Note: This API call is for DOS and Win16 personality only. Use Family API for portability.

2018/09/07 05:04 · prokushev · 0 Comments

DOSKRNL

DOSKRNL is a specialized FreeDOS kernel changed to support DOS in MVM. Mostly all FAT code removed as well as CONFIG.SYS parser. So it smaller in compare with original FreeDOS kernel. Also image format changed from exe to raw binary because slightly different load logic.

Initialization

DOSKRNL loaded at address differ from standard DOS 0060:0000 address. On test environment it was loaded to 0280:00, but OS/2 v 2.0 Kernel Design Guide says it is at 0070:0000. Later documentation says nothing about loading address. osFree DOSKRNL expects unknown load address, but in low address of conventional memory. Refer OS/2 v 2.0 Kernel Design Guide for memory map and initialization information.

On DOSKRNL startup registers are following:

- CS:IP - starting point
- SS:BP - pointer to the DOSKRNL init structure (also referenced as VDOS init in IBM documents)
- SS:SP - stack pointer (size around 800 bytes)

Interrupts disabled.

DOSKRNL init structure:

Offset	Size	Description
0	2	First free segment after DOSKRNL
2	2	Size of memory - first free segment (paragraphs)
4	2	Size of init area (paragraphs)
6	2	Value of BREAK setting
8	2	Value of DOS setting
10	4	Far pointer to list of DOS DEVICE setting
14	4	Far pointer to SHELL (filepath only)
18	4	Far pointer to SHELL (arguments)
22	4	Far pointer to linked list of VDD
26	1	Current drive (1-A, 2-B, 3-C, ...)
27	1	Boot drive (1-A, 2-B, 3-C, ...)
28	1	Last drive (1-A, 2-B, 3-C, ...)
???	???	???

DEVICES is ASCIIZ string with list of DOS devices to be loaded divided by 0AH

SHELL is ASCIIZ string.

SHELL arguments is ASCIIZ string (first byte is a string length, string ends with 0dh)

VDD is linked list of structures:

VDDs linked list in standard DOS Device drivers format. Entry can be placed in HMA, so need to be sure A20 enabled for such entries. Named as DOSDDTYPE in DDK

offset	Size	Description
0	4	Pointer to next device (Must be set to -1 for last device)
4	2	Attributes
		Bit 15 = 1 if char device 0 if blk
		if bit 15 is 1
		Bit 0 = 1 if Current STDIN device
		Bit 1 = 1 if Current STDOUT output
		Bit 2 = 1 if Current NULL device
		Bit 3 = 1 if Current CLOCK device
		Bit 4 = 1 if SPECIAL
		Bit 14 is the IOCTL bit
		Bit 13 is the NON IBM FORMAT bit
6	2	Pointer to Device strategy entry point
8	2	Pointer to Device interrupt entry point
10	8	character device name field. Character devices set a device name. For block devices the first byte is the number of units

Memory map on DOSKRNL start:

Init area	[BP+0+BP+2-BP+4]:0
Free	[BP+0]:0
DOSKRNL	???:0
Some data (VDD entries was found here)	???
CMOS Data	40:0
Interrupt vectors	0:0

As in most DOS kernels, DOSKRNL moves initialization code to higher conventional memory. But uses init structure, not BIOS INT 12h, for memory information.

DOSKRNL search and initialize XMS driver and moves some parts to HMA. Most other initialization things is same. DOSKRNL initializes standard device drivers and adds and initializes VDDs, passed via init structure. Some things (at least COMMAND.COM, device drivers) also passed via init structure. not CONFIG.SYS. At the present time interface not investigated well.

CONFIG.SYS

CONFIG.SYS device information and some other settings not parsed by DOSKRNL, but passed via init structure. Here list of DOSKRNL options:

- RMSIZE
- DEVICE
- SHELL
- DOS
- BREAK
- FCBS
- LASTDRIVE

todo add more info here

Also most of these setting can be adjusted via DOS Properties (aka DOS Settings). Refer OS/2 Version 2.0 Volume 2: DOS and Windows Environment for more information.

Properties

- COM_DIRECT_ACCESS
- COM_HOLD
- COM_RECEIVE_BUFFER_FLUSH
- COM_SELECT
- DOS_AUTOEXEC
- DOS_BACKGROUND_EXECUTION
- DOS_BREAK
- DOS_DEVICE
- DOS_FCBS
- DOS_FCBS_KEEP
- DOS_FILES
- DOS_HIGH
- DOS_LASTDRIVE
- DOS_RMSIZE
- DOS_SHELL
- DOS_STARTUP_DRIVE
- DOS_UMB
- DOS_VERSION
- DPMI_DOS_API
- DPMI_MEMORY_LIMIT
- DPMI_NETWORK_BUFF_SIZE
- EMS_FRAME_LOCATION
- EMS_HIGH_OS_MAP_REGION
- EMS_LOW_OS_MAP_REGION
- EMS_MEMORY_LIMIT
- HW_NOSOUND
- HW_ROM_TO_RAM
- HW_TIMER
- IDLE_SECONDS
- IDLE_SENSITIVITY
- INT_DURING_IO
- KBD_ALTHOME_BYPASS
- KBD_BUFFER_EXTEND
- KBD_CTRL_BYPASS
- KBD_RATE_LOCK

- MEM_EXCLUDE_REGION
- MEM_INCLUDE_REGION
- MOUSE_EXCLUSIVE_ACCESS
- PRINT_SEPARATE_OUTPUT
- PRINT_TIMEOUT
- SESSION_PRIORITY
- VIDEO_8514A_XGA_IOTRAP
- VIDEO_FASTPASTE
- VIDEO_MODE_RESTRICTIONS
- VIDEO_ONDEMAND_MEMORY
- VIDEO_RETRACE_EMULATION
- VIDEO_ROM_EMULATION
- VIDEO_SWITCH_NOTIFICATION
- VIDEO_WINDOW_REFRESH
- XMS_HANDLES
- XMS_MEMORY_LIMIT
- XMS_MINIMUM_HMA

API

According different sources IBM DOSKRNL implements API on level of MS-DOS 5.x. osFree DOSKRNL supports API as in FreeDOS kernel. More information about implemented functions available here. DOSKRNL also implements some extensions to API mostly aimed to communicate with host system.

Supervisor calls

DOSKRNL uses SVC interface to call host functions.

From:
https://osfree.org/doku/ - **osFree wiki**

Permanent link:
https://osfree.org/doku/doku.php?id=en:docs:kernel:doskrnl

Last update: **2024/05/25 02:43**