Тепер бібліотека є завантаженою в пам’ять і доступна для використання. Далі потрібно отримати адреси функцій, які будуть викликатись з неї. Для цього використовують функцію GetProcAddress, яка отримує на вхід дескриптор бібліотеки та ім’я функції яку необхідно завантажити. Функція GetProcAddress повертає значення типу FARPROC, це і є вказівник на потрібну функцію. Отримане значення за допомогою приведення типів необхідно привести до правильної сигнатури. Найпростіший варіант це визначити новий тип вказівника на функцію з необхідною кількістю та типами параметрів та типом значення, що вона повертає, визначити змінну цього типи та виконати приведення до нього. Тепер з отриманою змінною вказівника можна поводитись як зі звичайною функцією — викликати з необхідними їй списком параметрів та отримувати результат її роботи.

Якщо функцію з заданим ім’ям не знайдено в бібліотеці то GetProcAddress поверне значення NULL. Це свідчить про те, що сталась помилка і подальше використання бібліотеки може привести до невизначеної поведінки всієї програми, оскільки з великою імовірністю завантажено неправильну бібліотеку. Тому необхідно контролювати отримане значення вказівника і вразі помилки переривати роботу програми.

Системна функція FreeLibrary вивантажує бібліотеку коли вона вже непотрібна. В якості параметру вона отримує дескриптор бібліотеки, що необхідно вивантажити, та повертає логічне значення, що показує чи успішно була вивантажена бібліотека.

Технічно функція FreeLibrary не вивантажує бібліотеку, а лише зменшує кількість вказівників на неї. Власне вивантаження з пам’яті буде виконано лише тоді коли лічильник вказівників досягне нуля. Це так званий механізм підрахунку посилань. Він дозволяє економити ресурси системи коли одну бібліотеку використовують декілька додатків. Перший виклик LoadLibrary завантажує бібліотеку в пам’ять, а всі інші лише збільшують лічильник посилань.

Тому дуже важливо викликати FreeLibrary кожен раз при завершенні програми, навіть при аварійному, лічильник посилань ніколи не дійде до нуля і бібліотека не буде вивантажена, аж до перезавантаження комп’ютера. В цей час її файл неможливо пермітити, перейменувати або перекомпілювати проект, оскілки файл вважається зайнятим.

З DLL бібліотеки можна викликати лише окремі функції. А при розробці програмного коду бібліотеки в об’єктно-орієнтованому стилі весь функціонал розділений між об’єктами, які досить часто повинні бути в пам’яті на протязі всієї роботи програми. Виникає ситуація при якій об’єкт повинен бути створений перед викликом функції та видалятись після кількох викликів різних функцій. Навіть якщо ствоювати об’єкт «ліниво», тобто при кожному виклику кожної функції перевіряти чи сторений об’єкт, якщо ні то створити його, залишається невирішеною проблема видалення.

Найбільш простий варіант це в бібліотеці створити функції, що необхідно вилки до початку роботи та після її завершення. Але може ускладнити використання бібліотеки, оскільки розробники часто забувають викликати такі функції, особливо ті що пов’язані з очисткою ресурсів, адже їх влив не роботу програми не завжди очевидний.

Для вирішення цієї проблеми можна додати спеціальну функцію DllMain, яка буде автоматично викликана при виникненні однієї з описаних нижче подій. Щоб визначити, що це за подія необхідно порівняти значення параметру fdwReason з однією з наступних констант:

DLL\_PROCESS\_ATTACH — бібліотека була завантажена в віртуальний адресний простір поточного процесу, як результат запуску процесу або виклику LoadLibrary.

DLL\_PROCESS\_DETACH - бібліотека вивантажена з віртуального адресного простору процесу, що викликав через невдалу спробу завантаження або тому, що лічильник посилань досяг нуля.

DLL\_THREAD\_ATTACH — Поточний процес створює новий потік. Коли це відбувається, система викликає функцію точки входу всіх бібліотек DLL, що в даний час приєднані до цього процесу. Виклик виконується в контексті нового потоку.

DLL\_THREAD\_DETACH — Потік завершився повністю. Якщо DLL зберігає вказівник на виділену пам'ять, потрібно використовувати цю можливість, щоб звільнити її. Система викликає функцію точки входу всіх завантажених в даний момент бібліотек DLL з цим значенням.[XXXXXX]

Найбільш цікавими є перші дві події, тому, що саме при їх виникненні можна виконати ініціалізацію і очитку ресурсів в середині бібліотеки.

При додавані до порожнього проекту функції DllMain можна отримати помилку, що така функція уже визначена. Щоб позбавитись від неї необхідно просто іще раз перекомпілювати проект.

<https://msdn.microsoft.com/ru-ru/library/windows/desktop/ms682583(v=vs.85).aspx>