Таблица 1

|  |  |
| --- | --- |
|  | Функция для получения системной информации |
|  | Получить имя компьютера с помощью функции  BOOL GetComputerNameA(  LPTSTR lpBuffer, // Адрес буфера  LPDWORD nSize // Адрес размера буфера  ); |
|  | Получить имя компьютера с помощью функции  BOOL GetComputerNameA(  LPTSTR lpBuffer, // Адрес буфера  LPDWORD nSize // Адрес размера буфера  ); |
|  | Получить имя текущего каталога  DWORD GetCurrentDirectoryA(  DWORD nBufferLength, // Размер буфера в символах  LPTSTR lpBuffer // Адрес буфера для тек. каталога  ); |
|  | Получить имя текущего каталога  DWORD GetCurrentDirectoryA(  DWORD nBufferLength, // Размер буфера в символах  LPTSTR lpBuffer // Адрес буфера для тек. каталога  ); |
|  | Получить список значение переменной окружения PATH.  DWORD GetEnvironmentVariableA(  LPCTSTR lpName, // Адрес строки с именем переменной  LPTSTR lpBuffer, // Адрес буфера для значения переменной  DWORD nSize // Размер буфера в символах  ); |
|  | Получить имя активной раскладки клавиатуры  BOOL GetKeyboardLayoutNameA(  LPTSTR pwszKLID // Адрес буфера для раскладки  );  Примечание: Размер буфера не меньше 9 символов |
|  | Получить путь к текущему временному каталогу  DWORD GetTempPathA(  DWORD nBufferLength, // Размер буфера в символах  LPTSTR lpBuffer // Адрес буфера для временного каталога  ); |
|  | Получить каталог Windows  UINT GetWindowsDirectoryA(  LPTSTR lpBuffer, // Адрес буфера для имени каталога  UINT uSize // Размер буфера  );  Примечание: Размер буфера не меньше 260 символов |
|  | Получить имя системного каталога Windows  UINT GetSystemDirectoryA(  LPTSTR lpBuffer, // Адрес буфера для системного каталога  UINT uSize // Размер буфера  );  Примечание: Размер буфера не меньше 260 символов |
|  | Получитьимя текущего пользователся  BOOL GetUserNameA(  LPTSTR lpBuffer, // Адрес буфера для имени пользователя  LPDWORD nSize // Адрес размера буфера  ); |
|  | Получение числа тактов процессора после запуска системы  DWORD GetTickCount(VOID);  Значение возвращается в регистровой паре edx:eax |
|  | Функция создает или открывает каталог, физический диск, том, буфер консоли (CONIN$ или CONOUT$), устройство на магнитной ленте, коммуникационный ресурс, почтовый слот или именованный канал.  DWORD CreateFile(  LPCTSTR lpFileName, // имя файла  DWORD dwDesiredAccess, // режим доступа  DWORD dwShareMode, // совместный доступ  LPSECURITY\_ATTRIBUTES lpSecurityAttributes, // SD (дескр. защиты)  DWORD dwCreationDisposition, // как действовать  DWORD dwFlagsAndAttributes, // атрибуты файла  HANDLE hTemplateFile // дескр.шаблона файла  )  Функция возвращает дескриптор, который может быть использован для доступа к объекту. |
|  | The GetLogicalDriveStrings function fills a buffer with strings that specify valid drives in the system.  DWORD GetLogicalDriveStrings(  DWORD nBufferLength, // size of buffer  LPTSTR lpBuffer // address of buffer for drive strings  );  Функция возвращает имена логических дисков |
|  | The GetLogicalDriveStrings function fills a buffer with strings that specify valid drives in the system.  DWORD GetLogicalDriveStringsW(  DWORD nBufferLength, // size of buffer  LPTSTR lpBuffer // address of buffer for drive strings  );  Функция возвращает имена логических дисков |
|  | Получить каталог Windows в многопользовательском режиме  UINT GetSystemWindowsDirectoryA(  LPTSTR lpBuffer, // Адрес буфера для имени каталога  UINT uSize // Размер буфера  ); |
|  | Форматирует строку сообщения.  DWORD FormatMessage (  DWORD nSize, // size of buffer  LPTSTR lpBuffer // address of buffer for drive strings  ); |
|  | Функция GetLongPathNameA по короткому имени пути извлекает длинный путь  DWORD GetLongPathName(  LPCTSTR *lpszShortPath*, // строка пути с нулем в конце  LPTSTR *lpszLongPath*, // буфер короткой формы  DWORD *cchBuffer* // размер буфера короткой формы  ); |
|  | Функция FormatMessageA требует определения сообщения как вводимых данных. Определение сообщения может придти из буфера, который передается в функцию формирует строку сообщения.  DWORD FormatMessage (  DWORD nSize, // размер буфера выводимых данных  LPTSTR lpBuffer // указатель на буфер, который получает строку с завершающим нулем, устанавливающую форматированное сообщение  ); |
|  | GetLogicalDriveStrings  DWORD GetLogicalDriveStringsW(  DWORD nBufferLength, // входной параметр – размер строки  LPTSTR lpBuffer // выходной параметр – адрес буфера строки  );  Функция возвращает буфер строки из установленных логических дисков на компьютере |
|  | GetShortPathName  DWORD WINAPI GetShortPathName (  LPCTSTR lpszLongPath, //Путь строки  LPTSTR lpszShortPath, //Указатель на буфер для получения нулевым краткую форму пути, который указывает lpszLongPath  DWORD cchBuffer); Размер буфера, который указывает на lpszShortPath в TCHARs  Функция возвращает краткую форму пути по указанному пути. |
|  | PUSH OFFSET IDP  PUSH DWORD PTR [EBP+8]  CALL GetWindowThreadProcessId@8  Функция GetWindowThreadProcessId  Синтаксис:   |  | | --- | | DWORD GetWindowThreadProcessId(Wnd: HWND; lpdwProcessId : LPDWORD); |   Описание: Возвращает идентификатор процесса к которому принадлежит данное окно  Параметры:  Wnd: Идентификатор окна. lpdwProcessId : 32битное значение идентификатора процесса Возвращаемое значение: идентификатор нити  Как использовать данную функцию ??   |  | | --- | | **Var**  mProcessID,mThreadID : Dword;  **Begin**  ...*// Получение описателя в WD*  mThreadID:= GetWindowThreadProcessId(Wd,@mProcessID); *// Получения идентификаторов.*  ...  **End**; |   Эта функция хороша тем, что для всех окон одного приложения этот идентификатор будут един, если они запущены внутри одной нити (что зачастую и делается). То есть Вы можете разбить окна по нитям и /или по процессам, к которым они относятся. А так же Вы мож ете определять, какие окна данного приложения в настоящий момент доступны, видны, свернуты и т.д. |
| 22 | Смена генерируемых событий между кнопками мыши  BOOL SwapMouseButton(  BOOL fSwap //Если сменить, то передавать true  ); |
| 23 | Установить время между нажатиями мыши при двойном клике  BOOL SetDoubleClickTime(  UINT uInterval //Интервал времени  ); |
| 24 | Получить количество процессов в системе  WORD GetMaximumProcessorGroupCount(); |
| 25 | Установить имя компьютера  BOOL SetComputerName(  LPCTSTR lpComputerName //Адрес строки с именем  ); |
| 26 | Установить время между нажатиями мыши при двойном клике  UINT GetDoubleClickTime(); |
| 27 | Создать новую очеpедь сообщений пpикладной задачи указанного pазмеpа. Стаpая очеpедь удаляется.  BOOL SetMessageQueue(  UINT Msg //Размер очереди  ); |
| 28 | Установить код последней ошибки для вызванного потока  void SetLastError(  DWORD dwErrCode //Код ошибки  ); |
| 29 | Получить идентификатор текущего языка  LANGID GetUserDefaultUILanguage(); |
| 30 | Получить текущий тип даты в строковом представлении заданного формата. Возвращает количество символов в выходной строке.  UINT GetDateFormatA  (  LCID lcid, //Идентификатор языка  DWORD dwFlags, //Флаги формата даты  const SYSTEMTIME\* lpTime, //Указатель на информацию о дате в формате  LPCSTR lpFormat, //Формат строки  LPSTR lpDateStr, //Указатель на выходную строку  INT cchOut //Размер выходной строки  ) |
| 31 | Установить атрибуты для файла или каталога  BOOLSetFileAttributes(  LPCTSTR lpFileName, //Адрес строки с путем к файлу  DWORD dwFileAttributes //Байты атрибутов  ); |
| 32 | Получить атрибуты файла или каталога  DWORD GetFileAttributes(  LPCTSTR lpFileName //Адрес строки с путем к файлу  ); |
| 33 | Получить размер свободного пространства на диске  BOOL GetDiskFreeSpace(  LPCTSTR lpRootPathName, //Адрес строки с путем к корневому каталогу диска  LPDWORD lpSectorsPerCluster, //Адрес числа, содержащего количество секторов в кластере  LPDWORD lpBytesPerSector,//Адрес числа, содержащего количество байт в кластере  LPDWORD lpNumberOfFreeClusters, //Адрес числа, содержащего количество свободных кластеров  LPDWORD lpTotalNumberOfClusters//Адрес числа, содержащего общее количество кластеров  ); |
| 34 | Воспроизвести звуковой сигнал  BOOL Beep(  DWORD dwFreq, //Частота звучания, Гц  DWORD dwDuration //Длительность звучания  ); |
| 35 | Определяет позицию курсора  BOOL GetCursorPos(  LPPOINT lpPoint //адрес структуры (X; Y), куда запишется координата курсора  ); |
|  |  |