THE UNIVERSITY OF ARIZONA

TUCSON, ARIZONA 85721

DEPARTMENT OF COMPUTER SCIENCE

Icon Newsletter #19

Madge T. Griswold and Ralph E. Griswold

September 25, 1985

1. Implementation News

Version 5.9 of Icon for MS-DOS Systems

An implementation of Version 5.9 of Icon for MS-DOS has been completed by Cheyenne Wills of Mechanicsburg, Pennsylvania. He has placed it in the public domain to make it as widely available as possible.

This implementation requires MS- or PC-DOS Version 2.0 or higher and has been tested on a number of computers, including the IBM PC, XT, and AT. It should run on any computer with an 8086/88/186/286-family processor; IBM hardware compatibility is not required.

This is a small memory model implementation, using only two segments. As a result, the amount of space for Icon programs and data is limited. Nevertheless, many useful programs run quite well and this implementation offers many persons the opportunity to use Icon who otherwise could not.

A large memory model implementation, which will considerably extend the usefulness of Icon on personal computers, is in progress.

We are distributing copies of the small memory model implementation for the cost of media, handling, and shipping. To obtain a copy, use the request form at the end of this Newsletter.

Version 5.9 of Icon for the WICAT

Gary E. Sarff has completed an implementation of Version 5.9 of Icon for the WICAT running under MCS. It runs on the WICAT System 200 and the small desktop model WICAT System 150. He will provide copies to other WICAT users on 9-track tape or 5-¼″ System 150 floppies. To obtain more information or a copy of his implementation, contact him directly:

Mr. Gary E. Sarff
Hood Center
Box E
Norris City, IL 62869

Version 5.10 of Icon for UNIX Systems

Version 5.10 of Icon for UNIX* Systems is now available. Its features include:

- An implementation for the Ridge 32, done by Janalee O'Bagy at the University of Arizona. Other supported implementations include the AT&T 3B20, the PDP-11, the Sun Workstation, and the VAX-11.
- New options for producing sorted lists from tables that require less storage than the standard options.
- The availability of the personalized interpreter facility for all the supported implementations.
- Improved performance and the correction of a number of bugs.
- Reorganization of the implementation to facilitate porting.

To obtain a copy of this system, use the request form at the end of this Newsletter.

Version 5.10 of Icon for the UNIX-PC

Version 5.10 of Icon has been successfully ported to the AT&T UNIX-PC. The executable files soon will be available to AT&T employees via the UNIX-PC "store" bulletin board. Others can obtain a copy by sending a formatted 5-¼" 2S/2D diskette and a self-addressed, stamped diskette mailer to:

O. R. Fonorow
IW 1Z-261
1100 E. Warrenville Road
Naperville, IL 60566

The IIT Implementation of Icon

The implementation of Icon being done by Tom Christopher and his associates at the Illinois Institute of Technology is now running. This implementation generates VAX assembly-language code and uses a new model for implementing Icon expression evaluation. Initial tests indicate very fast execution of control operations.

The IIT group hopes to have an initial release available for distribution by the end of the year.

2. Implementation Book

We are in the process of writing a book on the implementation of Icon. This book is intended as a complement to books on compiler writing and it emphasizes the run-time system.

The tentative table of contents is:

*UNIX is a trademark of AT&T Bell Laboratories.

3. Contribution from a User

Professor John Slimick at the University of Pittsburgh at Bradford sends in these remarks regarding his experience with teaching Icon:

> I chose to use Icon as one of the two featured languages in our upper-division Programming Languages course last fall (LISP was the other). I spent slightly more than ten weeks on Icon, and covered the text through Chapter 15. LISP got rather short shrift, being bracketed by Thanksgiving and semester end, plus a carryover of the last Icon project.
>
> As the "at" in the name of the school implies, we are remote from the large, high performance campus in Pittsburgh. Our students view the BS as their terminal degree, so we are more oriented to software engineering than preparation for graduate work in computer science. All of the students had had two semesters of Pascal plus Data Structures; most had had either VAX Assembler or two semesters of COBOL.
>
> The general response was very positive. The shift from the restricted data types in Pascal to the rich set in Icon was made more easily than I had anticipated. Most had little difficulty with the extended operators, the "everything-has-a-value" philosophy, and new concepts like break, next, return, and suspend. I expected, but did not observe, problems with handling the "failure mode" concept; by the end of the Icon portion most were handling "fail" as though they had been using it all the time. One of the measures of success in such an undertaking is what the undergraduates suggest should be done with the course material. Two or three suggested that we dump Pascal and use Icon in the first courses; no one suggested that we abandon Icon. A few others used Icon in projects, either late in fall semester or in the winter semester. All the students commented on the "friendliness" of the text as opposed to either VMS manuals or their other textbooks.
>
> On the other hand, I lost the class towards the end, and I can remember exactly when it occurred: Section 14.6. The association of backtracking with the eight queens problem put almost everyone on "overload". Reducing the

problem to a four queens or some other simpler problem would have helped. I did manage to struggle through the early parts of Chapter 15, but LISP was looming and it looked like it was going to take a little while to get a good understanding of both backtracking and string scanning across. There had been some trouble spots, primarily in the area of syntax; one student unerringly used the syntax

```
while <expr>
{ <expr> ...
```

and wondered why the programs always hung. Another proved that one could nest ifs as deeply in Icon as anything else. A third, recalling his structured programming, usually had a three-line "main" procedure, which at once called exactly one other procedure, and that one sixty lines long. A fourth found that using lists of tables tended to exhaust free storage relatively quickly. There may not be any one "right" way to write an Icon program, but it seems there are innumerable "wrong" ways.

As most other correspondents have observed, I found the exercises in the text to be on the thin side. I tended to work them on the board and assign similar ones as homework. As one might expect, for programming problems I assigned a cross-reference generator (but with a finite set of noise words to eliminate) and a reading level estimate (based in part on vowel counting, but a diphthong counts only as one vowel); a more creative assignment was to extend Problem 4.4 to read in the following pattern:

```
/\        /\
/ \_______/ \
|           |
\   X    X  /
\           /
 |    "    |
  |  """  |
   \     /
    \___/
```

and generate an Icon program that prints out exactly the same set of characters (and then compile and run the program — easy to check!); this exercise allowed the Icon students to proclaim that "at UPB we don't write programs--we write programs that write programs!", which helped keep the morale high. The final assignment was, given an "animator" in the spirit of TR 83-14a that featured the upto function, extend this animation to many, and then to bal (source of much confusion).

In conclusion, I feel that I made the right decision in offering Icon where I did, and I am looking forward to offering it again this fall.

4. New Documents

Two technical reports related to Icon have been published recently:

- A bibliography of documents related to SNOBOL4, SL5, and Icon.
- A description of the features of Version 5.10 of Icon.

Copies of these reports are available, free of charge. Use the document request form that follows.

Request for Icon Documents

Please send the documents checked below to:

____

____

____

____

____

☐ *Bibliography of the SNOBOL, SL5, and Icon Programming Languages*, TR 85-13.

☐ *Version 5.10 of Icon*, TR 85-16. (*Note:* A copy of this document is included with the Version 5.10 system.)

Please add my name to the Icon mailing list.

Return this form to:

Icon Project
Department of Computer Science
The University of Arizona
Tucson, AZ 85721
U.S.A.

Request for Version 5.9 of Icon for MS-DOS

Note: This implementation of Icon runs on the IBM PC, XT, AT, and other 8086/88/186/286 family computers. IBM hardware equivalence is not required; only DOS compatibility is needed: PC- or MS-DOS Version 2.0 or above.

This system is distributed on a 5-¼" 2S/2D diskette. Enclose a check for $15 payable to the University of Arizona to cover the costs of media, handling, and shipping.

Ship to:

name	______________________
address	______________________
	______________________
	______________________
	______________________
telephone	______________________
electronic mail address	______________________

Return this form with payment to:

Icon Project
Department of Computer Science
The University of Arizona
Tucson, AZ 85721

Request for Version 5.10 of Icon for UNIX

Version 5.10 of Icon for UNIX can be configured for the following computers:

AT&T 3B20
PDP-11 (models with separate I and D spaces)
Ridge 32
Sun Workstation
VAX-11

Note: The distribution package contains a brief description of Icon, installation instructions, and supporting documentation. Documentation regarding the porting of Icon to other computers is optional and is supplied only if requested below.

Contact Information:

name ______________________________

address ______________________________

____

____

____

telephone ______________________________

electronic mail address ______________________________

computers ______________________________

operating systems ______________________________

Tapes are written in 9-track *cpio* or *tar* format. Specify preferred format and tape recording density:

☐ cpio ☐ tar

☐ 6250 bpi ☐ 1600 bpi ☐ 800 bpi

☐ Enclose documentation on porting Icon to other computers.

Send this form, together with a magnetic tape (600′ is sufficient) *or* a check for $20 payable to the University of Arizona, to:

Icon Project
Department of Computer Science
The University of Arizona
Tucson, AZ 85721