Nascom Documentation Library – Polydos Nasforth

Forward

Nasforth is a stand-alone forth for the Nascom computer. It used a RAM based ‘disk’ for program and data storage, with the ability to dump and load that ‘disk’ from tape cassette based backup.

I wrote an extension for Nasforth that would:-

1. Eliminate cassette storage.
2. Create or reuse a named Polydos disk file to act as the forth ‘disk’
3. Be able to call the majority of Polydos system calls from Nasforth code.

Installation

1. Run Nasforth v1.11 from the Polydos prompt. Check Nasforth is at the ‘ok’ prompt
2. Send file dforth.fth to the nascom from a PC via the serial port. Nasforth is always waiting for input from the local keyboard or the serial port. You may need to slow the data rate down so that nasforth has time to compile. Most terminal programs allow inter-character and inter-line delays that can be added – Tera Term under Windows does for instance
3. The code in dforth.fth will be compiled into Nasforth. When all the file is loaded, finalize Nasforth with word SYS-SAVE from the local keyboard. Note the highest address in use from the list of blocks sent to the ‘tape cassette’ (if it were present)
4. In my case, the last line of the report was 4400 00DE, so I rounded that up to 4500 hex.
5. Press CTRL-SHIFT-@ to drop out into Polydos
6. Now save the new version of forth to the disk - SAVE DFORTH.GO 1000 4500 1000 1000 <cr> in my case
7. That’s it, run with the command DFORTH as normal from Polydos

Storing / loading forth source code

Via Command File

Nasforth can be started from a line in a command text file. What I discovered was, once started, that Nasforth would accept the rest of the command file as input text. So if a file starts with

NASFORTH.GO and the remainder of the command file is a forth program, then that will be compiled by Nasforth and it’s all self-pacing.. How convenient is that?

The command file can be edited by the Polydos text editor and thus the clunky line editor traditional in all fig-forths is avoided.

Via Figforth Line Editor

If you do happen to prefer the traditional line editor that is included in Nasforth, then that is now supported:-

1. Run forth word WARM . This will ask you whether you want to create a file or use an existing one. The files are all named with a .FO extension. They are a container for the traditional forth ‘screens’ used by the line editor.
2. If creating a new file, you will be asked how many screens you want, with a hint of the maximum number possible in the remaining space on the disk.
3. Once that’s done, you would start writing your forth program with the inbuilt line editor just like you did with tape cassette (See section 6 of the Nasforth manual). Before leaving Nasforth, run the word FLUSH, to ensure all edits are stored in the disk fiile. If you leave Nasforth using BYE, that does a FLUSH for you.
4. N.B. Screen 1 to N (where N is the file size you chose) is available to you. Do not use screen 0 as it malfunctions – always did do, even with cassette. May be fixed in a later version.
5. To reload the program another time, rerun WARM, select the required existing .FO file and then execute 1 LOAD (or whereever the first ‘screen’ of your program is located)

Polydos system calls

The following Polydos calls are defined in the new extension:-

1.	DSIZE	RETURN DISK SIZE IN SECTORS
2.	DRD	READ SECTORS
3.	DWR	WRITE SECTORS
4.	RDIR	READ DIRECTORY
5.	WDIR	WRITE DIRECTORY
6.	CFS	CONVERT FILE SPECIFIER
7.	LOOK	LOOKUP FILE IN DIRECTORY
8.	ENTER	ENTER FCB INTO DIRECTORY
9.	COV	CALL AN OVERLAY
10.	COVR	CALL AN OVERLAY AND RESTORE

11. CKER	CHECK FOR A SYSTEM ERROR
12. POUT	OUTPUT CHAR TO PRINTER

For details about input and output parameters, refer to the dforth.fth source code and the Polydos programmers guide.

That’s it for now, hope you enjoy Polydos Nasforth in a few programming projects – Bob Edwards Oct 2023.