PFORTH for MVS

Port of pforth to MVS 3.8j

PFORTH

- A portable forth written in ANSI-C
- Public Domain
- http://www.softsynth.com/pforth/

MVS 3.8j

- MVS is “Multiple Virtual Storage”
- IBM operating system for the S/370 architecture
- MVS 3.8j is a public domain version of the ‘80
- Can be run under Linux using Hercules

Hercules

- Available at http://www.conmicro.cx/hercules/
- Emulator for the IBM mainframe hardware (System/370, System/390 and z/Series)
- Runs under Linux and is released under the open source software license QPL.

Dignus C/C++

- A cross compiler targeting the 390 architecture
- Also available under an hobbist license.

Objective

- A native pforth running on MVS under TSO and under JES2
- A cross pforth running on Linux and able to generate full resident load modules (statically linked binary)

Hacking guidelines

- Free-time project
- Minimize the work needed to have a version of pforth running on MVS (just minimalist a port)
- As much ANS as possible for an EBCDIC platform
- Start doing some forth on this platform and decide about the next steps.

Make and install (1)

- The base dictionary is generated on Linux using a big-endian version of pforth
- This produce a .h containing the dictionary
- But this .h contains hard coded ASCII character

Make and install (2)

- Compile with the Python wrapper Dcc.py (from a make –n) u.s.w.
- This produce a load module in XMIT format that can be binary transferred via 3270 and instantiated
- NPA100.TEST.BIN(PFORTH)
- NPA100.TEST.CLIST(PFORTH)

Run

- EX 'NPA100.TEST.CLIST(PFORTH)‘ (from TSO)

-
  ```
  /* ALLOCATE NEEDED DD AND RUN
  /* PFORTH
  ALLOC F(STDOUT) DA(*)
  ALLOC F(STDERR) DA(*)
  ALLOC F(STDIN) DA(*)
  CALL 'NPA100.TEST.BIN(PFORTH),
  FREE F(STDIN)
  FREE F(STDERR)
  FREE F(STDOUT)
  ```

Status

- „Half working“
- EBCDIC related problems: for ex. the FORTH word .“ does not work because search for “ that was hard coded in ASCII during the cross generation of the dictionary

Problems

- S390 is big-endian – x86 is little-endian
- MVS is EBCDIC
- MVS has a unusual and complicated record oriented file system

Solutions

- Read the dictionary from a file on MVS
- Generate the dictionary to be embedded on an EBCDIC system

Questions

- Fragen ?
- Domande ?