STABLE

an extreme small an fast FORTH-VM

Many thanks to Sandor Schneider who gave stable to public domain.
See STABLE in action:video (22M).

Stable (*STAck Bytecode Language & Engine*) is an extreme minimal but useful stack oriented virtual machine, currently written in C (200 lines of extensively commented C code. The original comes with 47 lines of C code). The machine language consists only of printable letters which make it simple hackable with an editor. The goal of this engine is to be human hackable, simple and fast. So there are no complicated address calculations (for branch and jump addresses) and an easy to use instruction set. (43 instructions, 26 registers, 26 user defineable functions)

To avoid addresses to variables (registers) and functions, names are predefined. So instead of given the address for a function there is an ASCII opcode for that. Functions are upper case letters from A to Z which leads to 26 user defineable functions, registers are named from a to z (lower letter) which leads to 26 registers.

Simple down counter

```
with an editor write a;[#." "1-#] into counter.txt, then
$ ./stable counter.txt 10
=> 10 9 8 7 6 5 4 3 2 1
```

Explanation

```
Arguments
counter.txt  is the code to be executed
```

```
10          will be stored into register a

Code commented
a;    push content of register a onto data stack (which is 10)
[     begin a while loop if top of stack is non zero
#     duplicate top of stack
.     write top of stack to output and drop it from data stack
" "   write a space to output
1     push literal 1 onto data stack
-     subtract, which decrements the loop counter by 1
#     duplicate top of stack
]     jump to [ if top of stack is not zero, drop top of stack

Code with stack effect
( x y z)  means x is 3rd item, y is 2nd item and z is top of stack (TOS)

a;    ( count)
[
#     ( count count)
.     ( count)
" "   ( count)
1     ( count 1)
-     ( count')  count' is now decremented by 1
#     ( count' count')
]     ( count)
```

See stable.counter.mp4 (150k) running

Downloads

- makefile.stable Unix Makefile
- stable.c (5k) ANSI-C with function pointer. Tracing support. Floating support
- stable_fast.c (5k) direct threaded (gcc extension) and tos in register (4-6 times faster), floating support, no tracing

- stable_debug.c (5k) single step monitor included. Floating support, no tracing
- stable_glossar.html instruction set documentation
-

- stable.ori.c (2k) original source code
- stable.ori.txt (2k) original instructions

Samples

- counter.txt sample from above
- cal70.txt $ stable cal70.txt 2017 7
  Sandor Schneider commented:
  I think, you recognised, that in case 2016, the command: stable cal70 2016 5 1 In case of 29 day februar in the "c" register must be 1 ! Default value in all registers 0 !
- calminft.txt $ stable calminft.txt 2017 7
- chessmin.txt
- chess.txt
- div.txt $ stable div.txt 355 113 6
  $ stable div.txt 1 -3 55
- fib.txt $ stable fib.txt 20
- life.txt $ stable life.txt 46
- measure.txt $ stable measure.txt

Simulator See life.txt running ...
stable.life2.mp4 (1.7M)

Mandelbrot (ASCII)
Video of simulation (3M)
Image of ASCII Mandelbrot
Stable-Application mandel.st