

```

;      "Movement intro" done in 2016
;
;
; Programming by Sylwester Dariusz Ha (SylwesterH)
;
;
; Music by Necropolo/Avatar/Hokuto Force/Lethargy/Level 64/SIDRIP Alliance
; NTSC fix by Karmic/HVSC Crew
;
;
;      Use latest 64TASS crossassembler to compile it
;      music synchronize for WinVice 1.15
;-----
;      Start intro from BASIC
;-----
;.comment
;      *= $0801
;      .word sys,2016
;      .null $9e,^main
sys      .word 0

;.endc
;-----
;      Intro initialisations
;-----
;      *= $0b00
dcopy

initm = $1000
playm = $1003
;      *= $2c80
lcol1 .byte $09,$06,$0b,$02,$08,$04,$0c,$0e
      .byte $0a,$05,$0f,$03,$0d,$07,$01,$07
      .byte $0d,$03,$0f,$05,$0a,$0e,$0c,$04
      .byte $08,$02,$0b,$06,$09,$ff
lcol = $2c00
starpos = $2cf0
;      * = $3600
main sei

      lda $d012 ;test NTSC/PAL
      cmp $d012
      beq *-3
      bmi *-8
      cmp #$37
      beq notntsc
      lda #<ntscply
      sta ntsc1+1
      lda #>ntscply

```

```
    sta ntsc1+2
    lda #$f8
    sta ntsc2+1
```

```
notntsc jsr downc
```

```
    lda #<scroll
    sta $11
    lda #>scroll
    sta $12
    ldx #$00
-   lda #$4b
    sta $0400,x
    lda #$01
    sta $d800,x
    inx
    bne -
    ldx #$00
-   lda #$20
    sta $04c8,x
    lda #$00
    sta $d8c8,x
    inx
    cpx #$28
    bne -
    ldx #$00
    lda #$a0
-   sta $04f0,x
    sta $05f0,x
    inx
    bne -
    ldx #$00
    lda #$a0
-   sta $06f0,x
    inx
    cpx #$30
    bne -
;-----
;      Crack name D.Y.C.P init
;-----
init9  =*+$01
        ldx #0
init2  =*+$01
init3  =*+$02
sinit10 lda cname
        cmp #$20
        bcs +
```

```

        clc
        asl
        asl
        asl
        tax
        ldy #0
-       lda $0c00,x
init5   =*+$01
sinit11 =*+$02
        sta dcopy,y
        inx
        iny
        cpy #8
        bne -
        jmp init1

+       clc
        asl
        asl
        asl
        tax
        ldy #0
        lda $0d00,x
init7   =*+$01
fast1   =*+$02
        sta dcopy,y
        inx
        iny
        cpy #8
        bne fast1
init1   lda init2
        clc
        adc #1
        bcc init4
        inc init3
init4   sta init2
        lda init5
        clc
        adc #8
        bcc init6
        inc sinit11
init6   sta init5
        lda init7
        clc
        adc #8
        bcc init8

```

```

        inc fast1
init8   sta init7
        inc init9
        lda init9
        cmp #$19
        beq next1
        jmp sinit10
next1
;-----
;       Generate D.Y.C.P chars
;-----
        ldx #0
        ldy #0
-       tya
        sta $0428+$07,x
        iny
        iny
        iny
        inx
        cpx #$19
        bne -
        ldx #0
        ldy #1
-       tya
        sta $0450+$07,x
        iny
        iny
        iny
        inx
        cpx #$19
        bne -
        ldx #0
        ldy #2
-       tya
        sta $0478+$07,x
        iny
        iny
        iny
        inx
        cpx #$19
        bne -
;-----
;       Initialise IRQ and music
;-----
        lda #$7f
        sta $dc0d

```

```

lda $dc0d
lda #1
sta $d01a
lda #$1b
sta $d011
lda #$32
sta $d012
lda $dd00
and #%11111100
ora #%00000011
sta $dd00
lda #<irq1
ldx #>irq1
sta $0314
stx $0315
lda #0
jsr initm

```

```

;-----
;      Initialise Sprites
;-----

```

```

lda #$00
sta $d017    ;sprites expand 2x vertical (y)
sta $d01d    ;sprites expand 2x vertical (x)
sta $d01c    ;sprites multi-color mode select
lda #%11111111
sta $d015    ;sprite on/off
lda #0
sta $d01b    ;Sprite to Background Display Priority
lda #1
  sta $d027  ;sprite 0 color
  sta $d028  ;sprite 1 color
  sta $d029  ;sprite 2 color
  sta $d02a  ;sprite 3 color
  sta $d02b  ;sprite 4 color
  sta $d02c  ;sprite 5 color
  sta $d02d  ;sprite 6 color
  sta $d02e  ;sprite 7 color
ldx #$00
lda #$21
-   sta $07f8,x
inx
cpx #$08
bne -
lda #$01
sta $0840
ldx #$00

```

```

        lda #$00
-       sta starpos,x
        inx
        cpx #$10
        bne -
        lda #$68
        clc
        adc #$11
        sta starpos+1
        adc #$0d
        sta starpos+3
        adc #$0d
        sta starpos+5
        adc #$0d
        sta starpos+7
        adc #$0d
        sta starpos+9
        adc #$0d
        sta starpos+12
        adc #$0d
        sta starpos+15
        adc #$0d
        sta starpos+17
        cli

```

this

```

        jsr colmov
        jsr sptpos
        jsr sptmov
        jsr space
        ldx #$00
        lda pcol1
-       sta $d800,x
        inx
        cpx #$c8
        bne -

```

```

        jmp this

```

```

;-----
;       Move stars
;-----

```

```

sptpos ldx #$00
-   lda starpos+1,x
    sta $d001,x
    lda starpos,x
    asl
    ror $d010

```

```

    sta $d000,x
    inx
    inx
    cpx #$10
    bne -
    rts
sptmov ldx #$00
-      lda starpos,x
    clc
    adc starspd,x
    sta starpos,x
    inx
    inx ;
    cpx #$10
    bne -
    rts
;-----
;      Scroll move
;-----
scr    lda scpos
    sec
    sbc #$02
    and #$07
    sta scpos
    bcc +
    rts

+      ldx #$00
-      lda $0749,x
    sta $0748,x
    lda $0771,x
    sta $0770,x
    lda $0799,x
    sta $0798,x
    lda $07c1,x
    sta $07c0,x
    inx
    cpx #$28
    bne -
    ldy #$00
    lda ($11),y
    bne +
    lda #<scroll
    sta $11
    lda #>scroll
    sta $12

```

```

+      lda ($11),y
+      sta $076f
+      adc #$3f
+      sta $0797
+      adc #$40
+      sta $07bf
+      adc #$40
+      sta $07e7
+      inc $11
+      bne +
+      inc $12
+      rts
;-----
;      Enter IRQ
;-----
irq1  inc $d019
+      lda #$1b
+      sta $d011
+      lda #$1c
+      sta $d018
+      lda #$c8
+      sta $d016
+      ldx #$03
-      dex
-      bne -
+      lda #$00
+      sta $d020
+      sta $d021
+      ldx #$08
-      dex
-      bne -
+      lda pcol1a
+      cmp #$01
+      bne +
+      lda #$01
+      sta $d020
+      sta $d021
+      lda pcol1a
+      cmp #$0f
+      bne +
+      lda #$0c
+      sta $d020
+      sta $d021
+      lda #$59
+      sta $d012
+      lda #<irq2

```

```

        sta $0314
        lda #>irq2
        sta $0315
        jsr scr
        jmp $ea81
irq2    inc $d019
        ldx #$03
-       dex
        bne -
        lda #$00
        sta $d020
        sta $d021
        lda #$1b
        sta $d011
        lda #$15
        sta $d018
        lda #$c8
        sta $d016
        ldx #$03
-       dex
        bne -
        ldx #$00
        lda pcol1a
-       sta $db48,x
        sta $db70,x
        sta $db98,x
        sta $dbc0,x
        inx
        cpx #$28
        bne -
        lda #$da
        sta $d012
        lda #<irq4
        sta $0314
        lda #>irq4
        sta $0315
        jsr dclr
ntsc1   jsr playm
        jmp $ea81
irq4    inc $d019
        lda #$18
        sta $d018
        lda scpos
        sta $d016
ntsc2   lda #$ff
        sta $d012

```

```

        lda #<irq5
        sta $0314
        lda #>irq5
        sta $0315
        jmp $ea81
irq5    inc $d019
        lda #$32
        sta $d012
        lda #<irq1
        sta $0314
        lda #>irq1
        sta $0315
        jsr rout1
        jsr dmov
        jmp $ea81
;-----
;      Color movements
;-----
rout1
frwd    lda pcol2
        sta pcol1
        inc frwd+1
        lda frwd+1
        cmp #$30
        bne frwd2
        lda #$00
        sta frwd+1
frwd2   lda pcol2a
        sta pcol1a
        inc frwd2+1
        lda frwd2+1
        cmp #$60
        bne +
        lda #$30
        sta frwd2+1
+       rts
;-----
;      D800, D020, D021 color screen Fader
;-----
downc   lda #$10
        sta $ff
down1   ldx #5
        jsr waitx
        lda $d020
        and #$0f
        tax

```

```

        lda dcolor,x
        sta $d020
        lda $d021
        and #$0f
        tax
        lda dcolor,x
        sta $d021
        ldy #$00
        ldx #$d7
down2   inx
        stx down3
        stx down5
down3   =*+$02
down4   lda $d800,y
        and #$0f
        tax
        lda dcolor,x
down5   =*+$02
        sta $d800,y
        iny
        bne down4
        ldx down3
        cpx #$db
        bne down2
        dec $ff
        bne down1
        rts
;-----
;      Delay synchronisation
;-----
wait1   lda #$fb
        ldy #0
do1     iny
        bne do1
        clc
        adc #1
        bne do1
        rts
waitx   jsr wait1
        dex
        bpl waitx
        rts
;-----
;      Plasma color movements
;-----
colmov  ldx #$28

```

```

-    ldy logohf,x
    lda lcol,y
    sta $d8f0,x
    ldy logohf+$28,x
    lda lcol,y
    sta $d918,x
    ldy logohf+$28*2,x
    lda lcol,y
    sta $d940,x
    ldy logohf+$28*3,x
    lda lcol,y
    sta $d968,x
    ldy logohf+$28*4,x
    lda lcol,y
    sta $d990,x
    ldy logohf+$28*5,x
    lda lcol,y
    sta $d9b8,x
    ldy logohf+$28*6,x
    lda lcol,y
    sta $d9e0,x
    ldy logohf+$28*7,x
    lda lcol,y
    sta $da08,x
    ldy logohf+$28*8,x
    lda lcol,y
    sta $da30,x
    ldy logohf+$28*9,x
    lda lcol,y
    sta $da58,x
    ldy logohf+$28*10,x
    lda lcol,y
    sta $da80,x
    ldy logohf+$28*11,x
    lda lcol,y
    sta $daa8,x
    ldy logohf+$28*12,x
    lda lcol,y
    sta $dad0,x
    ldy logohf+$28*13,x
    lda lcol,y
    sta $daf8,x
    dex
    bmi +
    jmp -
+    ldx #$00

```

```

-      lda lcol+1,x
      sta lcol,x
      inx
      cpx #$1a
      bne -
      ldx plpos
      inc plpos
      lda lcol1,x
      cmp #$ff
      bne +
      lda #$00
      sta plpos
+      sta lcol+$1a
      rts

;-----
;      D.Y.C.P. clean
;-----

dclr   ldx #$18
      lda #0
-      .for i = 0, i < 25, i = i + 1
      #fill $3000,i,$18
      .next
      dex
      bpl -
      rts
fill   .macro
      sta (\1,x)+((\2)*(\3))
      .endm

;-----
;      D.Y.C.P. update
;-----

add1 = $3000
dmov   ldy dpos
      ldx dsin,y
      ldy #$00
-      lda dcopy,y
      sta add1,x
      inx
      iny
      cpy #$08
      bne -
      ldy dpos
      ldx dsin+$01,y
      ldy #$00
-      lda dcopy+$08,y
      sta add1+$18,x

```

```

inx
iny
cpy #$08
bne -
ldy dpos
ldx dsin+$02,y
ldy #$00
- lda dcopy+$10,y
  sta add1+$18*2,x
inx
iny
cpy #$08
bne -
ldy dpos
ldx dsin+$03,y
ldy #$00
- lda dcopy+$18,y
  sta add1+$18*3,x
inx
iny
cpy #$08
bne -
ldy dpos
ldx dsin+$04,y
ldy #$00
- lda dcopy+$20,y
  sta add1+$18*4,x
inx
iny
cpy #$08
bne -
ldy dpos
ldx dsin+$05,y
ldy #$00
- lda dcopy+$28,y
  sta add1+$18*5,x
inx
iny
cpy #$08
bne -
ldy dpos
ldx dsin+$06,y
ldy #$00
- lda dcopy+$30,y
  sta add1+$18*6,x
inx

```

```

    iny
    cpy #$08
    bne -
    ldy dpos
    ldx dsin+$07,y
    ldy #$00
-   lda dcopy+$38,y
    sta add1+$18*7,x
    inx
    iny
    cpy #$08
    bne -
    ldy dpos
    ldx dsin+$08,y
    ldy #$00
-   lda dcopy+$40,y
    sta add1+$18*8,x
    inx
    iny
    cpy #$08
    bne -
    ldy dpos
    ldx dsin+$09,y
    ldy #$00
-   lda dcopy+$48,y
    sta add1+$18*9,x
    inx
    iny
    cpy #$08
    bne -
    ldy dpos
    ldx dsin+$0a,y
    ldy #$00
-   lda dcopy+$50,y
    sta add1+$18*10,x
    inx
    iny
    cpy #$08
    bne -
    ldy dpos
    ldx dsin+$0b,y
    ldy #$00
-   lda dcopy+$58,y
    sta add1+$18*11,x
    inx
    iny

```

```

    cpy #$08
    bne -
    ldy dpos
    ldx dsin+$0c,y
    ldy #$00
-   lda dcopy+$60,y
    sta add1+$18*12,x
    inx
    iny
    cpy #$08
    bne -
    ldy dpos
    ldx dsin+$0d,y
    ldy #$00
-   lda dcopy+$68,y
    sta add1+$18*13,x
    inx
    iny
    cpy #$08
    bne -
    ldy dpos
    ldx dsin+$0e,y
    ldy #$00
-   lda dcopy+$70,y
    sta add1+$18*14,x
    inx
    iny
    cpy #$08
    bne -
    ldy dpos
    ldx dsin+$0f,y
    ldy #$00
-   lda dcopy+$78,y
    sta add1+$18*15,x
    inx
    iny
    cpy #$08
    bne -
    ldy dpos
    ldx dsin+$10,y
    ldy #$00
-   lda dcopy+$80,y
    sta add1+$18*16,x
    inx
    iny
    cpy #$08

```

```

    bne -
    ldy dpos
    ldx dsin+$11,y
    ldy #$00
-   lda dcopy+$88,y
    sta add1+$18*17,x
    inx
    iny
    cpy #$08
    bne -
    ldy dpos
    ldx dsin+$12,y
    ldy #$00
-   lda dcopy+$90,y
    sta add1+$18*18,x
    inx
    iny
    cpy #$08
    bne -
    ldy dpos
    ldx dsin+$13,y
    ldy #$00
-   lda dcopy+$98,y
    sta add1+$18*19,x
    inx
    iny
    cpy #$08
    bne -
    ldy dpos
    ldx dsin+$14,y
    ldy #$00
-   lda dcopy+$a0,y
    sta add1+$18*20,x
    inx
    iny
    cpy #$08
    bne -
    ldy dpos
    ldx dsin+$15,y
    ldy #$00
-   lda dcopy+$a8,y
    sta add1+$18*21,x
    inx
    iny
    cpy #$08
    bne -

```

```

        ldy dpos
        ldx dsin+$16,y
        ldy #$00
-       lda dcopy+$b0,y
        sta add1+$18*22,x
        inx
        iny
        cpy #$08
        bne -
        ldy dpos
        ldx dsin+$17,y
        ldy #$00
-       lda dcopy+$b8,y
        sta add1+$18*23,x
        inx
        iny
        cpy #$08
        bne -
        ldy dpos
        ldx dsin+$18,y
        ldy #$00
-       lda dcopy+$c0,y
        sta add1+$18*24,x
        inx
        iny
        cpy #$08
        bne -
        ldx dpos
        inx
        cpx #$28
        bne +
        ldx #$00
+       stx dpos
        rts

```

```

;-----
;       D.Y.C.P. sinus
;-----

```

```

dpos    .byte $00
dsin     .byte $0f
         .byte $0e
         .byte $0e
         .byte $0d
         .byte $0c
         .byte $0a
         .byte $09
         .byte $07

```

```

        .byte $05
        .byte $04
        .byte $03
        .byte $02
        .byte $02
        .byte $01
        .byte $01
        .byte $01
        .byte $01
        .byte $01
        .byte $01
        .byte $01
        .byte $02
        .byte $02
        .byte $03
        .byte $04,$05,$06,$07,$08,$0a,$0c,$0d
        .byte $0e,$0f,$0f,$10,$10,$10,$10,$0f
        .byte $0f,$0e,$0e,$0d,$0c,$0a,$09,$07
        .byte $05,$04,$03,$02,$02,$01,$01,$01
        .byte $01,$01,$01,$01,$01,$02,$02,$03
        .byte $04,$05,$06,$07,$08,$0a,$0c,$0d
        .byte $0e,$0f,$0f,$10,$10,$10,$10,$0f

        ;ntsc play
ntscply dec ntscnt
        bne +
        lda #6
        sta ntscnt
        rts
+       jmp playm
ntscnt .byte 6
        ;-----

;-----
;       Cracked game name and scroll text
;-----

.enc screen
;       0123456789abcdef012345678
cname .text " movement hf cracktro "

scroll .text "
        .text "hokuto force again is back in 2016 with new intro named as
movement hf cracktro ' ..... done in 2016 by hokuto force
members : code by botx, music by necropolo, some help by the overkiller, ntsc fix by karmic

```

```

always hands up in a-z order to our brothers that show their skills in the cracking
scene... "
.text "alpha flight - atlantis+f4cg - angels - avatar - excess - genesis project - laxity -
mayday - "
.text "nostalgia - onslaught . full respect to nowadays and past heroes for still sharing "
.text "the fun with us... acrise - active - aeg soft - ancients pledge inc. - arise - artline
designs - "
.text "arsenic - bonzai - booze design - camelot - censor design - civitas - code 7 -
commodore is awesome - "
.text "cosine - covert bitops - crescent - crest - dekadence - delysid - desire - dmagic -
dominators - "
.text "dual crew - fairlight - flood - hackersoft - hack n' trade - hitmen - hoaxers - k2 - krynn
- level 64 - "
.text "lnx - mhi - nightfall - noice - no name - offence - oxyron - padua - panda design -
people of "
.text "liberty - radwar - razor 1911 - remember - rgcd - samar productions - secure -
singular crew - "
.text "the movers - tnd - trsi - tropyx - vaudeville crew - wrath designs and last but not least
dohi - g-force - indie retro news - jcb - john64 - "
.text "kuklux - mankeli - mason - mdz - mix256 - reset magazine - skoe - ste'86 - swasti -
tlr - zagon and all our "
.text "friends in fb pri-vate groups - csdb site - gamebase 64 - vice team - lemon 64 and
the "
.text "forgotten rest..."
.text " "
.text "press space to exit!"
.text " "
.byte $00
crdts1 .text " Intro credits : "
crdts2 .text " Code by Sylwester Dariusz Ha"
crdts3 .text " (It's my life) cover by Necropolo "
crdts4 .text " NTSC Fix by Karmic "
.enc none
;-----
;      Cracked game relocations
;-----
space  lda $dc01
        cmp #$ef
        beq end
        rts
end     sei
        jsr $fda3
        jsr $fd15
        jsr $ff5b
        lda #$00
        sta $d020

```

```

        sta $d021
        ldx #0
        lda #0
-       sta $d400,x
        inx
        cpx #$50
        bne -
        lda #$17
        sta $d018
        jsr credits
        ldx #$00
-       lda reloc,x
        sta $0340,x
        inx
        cpx #$50
        bne -
        cli
        jmp $0340

reloc sei
        lda #$34
        sta $01
-       ldx #$00
-       lda crkgame,x
        sta $0801,x
        inx
        bne -
        inc $0349
        inc $034c
        lda $0349
        cmp #$00
        bne --
        lda #$37
        sta $01
        cli
        jmp $fce2 ; -----> Cracked game start address

credits ldx #0
-       lda crdts1,x
        sta $0400,x
        lda #0
        sta $d800,x
        lda crdts2,x
        sta $0450,x
        lda #0
        sta $d850,x
        lda crdts3,x

```

```

sta $0478,x
lda #0
sta $d878,x
lda crdts4,x
sta $04a0,x
lda #0
sta $d8a0,x
inx
cpx #$28
bne -
xx0  lda #5
      jsr waitx
      lda #0
xx1  lda crdtsc
-    sta $d800,x
      inx
      cpx #$28
      bne -
      inc xx1+1
      lda xx1+1
      cmp #$10
      bne +
      jmp ax0
+    jmp xx0
ax0  lda #5
      jsr waitx
      lda #0
ax1  lda crdtsc
-    sta $d850,x
      inx
      cpx #$28
      bne -
      inc ax1+1
      lda ax1+1
      cmp #$10
      bne +
      jmp bx0
+    jmp ax0
bx0  lda #5
      jsr waitx
      lda #0
bx1  lda crdtsc
-    sta $d878,x
      inx
      cpx #$28
      bne -

```

```

        inc bx1+1
        lda bx1+1
        cmp #$10
        bne +
        jmp cx0
+       jmp bx0
cx0     ldx #5
        jsr waitx
        ldx #0
cx1     lda crdtsc
-       sta $d8a0,x
        inx
        cpx #$28
        bne -
        inc cx1+1
        lda cx1+1
        cmp #$10
        bne +
        jmp crdend
+       jmp cx0
crdend  ldx #$a0
        jmp waitx
crkgame
;       .binary "filename.prg",2 ; put cracked game here
        .byte $00

;-----
;       Some binaries and includes to C64 RAM
;-----
*= $0c00
.include "fonts1x1.txt"
*= $2000
.binary "fonts1x4.prg",2
* = $1000
.binary "music.sid", $7e
* = $2800
logohf .binary "hf.prg",2
* = $0f00
crdtsc .byte $00,$09,$06,$0b,$02,$08,$04,$0c,$0e
        .byte $0a,$05,$0f,$03,$0d,$07,$01
* = $3500

;-----
; Synchronisations with music and colors data
;-----
pcol2  .byte $01,$01,$01,$07,$0d,$03,$0f,$05,$0a,$0e
        .byte $0c,$04,$08,$02,$0b,$06,$09,$00
        .byte $00,$00,$00,$00,$00,$00

```

```

        .byte $01,$01,$01,$07,$0d,$03,$0f,$05,$0a,$0e
        .byte $0c,$04,$08,$02,$0b,$06,$09,$00
        .byte $00,$00,$00,$00,$00,$00
pcol2a .byte $01,$0f,$0f,$0f,$0f,$0f,$0f,$0f
        .byte $0f,$0f,$0f,$0f,$0f,$0f,$0f,$0f
        .byte $0f,$0f,$0f,$0f,$0f,$0f,$0f,$0f
        .byte $0f,$0f,$0f,$0f,$0f,$0f,$0f,$0f
        .byte $0f,$0f,$0f,$0f,$0f,$0f,$0f,$0f
        .byte $00,$00,$00,$00,$00,$00,$00,$00
        .byte $00,$00,$00,$00,$00,$00,$00,$00
        .byte $00,$00,$00,$00,$00,$00,$00,$00
        .byte $00,$00,$00,$00,$00,$00,$00,$00
        .byte $00,$00,$00,$00,$00,$00,$00,$00
        .byte $00,$00,$00,$00,$00,$00,$00,$00
        .byte $00,$00,$00,$00,$00,$00,$00,$00
        .byte $00,$00,$00,$00,$00,$00,$00,$00
        .byte $00,$00,$00,$00,$00,$00,$00,$00
        .byte $00,$00,$00,$00,$00,$00,$00,$00
        .byte $00,$00,$00,$00,$00,$00,$00,$00
        .byte $00,$00,$00,$00,$00,$00,$00,$00
dcolor .byte $00,$0d,$09,$0f,$08,$0c,$00,$03
        .byte $0b,$06,$05,$02,$0e,$07,$04,$0a
pcol1  .byte $00
pcol1a .byte $00
plpos  .byte $00
scpos  .byte $00
starspd .byte $04,$00,$06,$00,$07,$00,$06,$00,$03,$00,$07,$00,$05,$00,$04,$00

```