

PASCAL Notes

PASCAL
=====

by Rory O'Farrell

Notes on the steps towards implementation of a "Tiny" Pascal for Nascom.

The best known of all Tiny Pascal compilers is that published in BYTE, Sept./Oct./Nov 1978, by Chung and Yuen. This is designed to be implemented on a North Star with floppies, and is written in Basic! (Ugh!). It can however be made to run on a Nascom with 8k Basic. The main changes involve the Functions - North Star Basic allows multiple line functions, Nascom does not, but the change is easy. In the original, FNG (x,y,z,) is defined with a multiple line definition - for Nascom load three variables not used elsewhere, say e.g. G1, G2, G3, then GOSUB the line where you have repeated the body of the FNG (x,y,z,) definition (Using G1, G2, G3, as the variables). Each call to the function is replaced by a line consisting of G1=value 1: G2=value 2 G3=value 3: GOSUB LINENO. This is quite effective in the case of FNG, the code generation function, and can be done on the fly, as you enter the program. The two error functions can be handled similarly. The notes in the back of the Nascom Basic Manual on string format conversions between one system and another are most useful, as North Star uses a different convention for MID\$ commands. The program as published in Byte can be fitted in less than 20K - less than 16K if comments are deleted.

The Basic program as published in Byte produces pseudo (p) codes for an ideal machine. Unfortunately Byte no longer make available the Pcode to 8080 conversion program, so one will have to be written. Then to compile a Pascal program there will be two stages - use the 16/20K basic program to produce a Pcode listing, and then use a Pcode to 8080/Z80 translator to produce machine code. This machine code will have a number of calls to runtime support routines (perhaps in Eprom) for functions such as multiply and divide., but Chung and Yuen state that their support package amounted to about 1K, so it should not prove an insurmountable task to write.

This combination of Basic/Machine language programs will not be the quickest way to compile a Pascal program, but it is possible (I hope) and may allow a Pascal compiler written in Pascal to be compiled. For the Pascal compiler written in Pascal, I would direct attention to a new book "Structured System Programming" by Welsh and McKeag of Queens University of Belfast (Prentice Hall International). This contains within it a very nice Pascal compiler, written in Pascal, to produce what they call MINI Pascal. It looks very nicely written, and is modular in form, so that the code generation section which generates code for a 24 bit representation of an ICL 1900 could be rewritten for a Z80 (I'd be interested to hear from anyone who has done/is prepared to do this). Unfortunately, this Mini Pascal compiler uses some Pascal constructs not implemented in the Tiny Pascal, so a certain amount of extension of that will be necessary, before the Mini Pascal can be compiled. A format control program in Pascal was given in PCW April 1980, and might be of use in the editor section of a Mini Pascal compiler. To get the vast amount of source program through the machine may create a problem in memory management, but a little bit of thought should overcome that difficulty - (auto tape start/stop?)

I would be interested to correspond with anyone else working along these lines.

(Ed: We have seen a Pascal for the Nascom advertised and we hope to get a copy for review shortly.)