Erdoğan Tan

Kimden: "Dirk Küppers" <dirk@kueppers.org>
Kime: "Erdoğan Tan" <erdogantan@superonline.com>
Gönderme tarihi: 08 Aralık 2016 Perşembe 13:18
Ek: OMNICENT.ASM; SNC_OMNI.ASM
Konu: Re: OMNISCENT source code

Hi Erdogan,

i've found the sources on a 16 year old backup cd, and was able to recover the data.
Attached you will find the original Sourcecode of omnicent and the unpacking source.
By now i've really not much time to support you by improving the code.

If you have any questiosn regarding the sourcecode, feel free to contact me.

Best regards

Dirk

Am 06.12.2016 um 14:27 schrieb Erdoğan Tan:

> Hi Dirk,
>
> Thank you for your answer
>
> I have attached, a zip file which contains
>
> 1) original snc_omni.com
> 2) uncompressed and disassembled (by IDA pro free) source code of SNC_OMNI.COM
> (decompression method: DOXBOX DEBUG, Unpack packed DOS binaries with DOSBox.pdf)
> I think recompiled version runs successfully.
> 3) TRDOS 386 source code draft of SNC_OMNI (snc_omni.s) -defective-
> (ASSEMBLER: NASM 2.11, command: nasm snc_omsi.s -l snc_omni.txt -o SNC_OMNI.PRG)
>
> running in TRDOS 386: just writing SNC_OMNI at command prompt (you can try other samples before improving/correcting SNC_OMNI.PRG)
>
> 4) a.img: LATEST TRDOS 386 Bootable floppy disk image with sample program (demo) files...
> - runs with
> - physical drive A: (i use rawwrite.exe for that)
> - any version of Oracle VirtualBox
> - BOCHS version 2.6.0 (recent versions have a problem about virtual floppy drive motor timing)
>
> TRDOS 386 kernel loads PRG (program) files as a new process (sysexec system call) and with a new page directory.
> (Command Interpreter of TRDOS 386 is a part of the KERNEL, i will start at the end of STARTUP sequence of the OS kernel. "?" is command reminder command - MSDOS like commands-)
>
> -ALL PRG files are a FLAT MEMORY executable files (no segmentation, no exe header...), available memory for programs: 4GB - 4MB (with flat memory addressing method, with paging)
> just as an MSDOS COM file but 32 bit code and programs starts at offset 0 (EIP=0)
> /// important note: program's source code list points to running memory address (EIP), directly... (snc_omni.txt) -very usefull for debugging- nasm snc_omni.s -l snc_omni.txt (source code list)
> PRG files run via TRDOS 386 system calls and also some other interrupts (video, keyboard, IO etc.)
> PRG files run in ring 3 and access IO ports via io(ctl) interrupt...
> (If you will try it please download files at http://www.singlix.com/trdos/386/
>
> kernel (ring 0, uses virtual pages = physical pages), user (programs) user virtual pages
> all user programs starts at offset 0 (first 4 MB -1st page dir entry- of the user's page directory points to KERNEL space (4MB)
> this is for interrupts and system calls... But, user's segment selectors have a base address as start of 5th MB)
>
> ++
>
> - TRDOS_kernel_06_11_2016.zip
>
> To access VGA memory fin the program, "direct access to vga memory" system calls (is used...
>
> I will be glad if you improve the code (you can compile it with NASM or MASM,
> as i have tried to explain it above, you can use equivalent systme calls...)
>
> If you have not time to improve TRDOS 386 version of the SNC_OMNI.COM
> it will be usefull for me if you get MSDOS version simpler (C structures, -complex pushes, enters, leaves- must be eliminated)
> If prefer to use registers or minimum levels of data structure (pointed by EBX, ESI or EDI) while entering a subroutine, no push-pops... (it is easy for me.. Also faster...)
>
> TRDOS 386 syscall method: (INT 40h = system call/interrupt -also there are other intrerrupts like as IBM PC ROMBIOS interrupts, INT 31h video, 32h keyboard, 33h disk, 34h IOCTL, 35h date&time)
> Entry:
> EAX = system call number
> EBX = argument 1 value or address
> ECX = argument 2 value or address
> EDX = argment 3 value or address
> Return: EAX = return value (some systems calls also returns with EBX, ECX, EDX values in addtition to EAX value)
>
> Note: You can share SNC_OMNI source code in INTERNET... Still it is one of most famous intros in minds from DOS days...
>
> If you compile snc_omni.prg, a.img (TRDOS 386, 6/11/2016) has FAT12 format, you can open it in windows OS (for example: as ImDisk virtual disk drive) and copy SNC_OMNI.PRG to into root directory or
> into a FAT32 fs directory of a physical (PATA) or virtual box -bochs- (PATA) drive. (TRDOS 386 will know/use FAT16, FAT32 disk if physical drive is a PATA drive... SATA drive recognization is not ready yet...)
>
> Best regards...
> Erdogan Tan
>
> ++
>
> Sample: (stars32.s)
>
> [Bits 32] ; Protected Mode (TRDOS 386) Program
>
> [org 0] ; TRDOS 386 PRG File

```
START:
   ; DIRECT VGA MEMORY ACCESS
   ;xor    ebx, ebx
   mov     bh, 5 ; Direct access/map to VGA memory (0A0000h)
   ;mov    eax, _video ; 1Fh
   mov     al, 1Fh ; sys _video ; TRDOS 386 Video functions
   int     40h   ; TRDOS 386 system call

   ; eax = 0A0000h
   and     eax, eax
   jz      terminate ; error (eax = 0)

   mov     ax,0013h              ;set vid mode 320x200x256 graph
   ;int    10h
   int     31h ; TRDOS 386 - Video interrupt

   mov     edx,Palette
   mov     ax,1012h              ; WRITE palette
   mov     bx,0
   mov     cx,256                ;write entire palette
   ;int    10h                   ;doesn't matter if we didnt define it all
   int     31h ; TRDOS 386 - Video interrupt

StarLoop:
   call    MakeStar       ;make stars 2x as thick
   call    MakeStar

   mov     dx,3dah
   mov     ah, 0 ; in (byte)
Vrt:
   ;in     al,dx
   int     34h ; TRDOS 386 - IOCTL interrupt
   test    al, 8
   jnz     short Vrt  ; Wait until Verticle Retrace starts
NoVrt:
   ;in     al,dx
   int     34h ; TRDOS 386 - IOCTL interrupt
   test    al, 8
   jz      short NoVrt  ; Wait until Verticle Retrace ends

   call    DisplayStars

   mov     ah,1            ;check to see if a char is ready
   ;int    16h
   int     32h ; TRDOS 386 Keyboard interrupt
   jz      short StarLoop  ;nope, continue

   mov     ah,0
   ;int    16h             ;get the character & put in AX
   int     32h ; TRDOS 386 Keyboard interrupt
```

----- Original Message -----
From: Dirk Küppers
To: Erdoğan Tan
Sent: Tuesday, December 06, 2016 12:44 AM
Subject: Re: OMNISCENT source code

Hi Erdogan,

Omiscent was written and developed in pure Assembler in TASM (or maybe MASM ?).
The Data-Structs are real ASM-Structs (i think most people did not even know they exists).
The Main Structures are the Sound, the World map and the base points for the course of the spaceship.
There are also some smaller Images, but most of the GFX is calculated with the help of fractals.
The Font is extracted from VGA-Memory.
The Demo uses a binary Version of Adaptive Arithmetic Coding for compressing. Unfortunately the decoder had a small bug and did not work for all files.
I have to look for the Sources, but i am not sure, if i still have them and this may take some days.
If you have a decompressed, disassembled version, than most of the work is already done.
Maybe i can help you with the disassembled files in the case i cannot find the original ones ?

Best Regards

Dirk

Am 05.12.2016 um 22:49 schrieb Erdoğan Tan:

> Hi, Dirk
> I am writing TRDOS 386 operating system
> (32 bit MSDOS like operating system, but simple and "flat memory" type,
> protected mode oparting systme with paging)
> For VGA ns ystsme calls test purposes,
> I have tried to port old msdos demos to TRDOS, usually i have successed
> with small files but
> even if i have uses IDA pro disaasbler and then i have successed to extract
> your demo program to (uncompressed format),
> i have failed to recompile it for 32 bit dos. Because of very complex data structures
> (difficulty is coming from c language and file -3d format- structures).
>
> Could you please send me source code of this demo if still you have it?
> my website: www.singlix.com/trdos/trdos386.html
>
> Regards...
>
> Erdogan Tan

Sürüm: 2016.0.7924 / Virüs Veritabanı: 4728/13542 - Sürüm Tarihi: 06.12.2016
