

# ***Лекція “Динамічні бібліотеки”***

Автор Голуб Б.Л.

2017

# DLL

- Бібліотеки, що мають динамічне підключення (*Dynamics Link Libraries, DLL*) - наріжний камінь операційної системи Windows, починаючи з найпершої її версії. В DLL утримуються всі функції Windows API.
- Три найважливіші DLL: *Kernel32.dll* (керування пам'яттю, процесами й потоками), *User32.dll* (підтримка користувальницького інтерфейсу, у тому числі функції, пов'язані зі створенням вікон і передачею повідомлень) і *GDI32.dll* (графіка й виведення тексту).

# *Переваги DLL*

- **Розширення функціональності додатка**

DLL можна динамічно завантажувати в адресний простір процесу, що дозволяє додатку, визначивши, які дії від нього потрібні, довантажувати потрібний код. Тому одна компанія, створивши якийсь додаток, може передбачити розширення його функціональності за рахунок DLL від інших компаній.

# *Переваги DLL*

- **Можливість використання різних мов програмування.**

Немає значення, на якій мові писати ту або іншу частину додатка. Так, користувальницький інтерфейс додатка можна створювати на Microsoft Visual C#, але прикладну логіку найкраще реалізувати на C++. Програма на C# може завантажувати DLL, написані на C++, JAVA, BASIC.

# *Переваги DLL*

- **Більш просте керування проектом.**

Якщо в процесі розробки програмного продукту окремі його модулі створюються різними групами, то при використанні DLL таким проектом управляти набагато простіше. Однак кінцева версія додатка повинна включати якнайменше файлів.

# *Переваги DLL*

- **Економія пам'яті**

Якщо ту саму DLL використає кілька додатків, в оперативній пам'яті може зберігатися тільки один її екземпляр, доступний цим додаткам. Приклад — DLL-версія бібліотеки C/C++. Нею користуються багато додатків.

# *Переваги DLL*

- **Розділення ресурсів**

DLL можуть містити такі ресурси, як шаблони діалогових вікон, рядка, значки й бітові карти (растрові зображення). Ці ресурси доступні будь-яким програмам.

# *Переваги DLL*

- **Спрощення локалізації**

DLL нерідко застосовуються для локалізації додатків. Наприклад, додаток, що містить тільки код без усяких компонентів користувацького інтерфейсу, може завантажувати DLL з компонентами локалізованого інтерфейсу.



# *Переваги DLL*

- **Рішення проблем, пов'язаних з особливостями різних платформ**

У різних версіях Windows утримуються різні набори функцій. Найчастіше розроблювачам потрібні нові функції, що існують у тій версії системи, якою вони користуються. Рішення такої проблеми – завантажити їх у вигляді DLL

# Переваги DLL

- Реалізація специфічних можливостей

Певна функціональність в Windows доступна тільки при використанні DLL. Наприклад, окремі види пасток, установлюваних викликом *SetWindowsHookEx* і *SetWinEventHook* можна задіяти при умові, що функція повідомлення пастки розміщена в DLL. Крім того, розширення функціональності оболонки Windows можливо лише за рахунок створення Com-об'єктів, існування яких припустимо тільки в DLL. Це ж має відношення й до додатків, що завантажуються Web-браузером, ActiveX-елементам, що дозволяють створювати Web-сторінки з більш багатую функціональністю.

# Підготовка Exe- і Dll- модулів

- Підготовка заголовного файлу із прототипами функцій, структурами й ідентифікаторами, експортованими з DLL. Цей файл включається у вихідний код всіх модулів DLL.
- На C/C++ створюється модуль (або модулі) вихідного коду з тілами функцій і визначеннями змінних, які повинні перебувати в DLL.
- Компілятор перетворить вихідний код модулів DLL в OBJ-файли (по одному на кожний модуль).
- Компонувальник збирає всі OBJ-модулі в єдиний завантажувальний DLL-модуль, у який в остаточному підсумку містяться двійковий код і змінні (глобальні й статичні), що належать до даного DLL. Цей файл буде потрібно при компіляції модуля, що виконується.
- Якщо компонентувальник виявить, що DLL експортує хоча б одну змінну або функцію, то створить і LIB-файл. Цей файл зовсім малюсінький, оскільки в ньому немає нічого, крім списку символічних імен функцій і змінних, експортованих з DLL. Цей LIB-файл теж знадобиться при компіляції EXE-файла.  
Створивши DLL, можна перейти до зборки модуля, що виконується.
- В усі модулі вихідного коду, де є посилання на зовнішні функції, змінні, структури даних або ідентифікатори, треба включити заголовний файл, наданий розроблювачем DLL.

# Неявне зв'язування EXE- і DLL-модулів

## СТВОРЕННЯ DLL

- 1) Заголовний файл із експортованими прототипами, структурами й ідентифікаторами (символьними іменами)
- 2) Вихідні файли C/C++, у яких реалізовані функції й визначені змінні
- 3) Компілятор створює OBJ-файл із кожного вихідного файлу C/C++
- 4) Компонувальник збирає DLL з OBJ-модулів
- 5) Якщо DLL експортує хоча б одну змінну або функцію, компонентувальник створює й LIB-файл

## СТВОРЕННЯ EXE

- 1) Заголовний файл із імпортованими прототипами структурами й ідентифікаторами
- 2) Вихідні файли C/C++, з яких викликаються імпортовані функції й змінні
- 3) Компілятор створює OBJ-файл із кожного вихідного файлу C/C++
- 4) Використовуючи OBJ модулі й LIB-файл і з огляду на посилання на імпортовані ідентифікатори компонентувальник збирає EXE-модуль (у якому також розміщується таблиця імпорту — список необхідних DLL і імпортованих ідентифікаторів)

# Схема зв'язування

