Hoe bij te dragen aan Forth Works

presentatie-README.txt

Readme voor presentatie en bla.fth 2024-08-12, wj

presentatie is een tekstfile met de presentatie

blah.fth is een tekstfile maar ook een forth programma

Starten van bla.fth
gforth bla.fth op de commandline
(alleen getest met gforth 0.7.3 op aarch64)

Gebruik van bla.fth
bla leest de presentatie file en geeft kort een instructie: n|p|q,
toont daarna de eerste pagina

volgende pagina: n toets (van next)
vorige pagina: p toets (van previous)
stoppen: q toets (van quit)
Letters zijn te klein: zoom in op je terminal om ze groter te maken
Mijn terminal programma heeft geen zoom: verlaag je scherm resolutie of installeer een ander terminal programma

Succes!

bla.fth

```
\ bla presentatie tool -- 2024-07 -- wj
\ scherm cursor
: csi 27 emit [char] [ emit ; \ csi = Control Sequence Introducer
: geen-cursor csi ." ?25l" ;
: toon-cursor csi ." ?25h" ;
\ scherm afmetingen
: breedte  ( -- n ) form swap drop ;
: hoogte   ( -- n ) form drop ;
\ aantal regels nodig voor een tekst
: #regels ( u.len -- u.regels )  breedte / 1+ ; \ Note: rough estimate, which is
sufficient for this purpose
\ een regel afdrukken, gecentreerd
: naar-starthoogte ( u.len -- ) >r  0  hoogte 2 / r>  #regels 2 / -  at-xy ;
: afbreken ( cadr u -- cadr u') \ als er een spatie in de string is
             \ in de buurt van het einde pas u aan tot op de spatie
    2dup 2dup >r 1- + r> 1- 20 min 0
    ?do dup c@ BL = if swap - swap drop unloop exit else 1- then loop 2drop ;
: opvullen ( u -- u ) dup breedte - abs 2 / spaces ;
: bla-deel ( cadr u - cadr' u' ) \ type een deel en update stringadres en
lengte, zo nodig tot nul
    dup  breedte < if 2dup  opvullen type dup - exit then \ string past op een
regel, zorg dat u' nul wordt
    over breedte afbreken opvullen dup >r type cr r@ - swap r> + swap ; \ te
lang voor een regel, typ een deel
create regel 0 , \ points to current displayed line
```

```
: bla. ( cadr u -- )  page regel @ .   dup naar-starthoogte   begin bla-deel dup
0= until 2drop ;
\ lees de tekst file en berg op in woorden bla000-bla999
create bla-teller 0 , \ reading lines from file
: bla-teller+  bla-teller @ 1+ 999 min bla-teller ! ; \ limit at 999 lines
: >bergop ( cadr u -- ) s" create blaxxx 0 , dup allot align  dup blaxxx !
blaxxx cell+ swap cmove" evaluate ;
: haalop> ( -- cadr u ) s" blaxxx cell+  blaxxx @" evaluate ;
\ find in >bergop and haalop> all substrings xxx and remember the addresses
\ - the assumption is these addresses will remain constant during the session
create digitadresses 0 , 0 , 0 , 0 , 0 , \ first 3 for >bergop, the rest for
haalop>
: getxxx-adresses ( -- )
    ['] >bergop >body 80 \ limit length to location of xxx substrings --
portability warning
    s" xxx" search if over digitadresses ! 3 - swap 3 + swap then
    s" xxx" search if over digitadresses cell+ ! 3 - swap 3 + swap then
    s" xxx" search if over digitadresses 2 cells +  ! then
    2drop
    ['] haalop> >body 40 \ limit length to location of xxx substrings --
portability warning
    s" xxx" search if over digitadresses 3 cells +  !  3 - swap 3 + swap  then
    s" xxx" search if over digitadresses 4 cells +  !  then
    2drop ;
\ replace xxx locations by formatted 3 digit bla-teller e.g "001" using
: set ( u -- )
    0 <# # # # #> \ make string 3 digits
    5 0 do 2dup digitadresses i cells + @ swap cmove loop 2drop ;

create fid 0 ,
create buf 1000 allot
align

: bla ( -- )
    cr cr ." n|p|q" 500 ms
    bla-teller @ 0= if
        s" presentatie" r/o open-file throw fid !
        getxxx-adresses
        begin
            buf 1000 fid @ read-line throw
         while buf swap bla-teller @ set >bergop  bla-teller+
        repeat
    then
    0 regel !
    geen-cursor
    begin
        regel @ set haalop> bla.
        key dup dup [char] q = if toon-cursor exit then
        [char] n = if regel @ 1+  bla-teller @ 1-  min regel ! then
        [char] p = if regel @ 1-  0  max regel ! then
    again
    toon-cursor ;

\ page ." bla presenteer tool -- 2024" cr breedte . 1000 ms

: tests-afbreken
s" 12345678901234567890123" afbreken  ~~ 2drop ." 23 expected" cr
s" 1234567890123456789012 " afbreken  ~~ 2drop ." 22 expected" cr
s" 1234 678901234567890123" afbreken  ~~ 2drop ."  4 expected" cr
s" 1 34567890123456789012" afbreken  ~~ 2drop ." 23 expected" cr
s" 1 34567890"              afbreken  ~~ 2drop ."  1 expected" cr
s"  234567890"              afbreken  ~~ 2drop ." 10 expected" cr
;
\ tests-afbreken bye
```

```
: tests-opvullen
42 emit breedte     opvullen 42 emit . cr
42 emit breedte 2 - opvullen 42 emit . cr
42 emit 0           opvullen 42 emit . cr
;
\ tests-opvullen bye

bla cr bye
```

presentatie

Een verhaal met gedachtensprongen over Forth Works -- Willem Jager
of: Waarom ik nog niets heb bijgedragen aan Forth Works
of: Een inkijkje in mijn chaotische werkwijze
... de gedachte: misschien heb ik wel iets nuttigs voor Forth Works om bij te dragen want het lijkt me sympatiek om code te delen
... bijvoorbeeld de string woorden in de noties applicatie, die gebruik ik veel en zijn handig in het gebruik
voorbeeld:
leeg>buf s" vim +$ " >buf bestand-plaats >buf (cadr.filenaam u) >buf (de buffer bevat nu de tekst: "vim +$ ~/w/noties/r/1722322443") buf> system
-- hoe is dat zo gekomen? --
In 2020 forth leren: notitie database
string manipulatie woorden fetch, store en append -- die waren er niet...
-- 2012 Forth standaard string woorden --
https://forth-standard.org/standard/string
-TRAILING /STRING BLANK CMOVE CMOVE> COMPARE SEARCH SLITERAL
Extension words: REPLACES SUBSTITUTE UNESCAPE
CMOVE ... en verder ??
dat was een verrassing
de details:
CMOVE (c-addr1 c-addr2 u --)
(eigenlijk CCOPY)
If u is greater than zero, copy u consecutive characters from the data space starting at c-addr1 to that starting at c-addr2, proceeding character-by-character from lower addresses to higher addresses.
CMOVE omslachtig in het gebruik, met veel stack manipulaties en zo
Dus:
Kopieer een string naar een buffer: >buf
: >buf (cadr u --) >lengte-begrenzing> dup >r 'buf #buf @ + swap move r> #buf @ + #buf ! ; ---> Wat is hier fout?
beginners foutje: move address units; cmove werkt met characters
dus gaat mis met meer dan 1 teken per address unit
byte-addressable
??? vraag: wie gebruikt er een machine met meer dan 1 teken per adres locatie?
Genoeg, we gaan door -->
Dus: een buffer en manipulaties op die buffer:
2500 CONSTANT tekstlengte
CREATE 'buf tekstlengte chars allot align
CREATE #buf 0 ,
leeg>buf (--)
>buf (cadr u --)
buf> (-- cadr u)
buf.ruimte (-- u)

'buf.ruimte (-- cadr)
>lengte-begrenzing> (cadr u -- cadr u)
invoer>buf (--) \ ACCEPT naar buffer
Dus misschien ook wel handig voor iemand anders...
Echter:
Eerst kijken of er zoiets al was in Forth Works, anders is het geen geschikte bijdrage
en vond:
String woorden
https://github.com/project-forth-works/project-forth-works/tree/main/Algorithms/String%20handling
$@ $! $+! $.
in analogie met c@ c! etc...
... met inspiratie vanuit BASIC (S1$, MID$ etc..)
Sprongetje
zie ook: https://forth-standard.org/proposals/string-store-and-fetch
en: https://home.hccnet.nl/a.w.m.van.der.horst/forthlectureC.html
$! $+! $@ $C+ etc.
ciforth string woorden
$!-BD $! $+! $, $/ $@ $C+ $\ $^ -TRAILING BL COUNT S"
Enfin, geen goede bijdrage dus...
De vraag blijft: wat zou ik kunnen bijdragen aan Forth Works?
Die string woorden zijn al in ruime mate en allerlei varianten beschikbaar
Bijkomende vraag: voldoet mijn code eigenlijk wel aan Generic Forth?
Sprongetje
Wat is Generic Forth?
Een subset van de ANS94 standaard forth woorden
Waarom ANS94? geen idee
zie https://github.com/project-forth-works/project-forth-works.github.io/blob/main/minimalforth.md
"So we defined a prototype of a word set extension based mainly on the core wordset of dpANS94."
Gericht op embedded applicaties
Want: geen OS en dus geen OS calls of SYSTEM, geen file of block woorden
Sprongetje
Vraag: waarom is Generic Forth relevant?
ik denk --> compatibiliteit
dat wil zeggen: in de meeste 'normale' Forths te gebruiken, dus nuttig voor iedereen die forth gebruikt
Sprongetjes
Vraag: analyse van forth code: welke forth woorden gebruikt mijn applicatie?
Vraag: welke forth woorden die ik gebruik zijn niet beschikbaar in:
Generic Forth -- NOforth -- gforth -- Minimal Forth -- ciforth -- ISO/ANS core ...
Vraag: hoe beantwoord ik die vraag?
Antwoord: een Forth applicatie, natuurlijk
Hoe?
lees forth code van file
overslaan van commentaar en string literals:
(." ABORT" S" .(C" PARSE \
vinden van definities:
: CREATE VARIABLE CONSTANT etc.
Vinden van aanroepen van woorden -- dat is dus de rest
Splitsen in forth woorden en in de applicatie gemaakte woorden
forth-language-words-<app>: lijst van forth woorden gebruikt in applicatie

application-words-<app>: lijst van woorden gedefinieerd in de applicatie
lijst van woorden in Generic Forth -- NOforth -- gforth -- Minimal Forth -- ciforth -- ISO/ANS core ...
Verschil bepalen tussen lijsten van woorden
Als we dat kunnen dan hebben we onze vragen beantwoord, maar met een aantal beperkingen
helpt niet als woorden van een specifieke forth qua functionaliteit afwijken van de standaards want deze analyse kan dit feit niet vaststellen
Demo?
vraag aan het publiek: is zo'n analyse programma een goede bijdrage voor forth works? In de categorie applicaties?
Sprongetje
Andere methode om compatibiliteit te bepalen:
ans-report.fs van gforth -- instrumenteert word header creatie mechanisme, werkt alleen voor gforth
Voorbeeld: zie gforth-environment-example-compatibility-report
Sprongetje
Alternatief: bla.fth (het programma dat deze presentatie toont)
vraag aan het publiek: is bla een goede bijdrage voor forth works?
Vragen over dit verhaal?
Bedankt voor jullie onverdeelde aandacht...
FIN

Vergelijking van mijn ad-hoc buffer woorden <--> Forth Works String Woorden:
CREATE 'buf 150 allot <--> 150 $VARIABLE BUF
leeg>buf >buf <--> BUF $! Store string in string variable
>buf <--> BUF $+! Append string to string variable
buf> <--> BUF $@ Fetch string variable
type <--> $. Type string
Gebruiksvoorbeeld:
Noties: leeg>buf s" abc" >buf 123 .>buf buf> type
Forth Works: 20 $VARIABLE BUF BUF s" abc" $! BUF s" 123" $+! BUF $.
Vergelijking
mijn ad-hoc string woorden (>buf etc.) zijn gemaakt voor de notitie applicatie
de Forth Works string woorden passen meer in de forth traditie, zijn korter
$VARIABLE kan gebruikt worden met meer stringbuffers
en kan de lengte van de buffer als metadata bevatten
zodat je overflow kan detecteren
FIN