# "Klakita E" - An assembler written in pure Tcl

"Klakita E" (it means: clicked Esperanto programming language) is a collection of Tcl-scripts that permit to create binary code for a handfull of computers and microcontrollers that use operating systems like

DOS, Linux, Android, Windows

and use processors like

Intel-Processors (8086 ... Pentium), ARMv6, ARMv7 and the AtTiny-microcontroller.

To start: `wish 0`

| scripts | widgets | data files |
|---|---|---|
| ====== | ======= | ========== |
| 0 gui | .t | 100 content page |
| 180 some buttons | | |
| 96 97 98 layouts | | |
| 1000 write source | .top1.t | 1001 source text |
| | .top2.t | 1002 prototypes of macros and args |
| 900 assemble source file | | |
| 901 expand macros I | | 1001 -> 1004 |
| 902 expand macros II | | 1003 macros |
| 903 expand macros III | | 1004 == 1005 ? |
| 904 calculate absolute adresses | | 1005 |
| 905 create binary code | | 1006 1007 1008 |
| 907 change permissions & start code - output eventually in 1009 | | |

estas markita

```
bind .t <ButtonRelease-1> {
set x ""
catch {set x [.t get sel.first sel.last]}
catch {
if [string is integer $x] {
incr x -1
set i $x
.b3 invoke
return}
if [string is double $x] {
set x [expr int($x)]
source $x
return
}}
}
```

# CONTENT

system

| | | | |
|---|---|---|---|
| 1. | L i n u x / Android | 64 | x86-64 |
| 2. | L i n u x / Android | 32 | 80386 |
| 3. | L i n u x / Android | 32 | ARMv7 |
| 4. | Windows wine/xp/10 | 32 | 80386 |
| 6. | D O S .com | 16 | 8086 |
| 10. | Microcontroller | 8 | AtTiny84 |

# example source files

111. start finish

118. Hello, world!

112. a= a<- a-> a+ a- read write

113. a+ a- jz jc jmp

115. jsr ret

110. empty source text

119. ascii-codes

120. ascii-codes compact
additional macros needed!
220. add them and retry

131. LED blinking AtTiny only

instrukcioj 1002

```
######## E macros ########
a=      number
a->     adress
a<-     adress
a+      number
a-      number
jmp     adress
jz      adress
jc      adress
jsr     adress
ret
start   adress
read    number  adress
write   number  adress
finish
space   number
no.     number
txt     "text"
```

>> <<

source text 1001 0 locations

```
# empty page
```

nl -- ++ content

instrukcioj 1002

```
######## E macros ########
a=      number
a->     adress
a<-     adress
a+      number
a-      number
jmp     adress
jz      adress
jc      adress
jsr     adress
ret
start   adress
read    number  adress
write   number  adress
finish
space   number
no.     number
txt     "text"
```

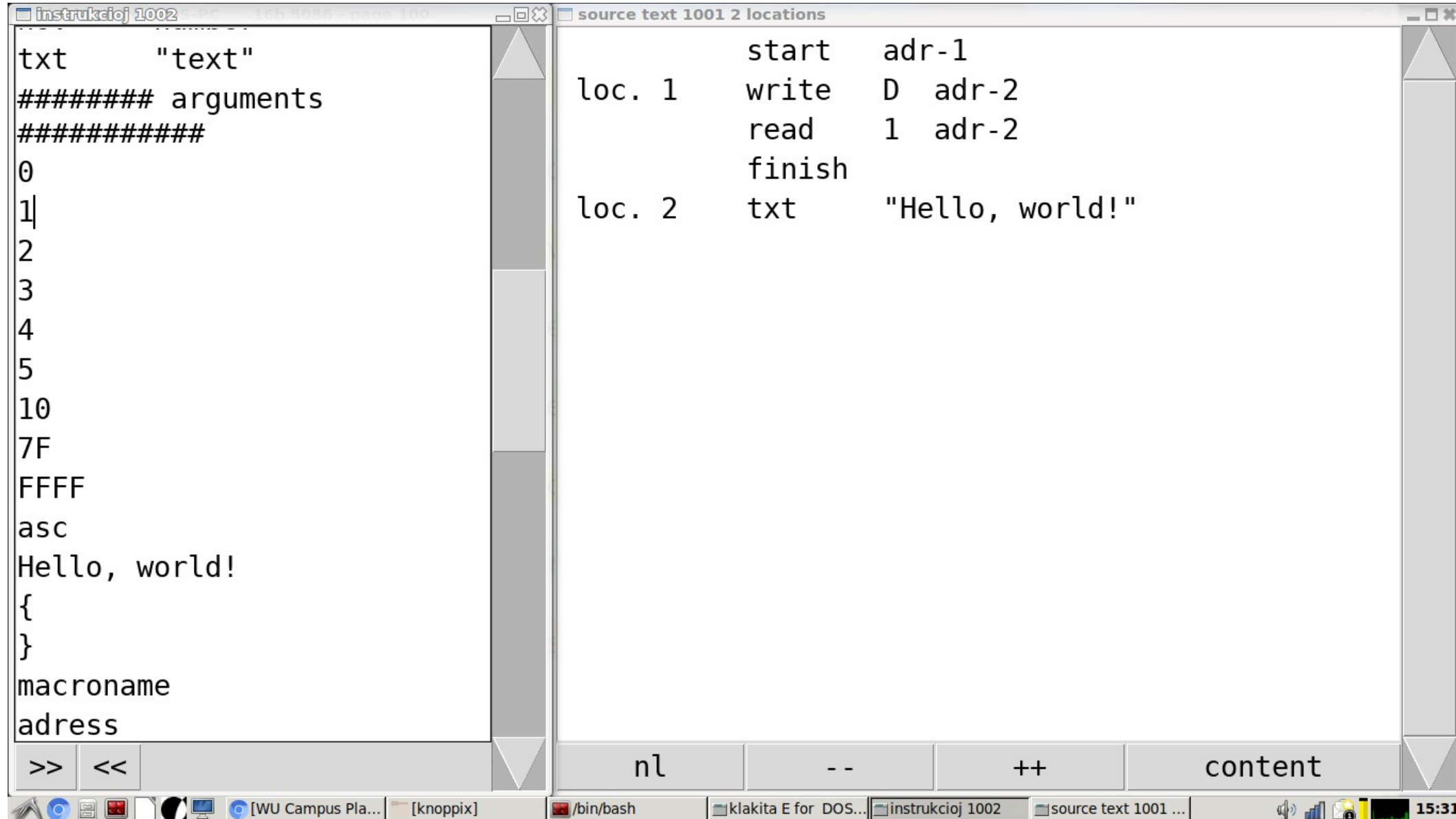
>> <<

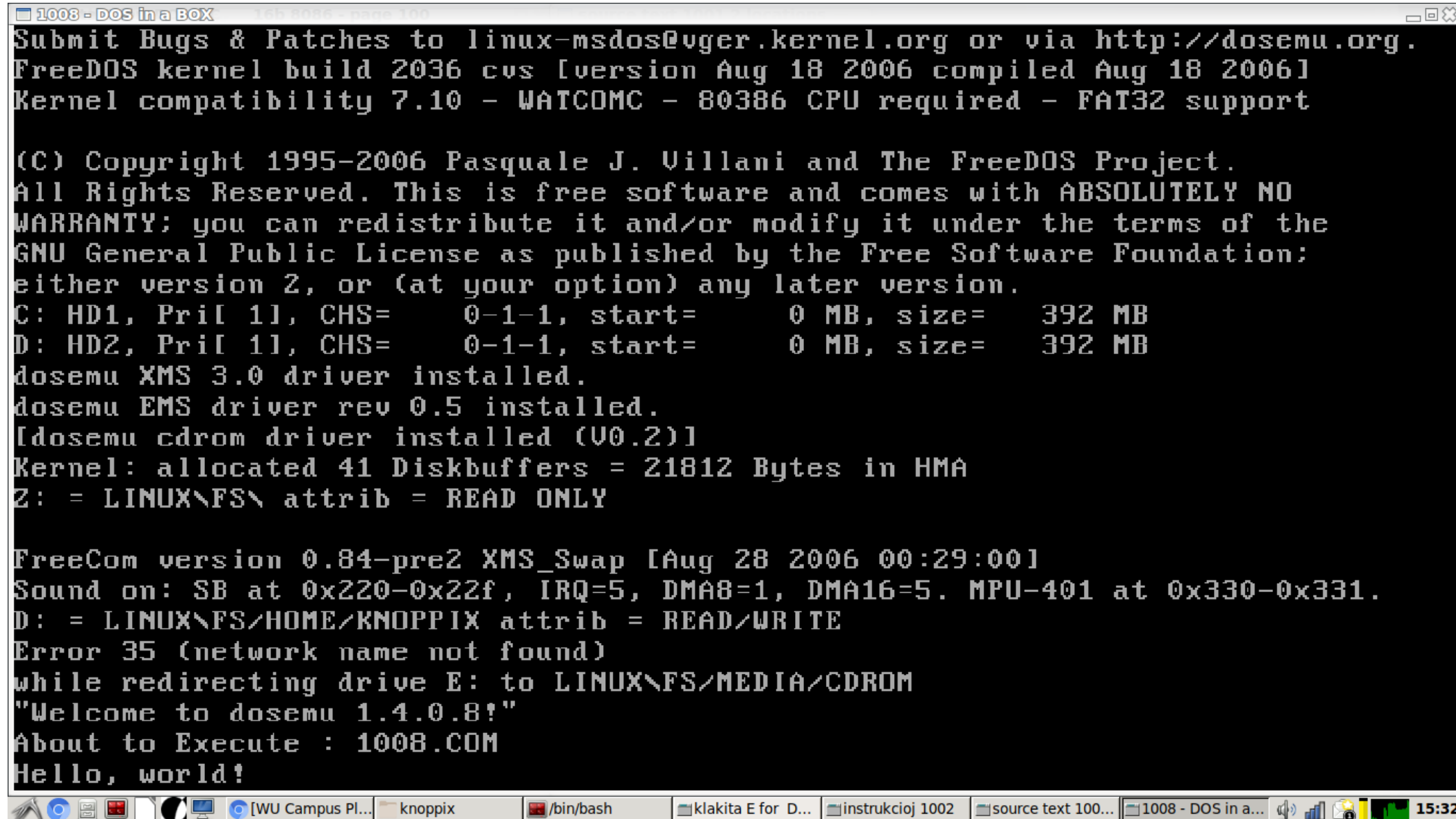
source text 1001 0 locations

```
start   adress
write   number  adress
read    number  adress
finish
txt     "text"
```

nl -- ++ content

[WU Campus Pla... [knoppix] /bin/bash klakita E for DOS... instrukcioj 1002 source text 1001 ... 15:30


instrukcioj 1002
txt "text"
######## arguments
###########
0
1
2
3
4
5
10
7F
FFFF
asc
Hello, world!
{
}
macroname
adress
>>
<<
source text 1001 2 locations
start adr-1
loc. 1 write D adr-2
read 1 adr-2
finish
loc. 2 txt "Hello, world!"
nl
--
++
content
[WU Campus Pla...
[knoppix]
/bin/bash
klakita E for DOS...
instrukcioj 1002
source text 1001 ...
15:31


1008 - DOS in a BOX
Submit Bugs & Patches to linux-msdos@vger.kernel.org or via http://dosemu.org.
FreeDOS kernel build 2036 cvs [version Aug 18 2006 compiled Aug 18 2006]
Kernel compatibility 7.10 - WATCOMC - 80386 CPU required - FAT32 support
(C) Copyright 1995-2006 Pasquale J. Villani and The FreeDOS Project.
All Rights Reserved. This is free software and comes with ABSOLUTELY NO
WARRANTY; you can redistribute it and/or modify it under the terms of the
GNU General Public License as published by the Free Software Foundation;
either version 2, or (at your option) any later version.
C: HD1, Pri[ 1], CHS= 0-1-1, start= 0 MB, size= 392 MB
D: HD2, Pri[ 1], CHS= 0-1-1, start= 0 MB, size= 392 MB
dosemu XMS 3.0 driver installed.
dosemu EMS driver rev 0.5 installed.
[dosemu cdrom driver installed (V0.2)]
Kernel: allocated 41 Diskbuffers = 21812 Bytes in HMA
Z: = LINUX\FS\ attrib = READ ONLY
FreeCom version 0.84-pre2 XMS_Swap [Aug 28 2006 00:29:00]
Sound on: SB at 0x220-0x22f, IRQ=5, DMA8=1, DMA16=5. MPU-401 at 0x330-0x331.
D: = LINUX\FS/HOME/KNOPPIX attrib = READ/WRITE
Error 35 (network name not found)
while redirecting drive E: to LINUX\FS/MEDIA/CDROM
"Welcome to dosemu 1.4.0.8!"
About to Execute : 1008.COM
Hello, world!
[WU Campus Pl...
knoppix
/bin/bash
klakita E for D...
instrukcioj 1002
source text 100...
1008 - DOS in a...
15:32

```
100  B8 05 01 FF E0 # header 105
105  B8 00 40 # ax= 4000
108  BB 01 00 # bx= 1
10B  B9 0D 00 # cx= D
10E  BA 26 01 # dx= 126
111  CD 21 # int 21
113  B8 00 3F # ax= 3F00
116  BB 00 00 # bx= 0
119  B9 01 00 # cx= 1
11C  BA 26 01 # dx= 126
11F  CD 21 # int 21
121  B8 00 4C # ax= 4C00
124  CD 21 # int 21
126  48 65 6c 6c 6f 2c 20 77 6f 72 6c 64 21
133
```

```
Files used 1000      --> 900  901,902,903,1003  --> 904    --> 905  -->  907
Data files 1001 1002          1004/1005     1006              1007     1008 1009
-------------------------------------------------------------------------------
          start   adr-1       header adr-1  100 header 105  B8 05 01 FF E0
loc. 1    write D adr-2       ax= 4000      105 ax= 4000     B8 00 40
                              bx= 1         108 bx= 1        BB 01 00
                              cx= D         10B cx= D        B9 0D 00
                              dx= adr-2     10E dx= 126      BA 26 01
                              int 21        111 int 21       CD 21
          read  1 adr-2       ax= 3F00      113 ax= 3F00     B8 00 3F
                              bx= 0         116 bx= 0        BB 00 00
                              cx= 1         119 cx= 1        B9 01 00
                              dx= adr-2     11C dx= 126      BA 26 01
                              int 21        11F int 21       CD 21
          finish              ax= 4C00      121 ax= 4C00     B8 00 4C
                              int 21        124 int 21       CD 21
loc. 2    txt "Hello, world!" txt "..."     126 txt "..."    48 65 6c 6c 6f 2c 20
                                                             77 6f 72 6c 64 21
```

```
add macros                          example: 1010
--------------
- add prototypes in file 1002       new-macro number adress
  argument prototypes are
  number, adress, {macro}, "text"

- add macros in file 1003           new-macro { ... arg-2
  the name of the macro at                     ... arg-1
  the beginning of line!!!                     ... }
  arguments are named arg-1 arg-2 ... arg-n

add maschine instructions       example: 1011
---------------------------
get informations from books or internet
- add instruction length of new instruction in proc increment in file 904
- add procedure for instruction in file 905

Don't forget to update your project!!!
```

```
 add the prototype of the macro 'repeat' to file 1002
#       start increment stop loop_macro  macro_after_loop
#        arg-1  arg-2  arg-3  arg-4       arg-5
 repeat adress number adress {macro}     {macro}

 add the macro to file 1003
repeat      {a<- arg-3
             a-> adr-3
 loc. 1      arg-4
             a<-     arg-1
             a+      arg-2
             a->     arg-1
             a-
 loc. 3      0
             jc      adr-1
             jmp     adr-2
 loc. 2      arg-5 }
```

add instruction  and= as follows:

add line in proc increment in file 904:

```
...
if {$t == "and="}        {set x   {1  2}}
...
```

add proc 'and=' in file 905:

```
...
#                              and ax,  number    -- 25 2bytes
proc and=          {number} {
set x "25"
set b [2bytes $number]
set x "$x $b"
return $x                    }
...
```

Thanks for listening!!!

Any questions?

contact:

uli.ender@esperanto.de