1. Написать программу на языке ассемблера, выполняющие следующие действия:
   а) получение информации согласно таблице 1;
   б) вывод полученной в пункте а информации с помощью функции MessageBoxA;
   в) выход в ОС путем вызова функции ExitProcess.
2. Получить объектный модуль в формате COFF.
3. С помощью лабораторной установки получить исполняемый PE-файл.
4. Загрузить полученный модуль с помощью системного отладчика SoftICE.

Таблица 1

	Функция для получения системной информации	Адрес загрузки
1.	Получить имя компьютера с помощью функции BOOL GetComputerNameA(LPTSTR lpBuffer, // Адрес буфера LPDWORD nSize // Адрес размера буфера);	00400000h
2.	Получить имя компьютера с помощью функции BOOL GetComputerNameA(LPTSTR lpBuffer, // Адрес буфера LPDWORD nSize // Адрес размера буфера);	00500000h
3.	Получить имя текущего каталога DWORD GetCurrentDirectoryA(DWORD nBufferLength, // Размер буфера в символах LPTSTR lpBuffer // Адрес буфера для тек. каталога);	00600000h
4.	Получить имя текущего каталога DWORD GetCurrentDirectoryA(DWORD nBufferLength, // Размер буфера в символах LPTSTR lpBuffer // Адрес буфера для тек. каталога);	00700000h
5.	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);	00800000h
6.	Получить имя активной раскладки клавиатуры BOOL GetKeyboardLayoutNameA(LPTSTR pwszKLID // Адрес буфера для раскладки); Примечание: Размер буфера не меньше 9 символов	00900000h
7.	Получить путь к текущему временному каталогу DWORD GetTempPathA(DWORD nBufferLength, // Размер буфера в символах LPTSTR lpBuffer // Адрес буфера для временного каталога);	00A00000h
8.	Получить каталог Windows UINT GetWindowsDirectoryA(LPTSTR lpBuffer, // Адрес буфера для имени каталога UINT uSize // Размер буфера); Примечание: Размер буфера не меньше 260 символов	00B00000h
9.	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	00C00000h
10.	Получитьимя текущего пользователься BOOL GetUserNameA(LPTSTR lpBuffer, // Адрес буфера для имени пользователя LPDWORD nSize // Адрес размера буфера);	00D00000h

11.	Получение числа тактов процессора после запуска системы DWORD GetTickCount(VOID); Значение возвращается в регистровой паре edx:eax	00400000h
12.	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.	
13.	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	
14.	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	
15.	Получить каталог Windows в многопользовательском режиме UINT GetSystemWindowsDirectoryA(LPTSTR lpBuffer, // Адрес буфера для имени каталога UINT uSize // Размер буфера);	00400000h
16.	Форматирует строку сообщения. DWORD FormatMessage (DWORD nSize, // size of buffer LPTSTR lpBuffer // address of buffer for drive strings);	00400000h
17.	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);	00400000h
18.	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);	00400000h

19.	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	00400000h
20.	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.	00400000h
21.	GetSystemMetrics DWORD GetSystemMetrics(DWORD Index // индекс запрашиваемой метрики); Функция возвращает заданную метрику компьютера	00400000h
22.	GetTimeFormatA GetTimeFormatA (LCID lcid, //идентификатор местности DWORD dwFlags, //флаги для изменения формата const SYSTEMTIME* lpTime, //время для форматирования LPCSTR lpFormat, //формат времени – 0 – по умолчанию LPSTR lpTimeStr, //результат - системное время в виде строки INT cchOut //размер строки lpTimeStr,0 – высчитать автоматически.); Функция возвращает заданное время в виде строки. Если время не указано, то возвращает текущее системное время.	00400000h
23.	int GetDateFormatA(_In_ LCID Locale, //идентификатор локали _In_ DWORD dwFlags, //флаги, обозначающие формат даты _In_opt_ const SYSTEMTIME *lpDate, //структура с датой, либо NULL для текущей _In_opt_ LPCSTR lpFormat, //строка формата _Out_opt_ LPSTR lpDateStr,//выходная строка _In_ int cchDate //размер выходной строки); Форматирует дату с использованием строки формата. Используется либо системная дата, либо передаётся в качестве параметра	00400000h
24.	Переименование файла с помощью функции BOOL WINAPI MoveFile (LPCTSTR lpExistingFileName, //старое имя файла LPCTSTR lpNewFileName //новое имя файла);	00400000h
25.	Копирование файла с выдачей информации о результате этого копирования BOOL WINAPI CopyFileA(_In_ LPCTSTR lpExistingFileName, //Имя существующего файла _In_ LPCTSTR lpNewFileName, //Имя нового файла	00400000h

	_In_ BOOL bFailIfExists //Если этот параметр TRUE и новый файл, указанный как lpNewFileName уже существует, то функция завершится с ошибкой. Если этот параметр FALSE и новый файл уже существует, то функция перезапишет его и завершится успешно.); Если функция завершится с ошибкой, то она вернет нулевое значение, иначе ненулевое.	
26.	Вывод сведений об операционной системе BOOL WINAPI GetVersionExA(_Inout_ LPOSVERSIONINFO lpVersionInfo //структура, возвращающая сведения об операционной системе); Если функция завершится успешно, она вернет ненулевое значение.	00400000h
27.	Получение атрибутов файлов DWORD WINAPI GetFileAttributesA (LPCTSTR filepath Путь к файлу);	00400000h
28.	Смена генерируемых событий между кнопками мыши BOOL SwapMouseButton(BOOL fSwap //Если сменить, то передавать true);	
29.	Установить время между нажатиями мыши при двойном клике BOOL SetDoubleClickTime(UINT uInterval //Интервал времени);	
30.	Получить количество процессов в системе WORD GetMaximumProcessorGroupCount();	
31.	Установить имя компьютера BOOL SetComputerName(LPCTSTR lpComputerName //Адрес строки с именем);	
32.	Установить время между нажатиями мыши при двойном клике UINT GetDoubleClickTime();	
33.	Создать новую очередь сообщений прикладной задачи указанного размера. Старая очередь удаляется. BOOL SetMessageQueue(UINT Msg //Размер очереди);	
34.	Установить код последней ошибки для вызванного потока void SetLastError(DWORD dwErrCode //Код ошибки);	
35.	Получить идентификатор текущего языка LANGID GetUserDefaultUILanguage();	
36.	Установить атрибуты для файла или каталога BOOLSetFileAttributes(LPCTSTR lpFileName, //Адрес строки с путем к файлу DWORD dwFileAttributes //Байты атрибутов);	
37.	Получить атрибуты файла или каталога DWORD GetFileAttributes(LPCTSTR lpFileName //Адрес строки с путем к файлу);	
38.	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);	
39.	Воспроизвести звуковой сигнал BOOL Beep(DWORD dwFreq, //Частота звучания, Гц DWORD dwDuration //Длительность звучания);	

Примечания:

Все функций API используют модель вызова STDCALL:

а) параметры передаются в стек в обратном порядке;

б) все параметры имеют размер 4 байта (DD, DWORD);

в) стек очищает вызываемая функция;

г) результат возвращается в регистре EAX.

Функция MessageBoxA

```
int MessageBoxA(
    HWND hWnd,          // Ссылка на родительское окно (в данном случае 0)
    LPCTSTR lpText,     // Адрес строки с текстом сообщения
    LPCTSTR lpCaption,  // Адрес строки с текстом заголовка окна
    UINT uType          // Стиль окна (рекомендуется 40h)
   );
```

Функция ExitProcess

```
VOID ExitProcess(
    UINT uExitCode      // Код завершения (0 в случае успеха)
   );
```

MessageBoxA и GetKeyboardLayoutNameA импортируется из user32.dll, GetUserNameA – из advapi32.dll, все остальные функции – из kernel32.dll

Имена функций чувствительны к регистру!
Все функции объявлять с помощью директивы “EXTRN <имя функции>: DWORD“
Пример:

```
extrn MessageBoxA:DWORD ; Функция MessageBox для формата ANSI
```

Готовую программу компилировать с помощью следующей команды:

```
ml /coff /c <имя исходного файла>
```