Blue Label Pascal for Polydos

Introduction

There were two options for writing programs under standard Polydos; Assembly Language and the Microsoft Basic interpreter. Assembly is slow to get results, tricky to understand others' work, but runs fast. Basic uses line numbers, which is clunky and runs slowly. Various Forth tools were available from third parties, programs ran reasonably fast, but not as fast as Blue Label Pascal. Not so many people like the Forth syntax. (I quite like Forth)

Blue Label Pascal, written by Anders Hejlsberg, was the protoype of the popular Turbo Pascal. Both were 'rapid application development' tools and a great deal of fun to use. I was quite productive with BL Pascal, compared to Forth, Assembler or Basic.

I always regretted that Blue Label Pascal was a cassette tape based application and never adapted for disk storage.

As a wintertime project, I adapted Blue Label Pascal v1.3, operating under the Polydos operating system, to:-

1. Store and load source files from disk
2. Store compiled code to disk
3. Execute compiled code stored on disk by typing the filename at the Polydos prompt. (Like a .BS file can be run, with the interpreter extension being loaded automatically)

This document provides guidance on:-

1. Installation
2. New and changed Pascal commands
3. Running a compiled program
4. Memory usage

Installation

The software may be obtained from my github page here.

The original BL Pascal handbook for v1.2 is available here.

The files to install:-

1. DPASCAL.GO is the BL Pascal development program. It is based on BL Pascal version 1.3.

   This file must be stored on the Polydos master (boot) disk or alternatively any disk where you are going to store compiled Pascal programs. The file's load address must be set to 1000H and the execution address set to 1000H. Warm start is 1003H.

2. PAfh.OV provides the new Polydos disk interface. It must be stored on the Polydos master (boot) disk. The file's load address must be set to C800H and the execution address set to 0000H.

New Commands

Two new commands have been introduced:-

The Disk command

Purpose:

Select disk or cassette storage

From start-up BL Pascal will use disk storage for source and compiled code. The Disk or D command allows the user to revert to Tape storage if he so wishes. On entering D <enter> the user is prompted:-

Select storage method:-

Disk [•]

Tape [ ]

Press D or T then ENTER to return to the main menu

So pressing T <enter> will cause Pascal to revert to being it's original tape based program. This will persist until the program is closed or the user enters the D option again. On restarting the program will be back in Disk mode.

The Nogo command

Purpose:

Set the highest memory address the Pascal system is allowed to use

From start-up BL Pascal for Polydos will use all RAM from 0C82 – BFFFH (just short of Polydos workspace). The user can change the upper limit with the Nogo or N command. On entering N <enter> the user is prompted:-

Enter highest address Pascal may use:

The user might enter: **B000** <enter>

and the program will reply:-

New value set

The new value defines the address at which a nogo area starts, which may be used for assembly language etc. safe in the knowledge that BL Pascal will not encroach on it.

Changed commands

The following three commands take different parameters when BL Pascal is in Disk mode. When in Tape mode, the three commands behave as described in the original BL Pascal handbook.

The Load command

Purpose:

Load source file from disk

Syntax:

L <filename> <enter>

If the file is found, it is loaded into BL the Pascal source code workspace, else an error messsage will be displayed. Multiple files may be loaded in turn and they will be appended one after the other. This is useful for loading libraries of reusable code into project.

If the filename does not include a drive number, drive 0 is assumed.

If the filename does not include a file type suffix, .TX is assumed.

The Save command

Purpose:

Save source file to disk

Syntax:

S <filename> <enter>

If a filename of the same name is found on disk, this will be marked for deletion and the new file will be stored on disk.

If the filename does not include a drive number, drive 0 is assumed.

If the filename does not include a file type suffix, .TX is assumed.

The Tape command

Purpose:

Save a compiled program to disk

Syntax:

T <filename> <enter>

This will store a compiled program to disk, with load and execute addresses set to 2140H. This is just above the BL Pascal runtime code.

If it is required to compile the code elsewhere then use e.g.:

T 3000 <filename> <enter>

This would cause the program to be stored with load and execute address set to 3000H.

If a filename of the same name is found on disk, this will be marked deleted and the new file will be stored on disk.

During file storage a list of blocks appears on the screen, like with tape storage. The disk light will be seen to flash on and off as those blocks are saved.

File saved and a command prompt **>** is shown when the save is complete.

If the filename does not include a drive number, drive 0 is assumed.

The filename does not need to include a file type suffix, .PA is assumed. This suffix must be used, if the program is required to be started from the Polydos command line.

Running a compiled program

When a compiled program, produced by the Tape or T command, is required to run, the file name can be entered at the Polydos command prompt:

e.g. MYFILE:3 <enter>

This will cause:-

1. The runtime portion of DPASCAL.GO:0 or failing that DPASCAL.GO:3 to be loaded into memory at 1000-213FH
2. The file MYFILE.PA on drive 3 also loaded into memory
3. MYFILE.PA will be executed

If the filename does not include a drive number, drive 0 is assumed.

If the filename suffix .PA is not present, that suffix is assumed always.

Memory usage

Besides the memory usage described in the original BL Pascal handbook, BL Pascal for Polydos uses:-

1. 256 bytes as sector buffer, starting at memory location EOFP (See Appendix B of the original manual on page 14) This is only required during the T command.
2. 19 bytes starting at C0C0H in Polydos system workspace, which is set aside for user applications.

Tips

If, like me, you're writing Pascal on a Nascom 4, don't forget to speed things up when a lot of calculation is required:

My Nascom 4 will tolerate up to OUT($1A,3) – but you may be lucky and go all out with OUT($1A,0)

Before user input, though, slow things down to 'normal' so the keyboard doesn't stutter too many letters every time you press a key:

OUT($1A,$20) does the trick.

Conclusion

CAUTION: This software must be regarded as experimental for now. I've used it for several months now without ill effect: However, best not to endanger your other software or data, make some dedicated disks for BL Pascal for Polydos.

I hope you are tempted to have another go with BL Pascal now that it is much more useful under Polydos.

I had a lot of fun puzzling out how to make this work - Bob Edwards, ham radio callsign G4BBY, S.W. England, April 2024