DOSKRNL

DOSKRNL is a specialized DOS kernel changed to support DOS in MVM.

Startup

On DOSKRNL start SS:BP contains some structure which contains initial information for DOS kernel. Exact structure format not know. SS:SP contains stack pointer (initially point to end of 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)
????	???	Unknown
22	4	Pointer to linked list of VDD
25	1	Current drive
26	1	Boot drive

Memory map on DOSKRNL start:

Init area	[BP+0+BP+2-BP+4]:0
Free	[BP+0]:0
DOSKRNL	60:0
CMOS Data	40:0
Interrupt vectors	0:0

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

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

offset	Size	Description
0	DWORD	Pointer to next device (Must be set to -1 for last device)
4	WORD	Attributes
	Bit 15 = 1 if char device 0 if blk	
	if bit 15 is 1	
	Bit 0 = 1 if Current sti device	
	Bit 1 = 1 if Current sto output	
	Bit 2 = 1 if Current NUL device	
	Bit 3 = 1 if Current CLOCK dev	
	Bit 4 = 1 if SPECIAL	
	Bit 14 is the IOCTL bit	
	Bit 13 is the NON IBM FORMAT bit	
6	WORD	Pointer to Device strategy entry point
8	WORD	Pointer to Device interrupt entry point

offset	Size	Description
0ah	8-BYTE	character device name field. Character devices set a device name. For block devices the first byte is the number of units

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

todo add more info here

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.