FROM THE EDITOR

The third newsletter was delayed by commitment of the editor to other activities at the University of Colorado. Similarly, the response time to correspondence and orders for Pascal has also been slow, at times as long as two months. Meanwhile, the following significant events regarding Pascal have transpired in the last eight months since newsletter number 2.

The Pascal User Manual and Report has been published by Springer-Verlag.

A questionnaire was mailed in September 1974 to all known Pascal users. The summary of the reponses is reported here.

The editor presented a paper at the VIM (CDC Users Group) Conference in October 1974 in Denver, Colorado on Pascal. Parts of that paper appear in this newsletter.

The new portable version of Pascal was released in November of 1974.

Recipients of PASCAL2 should receive UPDATE7 soon.

As always, items of interest for the newsletter can be sent to the editor for inclusion in the next issue. Write:

Mr. George H. Richmond
University of Colorado
Computing Center
3645 Marine Street
Boulder, Colorado 80302
USA

or phone:

(303) 492-6934

PASCAL USER MANUAL AND REPORT

The "Pascal User Manual and Report" by Kathleen Jensen and Niklaus Wirth has been published by Springer-Verlag. This report is Number 18 in the series "Lecture Notes in Computer Science". The price is $8.20 or DM 20,-. Springer-Verlag has offices at 175 Fifth Avenue, New York, NY 10010, USA; D-6900 Heidelberg 1, Neuenheimer Landstrasse 28-30, Germany; and D-1000 Berlin 33, Heidelberger Platz 3, Germany.

K. Jensen and N. Wirth
Lecture Notes in Computer Science No. 18

Pascal - User Manual and Report

Errata December 1974

Preface, line 24: minmum → minimum

page/line	(negative line numbers imply counting from the bottom)	
3/-10	is	are
3/-9	is	are
6/23	procedrue	procedure
12/-10	negetion	negation
14/14	subtrcct	subtract
17/1	(delete the quote ")	
20/5	assignement	assignment
31/23	label	labels
31/25	ccse	case
35/15	in not	is not
35/19	scclar	scalar
41/7	<, <=	<, >, <=
43/-11	small pox	smallpox
49	Replace last 3 lines ("Furthermore ...") by "Within the qualified statement S in with b do S the identifiers a and b now denote the components b.a and b.b respectively."	
57/24	procedure) or	procedure), or
62/-13	(delete the extra quote ")	
63/8	first := nil	first := nil
67/1	PROCEDURUS	PROCEDURES
67/-3	jl..jn	jl ... jn
71/-17	parcmeter	parameter
81/10	(move line left by 20 positions)	
84/26	(delete the extra quote ")	
92/18	(delete the comma after "12")	
97/13	procedrues	procedures
98/10	´debby	´debby ´;
101/-18	discriptors	descriptors
101/-5	(delete the extra period)	
102/23	B4	B1
106/16	either intege or	either integer or
109/-1	data	date
134/15	charccter	character
138/28	constrat	contrast
139/21	know	known
144/-8	skde2	side2
161/8	if no next	If no next
168/-1	world symbols	word symbols

PASCAL QUESTIONNAIRE

A questionnaire was mailed in September 1974 to people on the newsletter mailing list. Its purpose was twofold. First, it was an attempt to reduce the number of names on the list. About 70 people failed to respond and were dropped. The other purpose was to survey the Pascal community and report its profile.

A summary follows of the questionnaire. Individual responses cannot be given due to the large number of replies and specific requests by several individuals (mainly those associated with the computer industry) to keep their replies confidential.

Of 269 questionnaires mailed, 190 were returned. This is a response of 70.6 percent. No attempt was made to account for the effect of several people responding from the same site. Most replies were received in the period September through December 1974.

Do you have access to a Pascal compiler?

73.7 %	137/186	YES
26.3 %	49/186	NO

Which version of Pascal are you using?

41.6 %	47/113	Wirth, PASCAL2
25.7 %	29/113	Wirth, PASCAL1
9.7 %	11/113	Hartman, IBM 370
8.0 %	9/113	Nagel, DEC 10
6.2 %	7/113	Wirth, PASCAL0
5.3 %	6/113	Welsh, ICL 1900
4.4 %	5/113	Thibault, CII IRIS
3.5 %	4/113	Amble, Univac 1100
3.5 %	4/113	Burger, CDC 6000

How many Pascal jobs are run in a month?

15.0 %	12/80	Less than 10
31.3 %	25/80	To 100
28.8 %	23/80	To 500
11.3 %	9/80	To 1000
13.8 %	11/80	Over 1000

What is the nature of your installation?

73.7 %	112/152	Educational institution
11.8 %	18/152	Computer industry
9.9 %	15/152	Governmental facility
4.6 %	7/152	Business firm

What applications do you use Pascal for?

70.5 %	122/173	Educational
69.4 %	120/173	Software development
37.0 %	64/173	Scientific
6.9 %	12/173	Business

Several people reported having implemented their own version of Pascal. These implementations have not yet been distributed. Mr. Pierre Desjardins of the Université de Montréal is working on a Xerox Sigma 6 Pascal compiler. Monsieur Marcel Dupras of the Institut de Programmation of the Universite Pierre et Marie Curie in Paris, France has a CDC 3600 version of Pascal. There is also a CDC 3300 version of Pascal from Bratislava, Yugoslavia on which there is little information. Professor Donald B. Gillies at the University of Illinois (Urbana) has a PDP-11/20 Pascal compiler. Professor Doctor G. Goos at the Universitaet Karlsruhe in Germany has a B6700 version of Pascal. Monsieur Gerard Henneron of IREP in Grenoble, France has an IBM 370 Pascal compiler. Dr. J. Larmouth at Cambridge in England has a IBM 370 Pascal interpreter. Dr. Jan Madey of the Leibniz-Rechenzentrum der BAW in Munich, Germany has a Telefunken TR440 version of Pascal. Mr. David L. Russell and Mr. Jeffrey Y. Sue of the Digital Systems Laboratory of Stanford University have an IBM 370 Pascal compiler. And Mr. H. C. de Ruyter van Steveninck of Philips Research Laboratories in Eindhoven, The Netherlands has a version of Pascal for the Philips P1400.

HISTORY OF PASCAL, REVISED - G. Richmond

The Pascal Newsletter No. 2 contained a section on the history of Pascal. Since then, omissions were brought to the editor's attention and more information about the development of Pascal was discovered. The editor presented the revised history at the VIM 21 Conference in Denver last October and it is repeated here for the benefit of those who have not seen it before. Note in particular, the use of the terms PASCAL0, PASCAL1, PASCAL2, PASCAL-P1, and PASCAL-P2 to denote various versions of Pascal that have been released. A bibliography of material about Pascal is also included.

Pascal is an outgrowth of the early efforts of Dr. Niklaus Wirth to define a successor to Algol 60 by extension into the area of data definition facilities. In 1965, a proposal for such a successor was presented to an IFIP Working Group. It was deemed not to be a sufficient advance over Algol 60 and was released for wider publication and appeared in the Communications of the Association for Computing Machinery [Wir66]. Some of the aims of this proposal were to define a language that could be compiled quickly by a reliable translator, extend the utility of the language beyond strictly algebraic and numeric applications into areas such as information retrieval and symbol manipulation, and to oblige the programmer to express himself clearly without subverting the language to accomplish his task.

These goals reappear in the original definition of the language Pascal [Wir71a]. There was a desire to provide a language well suited to teaching programming as a systematic discipline based on fundamental concepts that were visibly reflected in the language. And Dr. Wirth wanted to demonstrate through an actual implementation that a significant and useful language can be realized in a reliable fast compiler.

The first work on Pascal began in 1968 at Eidgenössische Technische Hochschule (ETH) in Zürich, Switzerland on a CDC 6000 system with the construction of a Pascal compiler written in Fortran [Wir71b]. Although the compiler was completed, the result was an internal data structure and program contorted to fit the rules of Fortran. This compiler was unsuitable for translating at the source level into Pascal and a new compiler was started. It was entirely written in Pascal.

Once the project had progressed to the point that the paper compiler could theoretically translate itself, the compiler was translated by hand into a low level language and an executable compiler was produced. This hand translation only took one man-month of effort. Admittedly, only enough of the language facilities to allow self-compilation had been implemented, but 60 percent of the final version was already there. Almost 3000 lines of code were produced before testing could begin. The full compiler was available in 1970 and this version is called PASCAL0.

Further validation of the design of the language Pascal and of its implementation was provided one year later when Drs. J. Welsh and C. Quinn at Queen's University in Belfast, North Ireland transported Pascal from the CDC 6000 series computer to the ICL 1900 series computer [Wel72]. This project required six man-months of preparatory design and coding without access to an operational compiler. The major work involved the design and writing of expression recognizers to generate code for a radically different machine architecture. An additional man-month was devoted to writing an ICL machine code interpreter to run on the CDC machine. This tool would allow testing of the ICL compiler on the CDC machine. In a single trip to Zurich of eight working days, the ICL compiler was run for the first time and debugged. The binary of the ICL compiler was taken back to Belfast where it ran successfully.

The original report on Pascal provided a rigorous definition of the syntax of the language but only a verbal description of the semantics. The semantics were provided in 1972 by the publication of an axiomatic definition of Pascal [Hoa73]. As part of this project, the syntax of the language was revised slightly and the revised report on Pascal was issued [Wir72a]. To bring the existing compiler up to date, the necessary cosmetic changes, except for class variables which were retained from the previous version, were made and a new compiler was released. This version is refered to as PASCAL1 and was first available in late 1972. It is also known as PASCAL 6000-3.2.

However, the decision was made to rewrite the compiler completely. The changes included replacement of class variables with pointer variables to conform with the revised report, relocatable binary object code compatible with the standard CDC loader instead of a special absolute loader, introduction of a facility for separately compiled procedures and external Fortran subroutines, and new transput procedures for eliminating the need for a line control character and to accomodate CDC disk file structures. The new Pascal compiler was released in June 1974 and is called PASCAL2. It is also known as PASCAL 6000-3.4.

During the past two years, another method for implementing Pascal on computers other than the CDC 6000 series has been available from ETH. A portable Pascal compiler was released in early 1973 for this purpose. This first version is called PASCAL-P1. It was written to generate in an abstract language called P-code. By compiling the compiler, a P-code Pascal compiler is obtained. Thus, one only need provide an interpreter for P-code in order to obtain a Pascal compiler. However, the PASCAL-P1 compiler followed the original definition of Pascal (PASCAL0 compiler) and did not implement all the features available in the CDC 6000 version.

To remedy this problem, the portable Pascal compiler was rewritten and released in November 1974. It is called PASCAL-P2. The new compiler can be tailored to the target machine by specifying several parameters. Further details on this compiler can be found elsewhere in this newsletter.

With two methods of implementation possible, Pascal has been transfered to several different machines. The original bootstrap of the CDC version was done by Welsh and Quinn to an ICL 1900 machine [Wel72]. Since then, Mr. D. Thibault and Mr. P. Mancel in Paris, France bootstrapped Pascal to the CII Iris 80 [Thi73]. This machine has an instruction set equivalent to the Xerox Sigma 7. Another bootstrap is being attempted by Mr. P. Desjardins in Montreal, Quebec for the Xerox Sigma 6 [Des73]. Implementations via PASCAL-P1 were done by Dr. H. H. Nagel in Hamburg, Germany for the DEC System 10 [Gro74], by Mr. A. Hartmann at Caltech for the IBM 360 [Ric74b], and Dr. T. Amble in Trondheim, Norway for the Univac 1108 [Mol74]. Other IBM 360 Pascal projects are underway at the University of Manitoba by Dr. J. M. Wells using PL360 as the implementation language [Ric74b] and at Stanford University by Mr. D. Russell and Mr. J. Sue using the bootstrap technique [Rus74]. In a recent survey reported elsewhere in this newsletter, reponses indicated that projects were underway for the following machines: TI ASC, Data General 840, Raytheon 704, Univac AN/VYK-20, Honeywell G635, Burroughs 4700 and 6700, and PDP-11/45.

Several people have gone beyond just implementing Pascal on their computer. The best documented example is that of Dr. W. Burger [Bur73] who made extensive modifications to the compiler and supporting routines, particularly in the area of partial compilation. Control Data Corporation has modified Pascal to make it a cross-compiler. This version runs on the CDC 6000 series computer and produces binary code for their front-end processor.

After the development of Pascal in 1970 at ETH Zürich, Pascal spread through publication in journals, such as Acta Informatica [Wir71a] and Software-Practice and Experience [Wir71b]. Dr. Wirth spent the 1971-1972 academic year at Stanford University and gave several talks throughout the United States about Pascal. In 1972, Dr. Lyle Smith at the University of Colorado agreed to help Dr. Wirth distribute Pascal in the United States. In 1973, the editor took over this function after Dr. Smith left the University of Colorado. Other people with their own versions of Pascal have begun their own distribution mechanisms. Growth

of the Pascal community can be seen through the number of new CDC sites installing Pascal within the United States over the past several years.

1972	14
1973	22
1974	46

To this point in time, Pascal has been entirely supported by a small group at ETH in Zurich. Dr. Niklaus Wirth, Ms. Kathleen Jensen, Mr. Urs Ammann, Mr. H. H. Nägeli, and Mr. Helmut Sandmayr are the current members. In the past, Mr. E. Marmier, Mr. W. Bächi, and Mr. Rudy Schild have been associated with the project. The PASCALØ and PASCAL1 compilers are extremely stable with only 9 cards of corrective code (about Ø.1 % of the system) issued for them. The recently released PASCAL2 compiler (June 1974) has a poor initial history with 742 lines of corrective code (9.7 %) already released.

The mechanism for reporting problems is extremely informal. Evidence is mailed to Zürich and corrective code is later mailed out to all known Pascal sites. Unfortunately, none of the beneficial mechanisms of a formal reporting system like the CDC PSR Summaries exist. And no indications have been given as to what problems have been repaired by each corrective code release.

Since Pascal can be modified easily, many users have made extensive changes, particularly because of the spartan compilation listing produced by the compiler. Coordination of these individual efforts has not been attempted.

Documentation for Pascal is sparse, particularly concerning the internal construction of the compiler. The only published internal documentation on the compiler are the original papers on the design of the compiler [Wir71b] and on code generation [Wir72b].

Manuals for users of Pascal were at first limited to only the formal definition of the language [Wir71a,Wir72a]. In 1973, Dr. W. Burger wrote a Pascal Manual [Bur73] for his version of Pascal. This manual was revised to reflect PASCAL1 by the editor and distributed as part of the Pascal package. Just recently, a Pascal User Manual and Report by Ms. Jensen and Dr. Wirth [Jen74] was published and fulfills the need for a user document on Pascal (and PASCAL2).

Pascal is a good language for teaching programming [Lec74a] and the development of software [Wir71c, Wir73]. For CDC 6ØØØ computer users in particular, Pascal is a much more readable and versatile language that Algol or Fortran. It provides the control and data structures that are desirable for structured programming. With the PASCAL2 compiler supporting separate compilation, relocatable binary decks, and interfacing to Fortran subroutines, Pascal is now a viable language for programming on CDC computers. Its portablility to other computers gives it the potential to be as widely used a language as Fortran.

However, there are some problems with Pascal. First, the language has a number of minor deficiencies which were stated in a paper by Dr. Habermann [Hab73] and rebutted by Prof. Lecarme [Lec74b]. Although these are not serious problems, any further refinements to the language to solve them would just place another dialect of Pascal in the field. The problem of incompatible dialects is further compounded by the number of people willing to make significant changes to the compiler.

The second problem is a lack of support from any major computer manufacturer. Most implementation efforts have been in a university environment. However, manufacturer support may be undesirable as computer manufacturers have never been know as promoters of portable software. There is a need for a strong Pascal Users Group to carry on the task of developing, distributing, and supporting Pascal for all brands of computers. As the number of Pascal sites continues to grow, the present mechanism for distribution and support will become more inadequate.

However, the willingness of hundreds of sites to install Pascal and use it is proof of the merit of the language.

BIBLIOGRAPHY

The following bibliography includes material referenced above as well as references to material mailed to the editor.

Amm73 Ammann, U., "The Method of Structured Programming Applied to the Development of a Compiler", Proceedings of ACM International Computing Symposium, Davos, Switzerland (1973)

Bur73 Burger, W. F., "Pascal Manual", Department of Computer Science, TR-22, The University of Texas at Austin (1973)

Bro74 Bron, C., de Vries, W., "A Pascal Compiler for PDP11 Minicomputers", Department of Electrical Engineering, Twente University of Technology, Enschede, Netherlands (1974) (to appear in SOFTWARE-PRACTICE AND EXPERIENCE)

Des73 Desjardins, P., "A Pascal Compiler for the Xerox Sigma 6", SIGPLAN NOTICES 8, 6, pp. 34-36 (1973)

Fin74 Findlay, W., "The Performance of Pascal Programs on the MULTUM", Report No. 6, Computing Department, University of Glasgow, Scotland (July 1974)

Gro74 Grosse-Lindemann, C.-O., Nagel, H.-H., "Postlude to a Pascal-Compiler Bootstrap on a DECSystem-10", Bericht Nr. 11, Institut fur Informatik, Universitat Hamburg, Germany (1974)

Hab73 Habermann, A. N., "Critical Comments on the Programming Language Pascal", ACTA INFORMATICA 3, 1, pp. 45-57 (1973)

Hei73 Heistad, E., "Pascal - Cyber Version", Teknisk Notat S-305 Forsvarets Forskningsinstitutt, Norwegian Defense Research Establishment, Kjeller, Norway (June 1973)

Hoa73 Hoare, C. A. R., Wirth, N., "An Axiomatic Definition of the Programming Language Pascal", ACTA INFORMATICA 3, pp. 335-355 (1973)

Jen74 Jensen, K., Wirth, N., "Pascal User Manual and Report", Lecture Notes in Computer Science, Vol. 18, Springer-Verlag, New York (1974)

Kno74 Knobe, B., Yuval, G., "Making a Compiler Indent", Computer Science Department, The Hebrew University of Jerusalem, Israel (November 1974)

Kir73 Kristensen, B. B., Madsen, O. L., Jensen, B. B., Eriksen, S. H., "A Short Description of a Translator Writing System (BOBS-System)", Daimi PB-11, University of Aarhus, Denmark (February 1973)

Kri74a Kristensen, B. B., Madsen, O. L., Jensen, B. B., "A Pascal Environment Machine (P-code)", Daimi PB-28, University of Aarhus, Denmark (April 1974)

Kri74b Kristensen, B. B., Madsen, O. L., Jensen, B. B., Eriksen, S. H., "User Manual for the BOBS-System", Unpublished English Version, University of Aarhus, Denmark (April 1974)

Lec74a Lecarme, O., "Structured Programming, Programming Teaching, and the Language Pascal", SIGPLAN NOTICES 9, 7, pp. 15-21 (July 1974)

Lec74b Lecarme, O., Desjardins, P., "Reply to a Paper by A. N. Habermann on the Programming Language Pascal", SIGPLAN NOTICES 9, 10, pp. 21-27 (October 1974)

Mol74 Mølster, T., Sundvor, V., "Unit Pascal System for the Univac 1108 Computer", Teknisk Notat 1/74, Institutt for Databehandling, Universitetet I Tronheim, Norway (February 1974)

Nor74 Nori, K. V., Ammann, U., Jensen, K., Nägeli, H. H., "The Pascal(P) Compiler: Implementation Notes", No. 10, Berichte des Instituts für Informatik, Eidgenössische Technische Hochschule, Zürich (December 1974)

Ric74a Richmond, G., "Pascal Newsletter", No. 1, University of Colorado Computing Center, Boulder (January 1974)

Ric74b Richmond, G., "Pascal Newsletter", No. 2, University of Colorado Computing Center, Boulder (May 1974)

Rus74 Russell, D., Sue, J., Digital Systems Laboratory, Stanford, see Letters to the Editor (~~September 1974~~) (this issue)

Sol74 Solntseff, N., "McMaster Modifications to the Pascal 6000 3.4 System", Computer Science Technical Note 74-CS-2, McMaster University, Ontario, Canada (November 1974)

Thi73 Thibault, D., Mancel, P., "Implementation of a Pascal Compiler for the CII Iris 80 Computer", SIGPLAN NOTICES 8, 6, pp. 89-90 (1973)

Wel72 Welsh, J., Quinn, C., "A Pascal Compiler for the ICL 1900 Series Computer", SOFTWARE-PRACTICE AND EXPERIENCE 2, 1, pp. 73-77 (1972)

Wir66 Wirth, N., Hoare, C. A. R., "A Contribution to the Development of Algol", COMMUNICATIONS OF THE ACM 9, 6, pp. 413-432 (1966)

Wir71a Wirth, N., "The Programming Language Pascal", ACTA INFORMATICA 1, 1, pp. 35-63 (1971)

Wir71b Wirth, N., "The Design of a Pascal Compiler", SOFTWARE-PRACTICE AND EXPERIENCE 1, 4, pp. 309-333 (1971)

Wir71c Wirth, N., "Program Development by Step-Wise Refinement", COMMUNICATIONS OF THE ACM 14, 4, pp. 221-227 (April 1971)

Wir72a Wirth, N., "The Programming Language Pascal (Revised Report)", No. 5, Berichte der Fachgruppe Computer-Wissenschaften, Eidgenössische Technische Hochschule, Zürich (November 1972)

Wir72b Wirth, N., "On Pascal, Code Generation, and the CDC 6400 Computer", Computer Science Department, STAN-CS-72-257, Stanford University (1972) (out of print, Clearinghouse stock no. PB208519)

Wir73 Wirth, N., "Systematic Programming: An Introduction", Prentice-Hall, Englewood Cliffs (1973)

Wir74 Wirth, N., "On the Composition of Well-Structured Programs", COMPUTING SURVEYS 6, 4, pp. 247-260 (December 1974)

PORTABLE PASCAL

In November 1974, Mr. H. H. Nägeli at Eidgenössiche Technische Hochshule in Zürich released the updated version of the portable Pascal compiler called PASCAL-P2. This compiler complies with Standard Pascal and produces P-code, an assembly language for an abstract machine, as output. By implementing a P-code interpreter on any machine, one can obtain a quick implementation of Pascal on that machine. Both the source and object (P-code) version of the compiler are supplied in the implementation package. The code generators of the portable compiler

can be rewritten as the second step of the bootstrap process to obtain a true compiler for the target machine.

Portable Pascal must be configured for the target machine by specifying the maximum address available for program or data, the number of machine addressable units (words or bytes) necessary for each basic data type (integer, real, character, boolean, set, and pointer), the maximum length of a string, and the number of bits available to represent an integer. These numbers are substituted into an unconfigured version of the portable compiler. For sites without access to a PASCAL2 compiler, the distributor must compile the compiler and send a P-code version of the portable compiler. For sites with access to a PASCAL2 compiler, only the unconfigured version need be sent.

As the cost of obtaining portable Pascal is dependent on exactly what steps the distributor must take to prepare the compiler for the implementor, one should write and request an order form from one of the two distribution points before placing an order. In Europe, write:

Mr. H. H. Nägeli
Institut für Informatik
Eidgenössiche Technische Hochschule
Clausiusstrasse 55
CH-8006 Zürich
Switzerland

In North America, write:

Mr. George H. Richmond
University of Colorado
Computing Center
3645 Marine Street
Boulder, Colorado 80302
USA

A GENERALIZATION OF THE READ AND WRITE PROCEDURES - N. Wirth

Update No. 7 of the PASCAL 6000-3.4 compiler generalizes the read and write procedures in an important way. In Standard Pascal, the basic file operations are get(f) and put(f) , which copy the next element from a file f to its buffer variable f↑ (or vice versa) and advance the file position. In practice, the operations of advancing the position and of accessing the file buffer frequently occur as "action clusters", and it seems convenient to express them by one procedure.

Assume a variable x of type T and a file variable f of type file of T . Then we introduce the procedures

```
read(f,x)        as synonymous to

                 x := f↑; get(f)                    and
```

write(f,x) as synonymous to

```
f↑ := x; put(f)
```

Note that these facilities are already available in Pascal, but only for type T = char . The new update makes them applicable to any type T (except if T is a file itself).

The advantage of this facility is not only brevity, but also conceptual simplicity. For, it is now possible in most cases to ignore the existence of the buffer variable f↑ , whose value is sometimes undefined. From the definitions of get and put it follows immediately that write(f,x) can only be called, if eof(f) is true, and read(f,x) only if eof(f) is false. This condition is safely guaranteed by the use of the following simple schema for reading and sequential processing of a file f :

```
reset(f);
while ~eof(f) do
   begin read(f,x); process(x)
   end
```

Upon reading the last element of f , the condition eof(f) becomes true. If the file is empty, it becomes true through the call of reset(f) .

The generalized forms of read and write do, of course, not imply elimination of the basic operators and the buffer variable. The latter may in many applications be useful as a lookahead facility.

CORRECTIONS TO PASCAL2

These corrections have been distributed on paper as they have become available. Occasionally, the quality of printing of the copy from the University of Colorado has been poor. A new printer of much higher quality is now available at the Computing Center and persons wishing to have a better copy of the corrections may simply request it. A copy of the corrections is also available on magnetic tape if the tape is supplied with the request. Minitapes are adequate for this purpose.

Correction	Date of Release
UPDATE1	1 July 1974
UPDATE21	1 September 1974
UPDATE22	1 September 1974
UPDATE3	20 September 1974
UPDATE4	24 October 1974
UPDATE5	1 November 1974
UPDATE6	1 December 1974
UPDATE7	1 February 1975

PASCAL2 AND INTERACTIVE OPERATION

Many people have complained that PASCAL2 programs cannot be run interactively from a timesharing terminal. This problem is caused by the read routine that looks ahead on the INPUT file. Several people have submitted code which modifies the environmental support routines to solve the problem but these modifications tend to be very operating system dependent. The following sample program illustrates how to do interactive transput by avoiding the predefined file variables INPUT and OUTPUT. To succeed, the transput files must be segmented.

```
PROGRAM SAMPLE(OUTPUT,TTYIN,TTYOUT);
VAR TTYIN,TTYOUT : SEGMENTED FILE OF CHAR;
BEGIN
  RESET(TTYIN);
  REWRITE(TTYOUT);
  WHILE NOT EOF(TTYIN) DO BEGIN
    WHILE NOT EOS(TTYIN) DO BEGIN
      WHILE NOT EOLN(TTYIN) DO BEGIN
        TTYOUT↑ := TTYIN↑;
        PUT(TTYOUT);
        GET(TTYIN) END;
      WRITELN(TTYOUT);
      GET(TTYIN) END;
    PUTSEG(TTYOUT);
    GETSEG(TTYIN) END
END.
```

LETTERS TO THE EDITOR

Relevant extracts of letters addressed to the editor are included below. A considerable amount of corrective code for PASCAL2 has been received from the community and has been forwarded to ETH Zürich. It is much too voluminous to quote here.

10 June 1974

The department has access to a CDC 6400 Computer. A version of a Pascal Compiler is available, which corresponds to the second edition of the report "The Programming Language Pascal" by N. Wirth [Wir72a]. The new version of the Pascal Compiler from Zürich, containing different changes/extensions will become available too.

Our projects connected to the Pascal language are the following.

A LALR(1) parser generator system (BOBS-system) has been developed. It is programmed in Pascal but unforunately it depends on the earlier mentioned version of the compiler. A tape containing the system has

just now been produced together with an unpublished necessary English version of the user manual for the system [Kri74b]. We enclose this user manual as well as a short description of the BOBS-system [Kri73].

Furthermore a Pascal Compiler, translating the earlier mentioned second edition of Pascal to an intermediate language (a hypothetical stack machine) has now been finished. The Pascal Compiler is written in Pascal using BOBS-system, thereby becoming very applicable for changes and extensions in the future.

In connection to the stack machine, which we have designed ourselves and unluckily also named P-code, an interpreter on the CDC 6400 has been programmed in Pascal and a P-code emulator has been microprogrammed on a minicomputer system which is developed at our department and is being built and tested at the moment.

The Pascal environment machine (our P-code) is described in detail in the enclosed report [Kri74a]. As described here neither the constructed compiler nor the P-code machine is updated to the latest revised version of the Pascal language. However, it is our intention to perform this extension as soon as possible and at the same time to publish a revised description of the compiler/simulator/emulator system.

Mr. Bent Bruun Kristensen
Institute of Mathematics
University of Aarhus
Department of Computer Science
Ny Munkegade
8000 Aarhus C
Denmark

19 June 1974

A complete Pascal compiler for Univac 1100 series machines is being developed at The University of Copenhagen. The implementation is based on the PASCAL-P compiler from which we bootstrap. In July 1974 we will be able to compile the PASCAL-P compiler to symbolic Univac 1100 assembler code. The generated code has been checked by inspection for all language constructs supported and we find it more compact and efficient that corresponding code from any of the Algol compilers available (Univac ALGOL and NUALGOL). The July 74 version does not implement the file concept except the standard file operations READ and WRITE on INPUT and OUTPUT respectively. Packed structures are implemented using the partial word addressing of Univac 1100 machines. Formal procedures and functions will be implemented (not yet debugged) but the user must specify their parameters.

The next step in the bootstrap process will of course implement the file concept. It should then be a minor modification to generate

relocatable code (provided format specification in certain system procedure calls!). We have also plans for a version generating code in core ready for execution.

Mr. J. Steensgaard-Madsen
Datalogisk Institut, University of Copenhagen
Sigurdsgade 41
DK-2200 Copenhagen
Denmark

17 July 1974

A Pascal compiler for ICL 1900 series computers was constructed at Queen's University Belfast in 1971, as reported in [Wel72]. The compiler has so far been released to some 20 other ICL 1900 installations in the U.K., eastern Europe and Australia.

In Belfast the compiler has been used for teaching, experiments with language extensions, and further software development. The latter includes an interpretive implementation of a Pascal subset for introductory programming courses. The compiler itself is now being completely rewritten to bring it into line with Standard Pascal, and to further improve its clarity and structure. This version should be available for release by January 1975.

Dr. J. Welsh
Department of Computer Science
Queen's University
Belfast BT7 1NN
N. Ireland

6 August 1974

Pascal is now available on almost all the major machines. Considerable experience has also been gained in the use of Pascal for a variety of applications. I believe therefore that the time has come to take another look at Pascal and begin the process of standardizing the language. Existing versions of Pascal are not entirely compatible. Implementors on various machines e.g., PDP-10, CDC, PDP-11 have added minor changes which they have obviously found useful. There also exists at least two dialects of Pascal, vis BLAISE and SUE, which have also incorporated significant changes. Since one of the aims was to develop a portable language, I believe a delay in standardization, would only encourage proliferation of various dialects and changes and would nullify the impact and usefulness of PASCAL.

I would like to hear the views of other users of Pascal on this subject. Please write directly to the me or to the newsletter.

Mr. George Poonen
15 Orchard Avenue
Waban, Massachusetts Ø2168

21 August 1974

The PASCAL/3:4 system as distributed by ETH is incompatible with segmentation option of the SCOPE 3.4 loader. The problem occurs in P.INIT where the field length is taken from register AØ. The following update fixes P.INIT so that the field length will in fact be in AØ. The update is based on minimum knowledge of both P.INIT and the MEMORY macro so more efficient code is undoubtedly buildable.

```
*IDENT,MEMORY
*INSERT,SYS.714
*  CHANGES ARE TO HAVE B4 POINT TO TOP OF MEMORY,
*  EVEN AFTER SEGLOAD.
*INSERT,SYS.716
          SX7    B7
          SA7    TEMPB7          STORE B7
          MEMORY CM,STATUS,RECALL
          SA1    TEMPB7          RESTORE B7
          SA2    STATUS          FL TO AØ
          SB7    X1
          AX2    3Ø              STATUS HAS FL IN TOP HALF
          SAØ    X2              AØ = FL
*INSERT,SYS.761
 STATUS   DATA   Ø
 TEMPB7   BSS    1
```

Dr. G. Yuval and Dr. B. Knobe
The Hebrew University of Jerusalem
Institute of Mathematics
Department of Computer Science
Jerusalem, Israel

5 September 1974

We have completed a Pascal compiler for the (unique and now defunct) MULTUM computer, a 16-bit "mini" in the same class as the PDP 11/45. I enclose a report on the performance of this system [Fin74].

We are now contemplating a new project for a relative portable and adaptable compiler which will accept a language containing Standard Pascal. Initially this would be implemented on the ICL 1900 series computers, and would be configured for a "student" workload. We are currently debating the whole issue of standardization and extension of Pascal. When we resolve our ideas they will be written-up as a project working paper.

Mr. William Findlay
University of Glasgow
Computing Science Department
Glasgow, G12 8QQ
Scotland

13 September 1974

We have recently completed the bootstrap of a Pascal compiler written in Pascal on the IBM 360/370. The language we implemented was a minimal subset of Standard Pascal, just enough to allow the compiler to compile itself. We have omitted procedures and functions as formal arguments, arithmetic procedures, powersets, and packed records. Some features which were not used in the compiler have not been debugged. Our implementation was based on the CDC 6600 compiler of January 1972.

Our compiler generates OS object modules with purely reentrant code, and allows separate compilation of global procedures. It runs under a rudimentary assembly language monitor, and supports record I/O only (no character I/O). It compiles itself in a region size of 192 kilobytes, of which about 106 kilobytes is program. The compilation, with source listing, takes about 16 seconds of CPU time on an IBM 370/168.

We would like to stress that no more work on the compiler is planned at this time. We are in the process of preparing documentation on the system as it currently exists.

Mr. David L. Russell and Mr. Jeffrey Y. Sue
Digital Systems Laboratory
Stanford University
Stanford, California 94305

24 October 1974

A group consisting of Dr. S. V. Rangaswamy, Mr. N. Chakrapani and Dr. V. G. Tikekar is working towards disseminating Pascal language and implementing a rich subset of it on System 360/44. A few graduate

students have chosen projects, towards their Master´s degrees, related to this implementation under the supervision of the group. Features like file structures, exit labels, functions etc., are not included in this version. Towards this end, the hypothetical stack computer formulated by Dr. Wirth´s Group has been simulated on the System 360 by Mr. K. S. Natarajan (currently at the Department of Computer and Information Science at the Ohio State University, Columbus, Ohio as a doctoral student) as his project. Mr. N. Natarajan (currently at the National Center for Software Development and Computing Techniques, TIFR, Bombay) in his project has designed an interpreter (written in Pascal) using the recursive descent technique to interpret Pascal programs and to produce the machine code of the stack computer. Mr. M. K. Srivas is currently working on the implementation of a Pascal compiler (written in PL/1 on System 360).

This Group is thankful to Dr. N. Wirth and his group at ETH, Zurich, for providing information about Pascal and the details of the stack computer and to Mr. K. V. Nori of the Computer Group, TIFR, BOMBAY for invaluable discussions.

For any further information, kindly write to:

Dr. S. V. Rangaswamy
School of Automation
Indian Institute of Science
Bangalore 560 012
India

1 November 1974

Sometime ago we received a version of the PASCAL-P compiler, written in Pascal, generating code for the hypothetical Pascal machine.

First the interpreter was rewritten in ALGOL-W. There were some problems with incompatibility of line structure of Pascal transput and the rather rigid I/O format of ALGOL-W. These problems were solved by linking the ALGOL-W program to a number of Fortran based routines.

The interpreter did become operational. It was too slow, however to realistically perform a job like compilation of the compiler. Furthermore we remained with an unsolved problem concerning a requirement of too many blanks in the source input.

We then discontinued the line sketched above, in part influenced by the knowledge of a DEC 10 to replace the 360/50 by next fall. Furthermore there is an urgent need of a possibility to generate code (preferably from higher level languages) in order to feed several of the PDP 11´s that are operational in a variety of Institutes at our University. We are now working on a modified version of the PASCAL-P

compiler to generate code for a PDP 11. It is our intention to have this compiler operational on the Central Computing Facility (DEC 10) and to inject the generated code via data-links into a PDP 11 upon request.

Our choice for Pascal as a starting point was based on two observations: 1) the availability of a "readable" compiler of which a large part, including the concept of the virtual P-machine, could be used; and 2) the possibility of generating efficient and compact code for the target-machine. (We can surely never expect our PDP 11-Assembler programmers to switch to a high level language if its code has the efficiency of the usual Algol-implementations).

Drs. C. Bron
Associate Professor of Computer Science
Technical University of Twente
Enschede, Netherlands