

**Отчет по лабораторной работе №3
«Динамически подключаемые библиотеки»**

Выполнил: студент группы Р3217

Плюхин Дмитрий

Преподаватель: Зыков А. Г.

2017 год

1. Задание

Создайте библиотеку DLL, содержащую три функции:

1. Первая лабораторная работа.
2. Вторая лабораторная работа.
3. Седьмая л. р. Из 3-го семестра. (Подумайте, какие проблемы при этом возникнут и как их обойти).

2. Листинг основной части программы

Файл library.asm (код не приведен) содержит функции, реализующие возможности, предоставляемые программами, написанными в ходе выполнения первой, второй и седьмой лабораторных работ. Файл library.def содержит список экспортируемых будущей dll функций:

```
LIBRARY    library
EXPORTS    TestFunction
EXPORTS    FirstLab
EXPORTS    SecLab
EXPORTS    SevLab
```

Файл uselibrary.cpp предоставляет возможность продемонстрировать подключение dll, использование её функций и последующую выгрузку из памяти:

```
int main() {
    HINSTANCE hDll = LoadLibrary("library.dll");
    VOID (*testfunctionprototype)(LPCTSTR);
    VOID (*sort)(LPDWORD);
    VOID (*strmask)();
    VOID (*fpu)(LPCTSTR, LPCTSTR, LPCTSTR, LPCTSTR, LPVOID);
    if (hDll == NULL) {
        cout << "Library is not loaded" << endl;
        return 1;
    }
    cout << "Library loaded" << endl;
    FARPROC testfunction = GetProcAddress(hDll, "TestFunction");
    if (testfunction == NULL) {
        cout << "Proc wasn't found" << endl;
        return 1;
    }
    cout << "Proc found" << endl;
    testfunctionprototype = (VOID(*) (LPCTSTR)) testfunction;
    testfunctionprototype("Test");

    FARPROC firstlab = GetProcAddress(hDll, "FirstLab");
    if (firstlab == NULL) {
        cout << "Proc wasn't found" << endl;
        return 1;
    }
    cout << "Proc found" << endl;
    sort = (VOID(*) (LPDWORD)) firstlab;
    DWORD aac = 10;
    sort(&aac);
    cout << "Max element in original array from lab 1 is equal to " << *((LPDWORD) aac) << endl;

    FARPROC seclab = GetProcAddress(hDll, "SecLab");
    if (seclab == NULL) {
        cout << "Proc wasn't found" << endl;
        return 1;
    }
    cout << "Proc found" << endl;
    strmask = (VOID(*) ()) seclab;
    strmask();

    FARPROC sevlab = GetProcAddress(hDll, "SevLab");
    if (sevlab == NULL) {
        cout << "Proc wasn't found" << endl;
        return 1;
    }
    cout << "Proc found" << endl;
    fpu = (VOID(*) (LPCTSTR, LPCTSTR, LPCTSTR, LPCTSTR, LPVOID)) sevlab;
    fpu("1.1", "2.5", "0.2", "0.0001");
}
```

}

3. Результаты работы программы

В результате работы программы получены следующие данные:

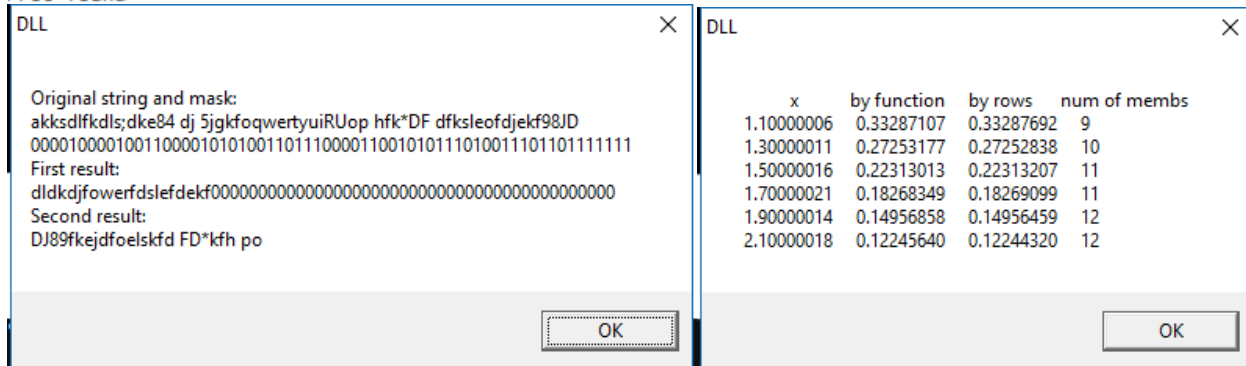
Library loaded

Proc found

Proc found

Max element in original array from lab 1 is equal to 56

Proc found



4. Вывод

Таким образом, DLL являются, с одной стороны, удобным и эффективным средством для организации повторяющихся вычислений в различных программах, их преимущество по сравнению со статическими библиотеками заключается в том, что DLL не требуется многократно загружать в память – достаточно всего лишь одной копии. С другой стороны, по прошествии некоторого времени становится все выше и выше вероятность возникновения такой ситуации, что разные приложения будут использовать разные версии одной и той же DLL и потому окажутся несовместимы друг с другом, более того, в памяти будет содержаться несколько копий одной и той же библиотеки, содержащие незначительные изменения, то есть, преимущество DLL будет сведено на нет. Единственные способы этого избежать заключаются в тщательном проектировании и централизованном распространении и управлении DLL.