This is part of **Win16 API** which allow to create versions of program from one source code to run under OS/2 and Win16. Under OS/2 program can be running under Win-OS/2 if program is Windows NE executable, and with help on Windows Libraries for OS/2, if it is OS/2 NE executable. Here is a WLO to OS/2 API mapping draft

2021/09/01 04:23 · prokushev · 0 Comments

Windows kernel for 8086 (KERNEL.EXE) doesn't uses DPMI or LDT for memory management. KERNEL.EXE for later versions of Windows emulates selectors functions via segments (like FAPI does), but not so good (mostly functions returns errors). KRNL286.EXE and KRNL386.EXE uses direct LDT manipulations. Kernels for WOW uses DPMI functions. For now, osFree tries to reuse only DPMI (in future it is possible to implement direct LDT manipulation, may be).

Note minimal checked windows version is 1.03.

Ordinal	Name	Description	Status			Version	ECMA-234
			Real	Std	Enh		
001	FatalExit	Display the current state of Windows and prompts for instructions on how to proceed				1.03	Yes
002	ExitKernel						
003	GetVersion	Return the current version of Windows				1.03	Yes
004	LocalInit	Initialize local heap					Yes
005	LocalAlloc	Allocate wBytes bytes of memory from the local heap				1.03	Yes
006	LocalRealloc	Reallocate the local memory block				1.03	Yes
007	LocalFree	Free the local memory block				1.03	Yes
008	LocalLock	Lock the local memory block				1.03	Yes
009	LocalUnlock	Unlock the local memory block				1.03	Yes
010	LocalSize	Retrieve the current size, in bytes, of the local memory block				1.03	Yes
011	LocalHandle						Yes
012	LocalFlags	Return information about the specified local memory block				1.03	Yes
013	LocalCompact	Generate free bytes of memory by compacting, if necessary, the module's local heap				1.03	Yes
014	LocalNotify						
015	GlocalAlloc	Allocate memory from the global heap				1.03	Yes

Ordinal	Name	Description	Status			Version	ECMA-234
			Real	Std	Enh		
016	GlobalRealloc	Reallocate the global memory block				1.03	Yes
017	GlobalFree	Free the global memory block				1.03	Yes
018	GlobalLock	Retrieve the absolute memory address of the global memory block				1.03	Yes
019	GlobalUnlock	Unlock the global memory block				1.03	Yes
020	GlobalSize	Retrieve the current size, in bytes, of the global memory block				1.03	Yes
021	GlobalHandle						Yes
022	GlobalFlags	Return information1 about the specified global memory block				1.03	Yes
023	LockSegment						
024	UnlockSegment						
025	GlobalCompact	Generate free bytes of global memory by compacting, if necessary, the system's global heap				1.03	Yes
026	GlobalFreeAll						
027	GetModuleName						
028	GlobalMasterHandle						
029	Yield	halts the current task and starts any waiting task				1.03	Yes
030	WaitEvent						
031	PostEvent						
032	SetPriority						
033	LockCurrentTask						
034	SetTaskQueue						
035	GetTaskQueue						
036	GetCurrentTask	Return the handle of the currently executing task				1.03	Yes
037	GetCurrentPDB	Return current PDB					
038	SetTaskSignalProc						
039	SetTaskSwitchProc						
040	SetTaskInterchange						
041	EnableDOS						
042	DisableDOS						
043	IsScreenGrab						
044	BuildPDB						
045	LoadModule	Load module					Yes
046	FreeModule	Release module					Yes

Ordinal	Name	Description	Status			Version	ECMA-234
			Real	Std	Enh		
047	GetModuleHandle	Retrieve the module handle of the specified module				1.03	Yes
048	GetModuleUsage	Return the reference count of a given module				1.03	Yes
049	GetModuleFileName	Retrieve the name of the executable file from which the specified module was loaded				1.03	Yes
050	GetProcAddress	Retrieve the memory address of the function whose name				1.03	Yes
051	MakeProcInstance	Bind the data segment of the module instance specified to the function pointed				1.03	Yes
052	FreeProcInstance	Frees the function specified from the data segment				1.03	Yes
053	CallProcInstance						
054	GetInstanceData	Copy data from a previous instance of an application into the data area of the current instance				1.03	Yes
055	Catch	Catch the current execution environment and copy it the the buffer				1.03	Yes
056	Throw	Restore the execution environment to the values saved in the buffer				1.03	Yes
057	GetProfileInt	Retrieve the value of an integer key from the the Windows initialization file				1.03	Yes
058	GetProfileString	Copy a character string from the user profile into the buffer				1.03	Yes
059	WriteProfileString	Copy the character string into the Windows initialization file				1.03	Yes
060	FindeResource	Determine the location of a resource in the specified resource file				1.03	Yes
061	LoadResource	Load a resource from the executable file associated with the module				1.03	Yes

Ordinal	Name	Description	Status			Version	ECMA-234
			Real	Std	Enh		
062	LockResource	Retrieve the absolute memory address of the loaded resource				1.03	Yes
063	FreeResource	Remove a loaded resource from memory by freeing the allocated memory occupied by that resource				1.03	Yes
064	AccessResource	Open the specified resource file and moves the file pointer to the beginning of the specified resource				1.03	
065	SizeOfResource	Supply the size in bytes of the specified resource				1.03	Yes
066	AllocResource	Allocate uninitialized memory for the passed resource				1.03	Yes
067	SetResourceHandle	Set up a function to load resources				1.03	Yes
068	InitAtomTable	Initialize an atom hash table and set its size				1.03	
069	FindAtom	Search the atom table for the character string				1.03	
070	AddAtom	Add the character string to the atom table				1.03	
071	DeleteAtom	Delete an atom				1.03	
072	GetAtomName	Retrieve a copy of the character string associated with atom				1.03	
073	GetAtomHandle						
074	OpenFile	Create, open, reopen, or delete a file				1.03	Yes
075	OpenPathname						
076	DeletePathname						
077	AnsiNext	Move to the next character in a string				1.03	Yes
078	AnsiPrev	Move to the previous character in a string				1.03	Yes
079	AnsiUpper	Convert a string or a character to upper case				1.03	Yes
080	AnsiLower	Convert the given string to lower case				1.03	Yes
081	_lclose	Close the file described by the file handle					Yes
082	_lread	Read a specified number of bytes from a file into memory					Yes

Ordinal	Name	Description	Status			Version	ECMA-234
			Real	Std	Enh		
083	_lcreat	Create and open a file, described by FileName, for reading and/or writing					Yes
084	_llseek	Move the current file position pointer of the file					Yes
085	_lopen	Open a file					Yes
086	_lwrite	Write a specified number of bytes of memory to a file					Yes
087	lstrcmp	Compare two strings					Yes
088	lstrcpy	Copy the contents from one string to another					Yes
089	lstrcat	Concatecate the contents of two strings					Yes
090	lstrlen	Determines the length of the string					Yes
091	InitTask						
092	GetTempDrive	Return a letter specifying the optimal drive for a temporary file				1.03	Yes
093	GetCodeHandle	Return the handle of the code segment containing the function pointed				1.03	
094	DefineHandleTable						
095	LoadLibrary	Load the library module contained in the specified file and returns a handle to the loaded module				1.03	Yes
096	FreeLibrary	Free memory occuped by library when module reference count equal to zero				1.03	Yes
097	GetTempFilename	Create a temporary filename				1.03	Yes
098	GetLastDiskChange						
099	GetLpErrMode						
100	ValidateCodeSegments						
101	NoHookDosCall						
102	Dos3Call						
103	NetbiosCall						
104	GetCodeInto						
105	GetExeVersion						
106	SetSwapAreaSize						
107	SetErrorMode						Yes
108	SwitchStackTo						
109	SwitchStackBack						

Ordinal	Name	Description	Status			Version	ECMA-234
			Real	Std	Enh		
110	PatchCodeHandle						
111	GlobalWire						
112	GlobalUnwire						
113	__AHSHIFT						
114	__AHINCR						
115	OutputDebugString						Yes
116	InitLib						
117	OldYield						
118	GetTaskQueueDS						
119	GetTaskQueueES						
120	UndefDynLink						
121	LocalShrink						Yes
122	IsTaskLocked						
123	KbdRst						
124	EnableKernel						
125	DisableKernel						
126	MemoryFreed						
127	GetPrivateProfileInt						Yes
128	GetPrivateProfileString						Yes
129	WritePrivatePRofileString						Yes
130	FileCdr						
131	GetDOSEnvironment						
132	GetWinFlags						Yes
133	GetExePtr						
134	GetWindowsDirectory						Yes
135	GetSystemDirectory						Yes
136	GetDriveType						Yes
137	FatalAppExit						Yes
138	GetHeapSpaces						
139	DoSignal						
140	SetSigHandler						
141	InitTask1						
142	GetProfileSectionNames						
143	GetPrivateProfileSectionNames						
144	CreateDirectory						
145	RemoveDirectory						
146	DeleteFile						
147	SetLastError						
148	GetLastError						
149	GetVersionEx						
150	DirectedYield						Yes
151	WinOldApCall						
152	GetNumTasks						Yes

Ordinal	Name	Description	Status			Version	ECMA-234
			Real	Std	Enh		
153	???						
154	GlobalNotify						Yes
155	GetTaskDS						
156	LimitEMSPages						
157	GetCurPID						
158	IsWinOldApTask						
159	GlobalHandleNoRip						
160	EMSCopy						
161	LocalCountFree						
162	LocalHeapSize						
163	GlobalLRUOldest						Yes
164	GlobalLRUNewest						Yes
165	A20Proc						
166	WinExec						Yes
167	GetExpWinVer						
168	DirectResAlloc						
169	GEtFreeSpace						Yes
170	AllocCSToDSAlisa						
171	AllocDSToCSAlias						
172	AllocAlias						
173	__ROMBIOS						
174	__A000H						
175	AllocSelector						
176	FreeSelector						
177	PrestoChangoSelector						
178	__WINFLAGS						
179	__D000H						
180	LongPtrAdd						
181	__B000H						
182	__B800H						
183	__0000H						
184	GlobalDOSAlloc						
185	GlobalDOSFree						
186	GetSelectorBase						
187	SetSelectorBase						
188	GetSelectorLimit						
189	SetSelectorLimit						
190	__E000H						
191	GlobalPageLock						
192	GlobalPageUnlock						
193	__0040H						
194	__F000H						
195	__C000H						
196	SelectorAccessRights						

Ordinal	Name	Description	Status			Version	ECMA-234
			Real	Std	Enh		
197	GlobalFix						Yes
198	GlobalUnfix						Yes
199	SetHandleCount						Yes
200	ValidateFreeSpaces						
201	ReplaceInst						
202	RegisterPtrace						
203	DebugBreak						
204	SwapRecording						
205	CVWBreak						
206	AllocSelectorArray						
207	IsDBCSLeadByte						Yes
310	LocalHandleDelta	Set the number of handle table entries to be allocated when the local heap manager runs out of handle table				1.03	
320	IsTask						Yes
323	IsROMModule						
324	LogError						
325	LogParamError						
326	IsROMFile						
334	IsBadReadPtr						Yes
335	IsBadWritePtr						Yes
336	IsBadCodePtr						Yes
337	IsBadStringPtr						Yes
347	IsBadHugeWritePtr						Yes
348	hmemcpy						
349	_hread						
350	_hwrite						
353	lstrcpyn						Yes
354	GetAppCompatFlags						
355	GetWinDebugInfo						
356	SetWinDebugInfo						

Not found in exports (check other module later): GlobalDiscard 1.03 LocalDiscard 1.03 LocalFreeze 1.03 LocalMelt 1.03 LockData 1.03 UnLockData 1.03 SetPriority 1.03 AddFontResource 1.03 RemoveFontResource 1.03

Group	Functions
Module manager	GETVERSION GETMODULEHANDLE GETMODULEUSAGE GETMODULEFILENAME GETPROCADDRESS MAKEPROCINSTANCE FREEPROCINSTANCE GETINSTANCEDATA CATCH THROW GETCODEHANDLE LOADLIBRARY

Group	Functions
Memory Manager	GlobalAlloc GlobalCompact GlobalDiscard GlobalFree GlobalLock GlobalReAlloc GlobalSize GlobalUnlock GlobalFlags LocalAlloc LocalCompact LocalDiscard LocalFree LocalLock LocalFreeze LocalMelt LocalReAlloc LocalSize LocalUnlock LocalHandleDelta LockData UnlockData LocalFlags
Task Scheduler	GetCurrentTask Yield SetPriority
Resource Manager	AddFontResource RemoveFontResource LoadBitmap LoadCursor LoadIcon LoadMenu LoadString LoadAccelerators FindResource LoadResource AllocResource LockResource FreeResource AccessResource SizeofResource SetResourceHandler
String Translation	AnsiUpper AnsiLower AnsiNext AnsiPrev
Atom Manager	InitAtomTable AddAtom DeleteAtom FindAtom GetAtomName
Windows Initialization File	GetProfileInt GetProfileString WriteProfileString
Debugging	FatalExit
File I/O	OpenFile GetTempFileName GetTempDrive
Registry	RegOpenKey RegCreateKey RegCloseKey RegDeleteKey RegSetValue RegQueryValue RegEnumKey

2022/11/17 15:22 · prokushev · 0 Comments