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 because different load logic.

Initialization

DOSKRNL loaded at address differ from standard DOS 0060:0000 address. osFree DOSKRNL expect unknown load address, but in low address of conventional memory.

On DOSKRNL startup registers are following:

- CS:IP - starting point
- SS:BP - pointer to the DOSKRNL init structure
- SS:SP - stack pointer (size around 800 bytes)

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, ...)
????	???	????

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

SHELL is ASCIIZ string.

SHELL arguments is ASCIIZ string (seems, first byte is as string length)

VDD is linked list of structures:

VDDs linked list (in init area) in standard DOS Device drivers format

offset	Size	Description
0	4	Pointer to next device (Must be set to -1 for last device)
		Attributes
		Bit 15 = 1 if char device 0 if blk
		if bit 15 is 1
		Bit 0 = 1 if Current sti device
4	2	Bit 1 = 1 if Current sto output

offset	Size	Description
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	???
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

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.

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://ftp.osfree.org/doku/ - **osFree wiki**

Permanent link:
https://ftp.osfree.org/doku/doku.php?id=en:docs:kernel:doskrnl&rev=1702470585

Last update: **2023/12/13 12:29**