual and Report, or the operational definition given by the Zurich Portable P-compiler. While neither of these are ideal definitions of the language, the implementors of Pascal on the majority of major mainframes have achieved a remarkable coherence by this route. The same implementors have worked closely with the standardisation process that has produced the new ISO standard definition of Pascal, and these implementations are likely to conform closely to this standard even in advance of its formal ratification. The Pascal Validation Suite will shortly provide an effective means of testing the conformance of such compilers, and a model implementation demonstrating the techniques necessary for complete conformance is being developed at UMIST with NPL support. Thus by the time the Common Base is operational a well-defined Pascal standard will exist and be supported by most mainframe implementations. Since Common Base PERQS will be used as program preparation facilities for eventual mainframe computing, it seems imperative that Common Base Pascal conforms to the same standards as the mainframe implementations.

2) UCSD Pascal and its derivatives

Many of the Pascal implementations available on the mini- and micro-computers have been based on the highly successful UCSD Pascal System. Although the original UCSD system was also based on the Zurich P-machine, it incorporates several minor language variations. While its original implementors have given a commitment to conform to the ISO standard, it is not clear that the commercially available derived systems will follow this lead. However, it is also unclear that divergence between the common base Pascal and such implementations would have serious disadvantages.

In considering UCSD Pascal, it is important to realise that its undoubted popularity is not due to the variations in language features that it provides, but to its integrated work-station interface for program preparation, compilation and testing. As such, the UCSD model may be more relevant to consideration of programmers work-station interfaces than to consideration of Pascal as a language. The language variations that UCSD has introduced are minor and of debatable value when compared with those supported by "standard" implementations.

3) Extended Languages for Modular Multiprogramming

The languages Concurrent Pascal, Pascal-Plus, Modula and indeed Ada have extended the concepts of Pascal in directions orthogonal to its original design, to provide facilities for modular multiprogramming. Of these, Ada and perhaps Modula must be considered as languages in their own right, with Ada as an obvious future candidate for inclusion in the Common Base. However, Concurrent

Pascal and Pascal-Plus should be considered as extended versions of Pascal itself. In this respect, Pascal-Plus has the advantage that it is an exact superset of "standard" Pascal, and the portable compiler, the mainframe compilers developed in Belfast and the PERQ compiler now being developed at UMIST are all being aligned with the ISO standard, with a facility to reject extended language features on request as the standard requires.

In summary, I believe the following points are relevant in determining the Common Base Policy, re Pascal:

- 1) Mainframe compilers will quickly conform to the draft ISO standard now undergoing ratification. A requirement for similar conformance must be desirable for any Pascal compiler in the Common Base.
- 2) Extended language systems, such as Pascal-Plus, do not necessarily conflict with this standard policy, since both the standard definition and the Pascal-Plus implementations under development provide well-defined means of handling true extensions.

The existing PERQ Pascal compiler requires modification for conformance to the ISO standard, and I am not aware of any effort by ICL or by Three Rivers to do so. Three routes to a standard conforming compiler might be considered:

- 1) to commission the modification of the existing PERQ compiler;
- 2) to adapt an existing UNIX Pascal compiler, (such as the VU Amsterdam compiler), and ensure its conformance;
- 3) to adopt the PERQ Pascal-Plus compiler now being developed as the Common Base compiler for standard Pascal.

As the development of the Pascal-Plus compiler is already underway, and is being carried out in tandem with the development of the model standard Pascal compiler for NPL, route 3 seems likely to provide a standard Pascal compiler for the PERQ as soon as either of the other routes. Needless to say I favour its adoption.

Jim Welsh
10 January 1982