

[Q]: Снова про недокументированные функции

[A]: Julius Goryavsky (2:5030/16.32)

Q> Покопавшись в bseord.h обнаружил весьма интересные ф-ции:

Q> Win32SwitchProgramRegister 156 Q> Prf32QueryDefinition 111 - это есть в pmsl.h, Q> Prf32AddProgram 109 но интересно имеет ли это Q> Prf32RemoveProgram 104 отношение к содержимому Q> Prf32ChangeProgram 110 WPS`овских фолдеров.

Про эти не знаю...

Q> Dos32OpenChangeNotify 440 - Вот это очень похоже на Q> Dos32ResetChangeNotify 441 мониторинг доступа к Q> Dos32CloseChangeNotify 442 файлам ?

Так и есть. Эти функции уведомляют об изменениях в

контролируемом каталоге. Эти функции активно использует WPS. Например, если создать каталог C:\DESKTOP\TEST то на desktop-е вскоре появится папка с именем test, хотя опроса содержимого c:\desktop с помощью dosfindfirst/next он не делает.

Q> Может кто-нибудь поделиться информацией об их вызове и Q> назначении.

Вот описание всяких недокументированных функций:

APIRET APIENTRY DosTmrQueryTime (QWORD Time)

Доступ к _очень_ точному аппаратному таймеру. Его

эффективная точность близка к _микро_секунде. Time - 8 байт содержащих текущее время в квантах системного таймера. См. DosTmrQueryFreq. Я использую эту функцию для таймирования процедур исполнение которых длится 200-400 тактов и получаю довольно точные результаты.

APIRET APIENTRY DosTmrQueryFreq (ULONG Freq)

Определить частоту системного таймера. Freq - частота

системного таймера. Приблизительно 1193182 Гц.

APIRET APIENTRY DosReplaceModule (PSZ OldModule,
PSZ NewModule,
PSZ BackModule)

Позволяет заменить загруженный программный модуль новой

копией. Эту функцию используют service pack-и и selective install для замены старых dll новыми. Например, когда при инсталляции нового видеодрайвера надо заменить dspres.dll.

OldModule - имя файла с заменяемой dll или exe.
NewModule - имя нового файла, который копируется на место старого. Может быть NULL.
BackModule - Имя backup-файла в который переименовывается старый модуль. Может быть NULL.

В свою очередь эта функция использует недокументированную функцию Dos32ICacheModule...

APIRET APIENTRY DosDumpProcess (ULONG Flag, ULONG Drive, ULONG pid)

Аналог оператора DUMPPROCESS в CONFIG.SYS.

Если Flag = 0 то запрещает дампование обломившихся

процессов на диск, если Flag = 1 то разрешает. Drive определяет номер диска (начиная с нуля) на который дампуот память обломившихся процессов. Если Flag = 2 то pid определяет процесс подлежащий дампованию.

APIRET APIENTRY DosForceSystemDump (ULONG Reserved)

Произвести дампование всей системной памяти на диск

специфицированный в операторе TRAPDUMP файла CONFIG.SYS.

APIRET APIENTRY DosQueryABIOSSupport(ULONG reserved)

Возвращает различные флаги характеризующие тип

системной шины и поддержки ABIOS:

бит 0: если равен 1 то шина - Micro Channel Architecture. бит 1: если равен 1 то шина - EISA. бит 2: если равен 1 то ABIOS поддерживается. бит 3: если равен 1 то ABIOS существует.

Если возвращает 0h то ABIOS есть, если не ноль - нет или

Not Supported.

```
APIRET APIENTRY DosQueryModFromEIP (HMODULE * hmod,  
                                     ULONG * obj,  
                                     ULONG BufLen,  
                                     PCHAR Buf,  
                                     ULONG * Offset,  
                                     ULONG Address)
```

Определить модуль, в адресное пространство которого

попадает указанный адрес. Вход: Address и BufLen. Остальное - на выходе. Address - проверяемый адрес. BufLen - длина буфера для имени модуля которому принадлежит адрес, Buf - сам буфер. Obj - номер объекта памяти в который попал адрес, Offset - смещение в модуле. hmod - Handle модуля которому принадлежит адрес.

```
APIRET APIENTRY DosSuppressPopUps (ULONG Flag,  
                                    ULONG Drive)
```

Работает подобно оператору SUPPRESSPOPUPS в CONFIG.SYS.

Flag = 0 - Disable всплывающий экран с Help,Retry,End Process и т.п. (HARDERR.EXE), Flag = 1 - Enable. Drive содержит номер диска на который надо сливать описание сбоя при Disabled Pop-Ups.

```
APIRET APIENTRY DosVerifyPIDTID (ULONG pid, ULONG tid)
```

Определяет существует ли цепочка tid в процессе pid.

Если вернули 0h - все Ok, иначе цепочка не существует. Так как процесс не может существовать без цепочки 1, то вызов DosVerifyPIDTID(pid, 1) определяет жив ли процесс.

```
HAPP APIENTRY WinHAPPFromPID (ULONG pid)
```

Получить HAPP по PID. Если вернула 0h то Error.

```
HSWITCH APIENTRY WinHSWITCHFromHAPP
```

Получить HSWITCH по HAPP. Если вернула 0h то Error.

```
DosOpenChangeNotify  
DosCloseChangeNotify
```

DosResetChangeNotify

[A]: Peter Fitzsimmons

Долгая история... Позволяет определить факт каких-либо

изменений на диске. За счет использования этих функций WPS определяет возникновение или исчезновение файловых объектов. Могут использоваться для контроля за изменением состояния каталога в одном из окон Нортон и т.п... Вот английское описание:

Q> Does anybody know why the DosNotify.. functions are Q> omitted from the 32 bit API.

(Fyi: they are DosFindNotify...(), not just

DosNotify...()).

I can't even find these in my 1.x header files.

I do know that they are still there -- IFSs must support

them, and the WorkPlace shell actively uses these services in the IFSs I have written.

Try prototyping them yourself and using them.

[later] After perusing \ddk\h (A great place find

"undocumented" os/2 APIs), it appears that you may be correct - the DosFindNotify...() apis no longer exist as 32bit apis (but you can still import the 16bit ones if you like).

They appear to have been replaced by something better

(which probably uses the FindNotify IFS services underneath). You'll have to figure these out on your own (please report back here);but they don't look too hard:

```
#pragma pack(1)
```

```
typedef struct _CNPATH { /* CHANGENOTIFYPATH */
```

```
    ULONG    oNextEntryOffset;  
    ULONG    wFlags;  
    USHORT   cbName;  
    CHAR     szName[1];
```

```
} CNPATH; typedef CNPATH *PCNPATH;
```

```
typedef struct _CNINFO { /* CHANGENOTIFYINFO */
```

```
    ULONG    oNextEntryOffset;  
    CHAR     bAction;  
    USHORT   cbName;
```

```
CHAR    szName[1];
```

```
} CNINFO; typedef CNINFO *PCNINFO;
```

```
#pragma pack()
```

```
Equates for ChangeNotifyInfo baction field #define RCNF_FILE_ADDED 0x0001 #define  
RCNF_FILE_DELETED 0x0002 #define RCNF_DIR_ADDED 0x0003 #define RCNF_DIR_DELETED 0x0004  
#define RCNF_MOVED_IN 0x0005 #define RCNF_MOVED_OUT 0x0006 #define RCNF_CHANGED  
0x0007 #define RCNF_OLDNAME 0x0008 #define RCNF_NEWNAME 0x0009 #define  
RCNF_DEVICE_ATTACHED 0x000A #define RCNF_DEVICE_DETACHED 0x000B APIRET APIENTRY  
DosOpenChangeNotify(PCNPATH PathBuf, ULONG LogSize, PHDIR hdir, ULONG ulReserved); APIRET  
APIENTRY DosResetChangeNotify(PCNINFO LogBuf, ULONG BufferSize, PULONG LogCount, HDIR hdir);  
APIRET APIENTRY DosCloseChangeNotify(HDIR hdir);
```

From:

<https://www.osfree.org/doku/> - **osFree wiki**

Permanent link:

<https://www.osfree.org/doku/doku.php?id=ru:os2faq:os2prog:os2prog.004>

Last update: **2014/06/20 05:08**

