

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

ІКНІ
Кафедра ПЗ



ЗВІТ

До лабораторної роботи №7

на тему: “Статичні та динамічні бібліотеки. WINDOWS та LINUX”

з дисципліни: “Операційні системи”

Лектор:
старший викладач кафедри ПЗ
Грицай О.Д.

Виконав:
студент групи ПЗ-24
Губик А. С.

Прийняв:
доцент кафедри ПЗ
Горечко О. М.

Тема роботи:Статичні та динамічні бібліотеки. WINDOWS та LINUX

Мета роботи:Ознайомитися з статичними та динамічними бібліотеками в операційних системах WINDOWS та LINUX. Навчитися реалізовувати статичні та динамічні бібліотеки.

Теоретичні відомості

Створення програмного забезпечення довгий і дорогий процес, тому існує необхідність його спростити та оптимізувати для подальшого модифікування застосунків. Часто виникає необхідність використання одних і тих ж функцій або даних у різних частинах проекту. Частина функцій у проекті використовується рідко, а частина при створенні нової версії застосунку буде постійно змінюватись або додаватимуться нові функції. Крім того, всі дані можна розділити за певною тематикою або за видом ресурсів і т.д. Також потрібно передбачити оновлення даних. Тому виникає необхідність винести частину коду і даних з основної частин проекту в окремі приєднанні модулі. Зрозуміло, що виносити кожен функцію або екземпляр даних в окремий файл чи модуль недоцільно, оскільки це спричинить довший відгук від програмного продукту, навантажить систему додатковими викликами, що у свою чергу зменшить продуктивність і ефективність застосунку. Зазвичай функції або дані об'єднують згідно функціональності, за тематикою та іншими властивостями в окремі скомпільовані спеціальні об'єктні файли – бібліотеки. Бібліотека – це файл, що містить один або декілька об'єктних файлів для вирішення близьких за тематикою завдань у розробці програмних продуктів. У бібліотеці можуть міститися об'єктні модулі, програмний код або дані, що можуть використовуватись окремо або разом на різних етапах створення проекту (компілювання, лінкування, завантаження чи виконання). Бібліотека містить символічний індекс, який складається з назв функцій і змінних, які містяться у бібліотеці. Це дозволяє прискорити процес лінкування програми, оскільки пошук функцій і змінних в об'єктних файлах бібліотеки відбувається набагато швидше, ніж пошук в наборі вказаних об'єктних файлів. Тому використання бібліотеки дозволяє компактно зберігати усі необхідні об'єктні файли в одному місці, і при цьому значно підвищити швидкість компіляції. Бібліотеки прийнято розділяти відповідно до способу з'єднання на статичні та динамічні. Статичні бібліотеки зв'язуються з проектом перед завантаженням і є частиною бінарного файлу. Динамічні можуть бути зв'язані з проектом статично і динамічно. У різних операційних системах є свої особливості створення і використання бібліотек

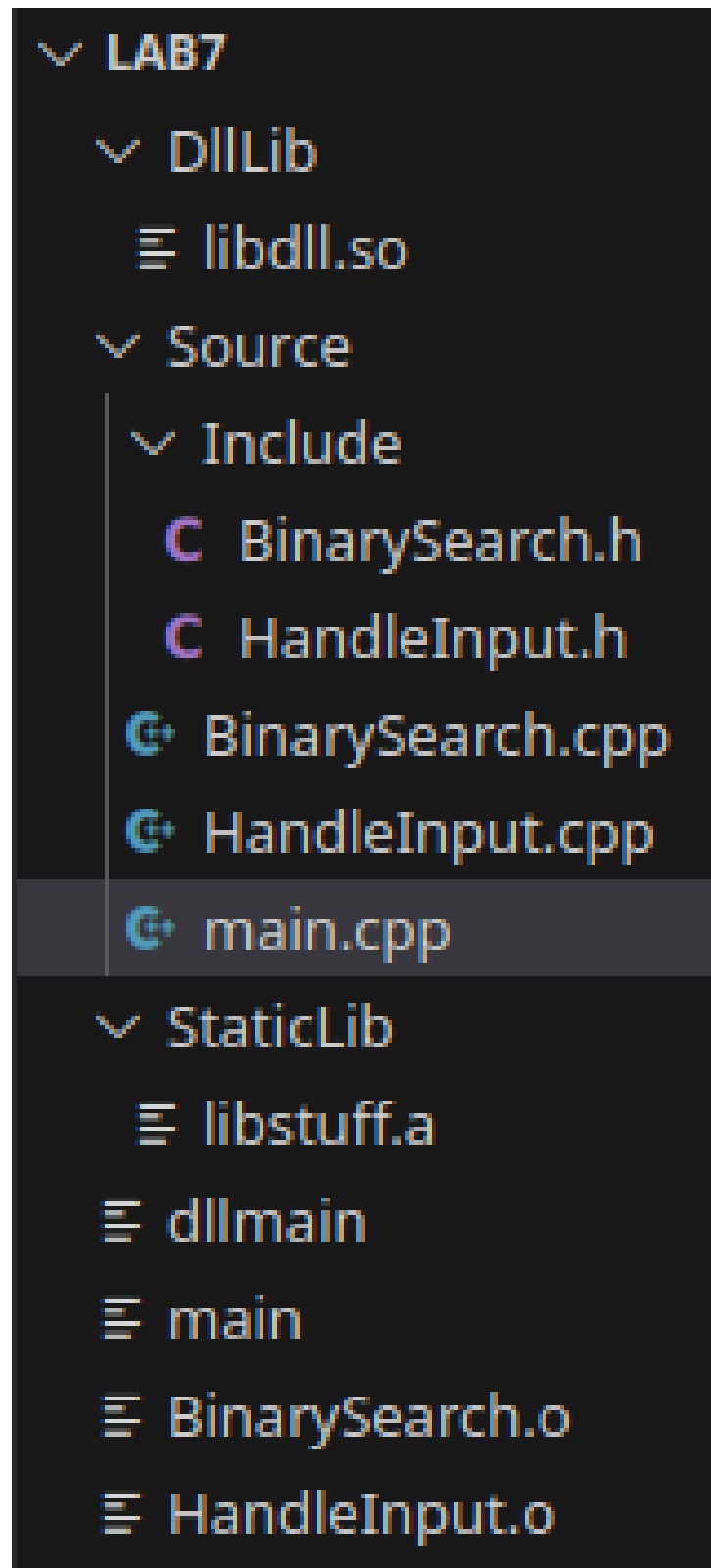


Рис. 1: Вміст директорій

```

1934 g++ main.cpp BinarySearch.cpp HandleInput.cpp -DPROP_SYNC -o main
1935 ./main
1936 ar rcs stuff.a BinarySearch.cpp HandleInput.cpp
1937 rm stuff.a
1938 ar rcs stuff.a BinarySearch.cpp HandleInput.cpp
1939 gcc main.cpp stuff.a -o main
1940 gcc main.cpp -Lstuff.a -o main
1941 ar rcs BinarySearch.cpp HandleInput.cpp
1942 ar rcs stuff.lib BinarySearch.cpp HandleInput.cpp
1943 rm stuff.lib
1944 ar rcs stuff.a BinarySearch.cpp HandleInput.cpp
1945 gcc -c BinarySearch.cpp -o BinarySearch.o
1946 gcc -c HandleInput.cpp -o HandleInput.o
1947 ar rcs stuff.a BinarySearch.o HandleInput.o
1948 gcc main.cpp stuff.a -o main
1949 ranlib *.a
1950 gcc main.cpp stuff.a -o main
1951 gcc main.cpp -L. -lstuff -o main
1952 gcc main.cpp -L. -lstuff.a -o main
1953 ar rcs libstuff.a BinarySearch.o HandleInput.o
1954 gcc main.cpp -L. -lstuff.a -o main
1955 gcc main.cpp -L. -lstuff -o main
1956 ran libstuff.a
1957 ranlib libstuff.a
1958 gcc main.cpp -L. -lstuff -o main
1959 gcc -c main.cpp -o main.o
1960 gcc main.o -L. -lstuff -o main
1961 g++ -c BinarySearch.cpp -o BinarySearch.o
1962 g++ -c HandleInput.cpp -o HandleInput.o
1963 g++ main.cpp -L. -lstuff -o main
1964 ./main
1965 g++ main.cpp libstuff.a -o main
1966 g++ main.cpp -I./Source/Include -L./StaticLib -lstuff -o main
1967 g++ main.cpp -I./Source/Include -LStaticLib -lstuff -o main
1968 g++ main.cpp -ISource/Include -LStaticLib -lstuff -o main
1969 g++ ./Source/main.cpp -ISource/Include -LStaticLib -lstuff -o main
1970 g++ -c Source/HandleInput.cpp -o HandleInput.o
1971 g++ -c Source/BinarySearch.cpp -o BinarySearch.o
1972 g++ -shared -fPIC HandleInput.o BinarySearch.o -o libdll.so
1973 g++ HandleInput.o BinarySearch.o -shared -fPIC -o libdll.so
1974 g++ -c -fPIC Source/BinarySearch.cpp -o BinarySearch.o
1975 g++ -c -fPIC Source/HandleInput.cpp -o HandleInput.o
1976 g++ HandleInput.o BinarySearch.o -shared -fPIC -o libdll.so
1977 g++ Source/main.cpp libdll.so -o dllmain

```

Рис. 2: Використані команди

Windows

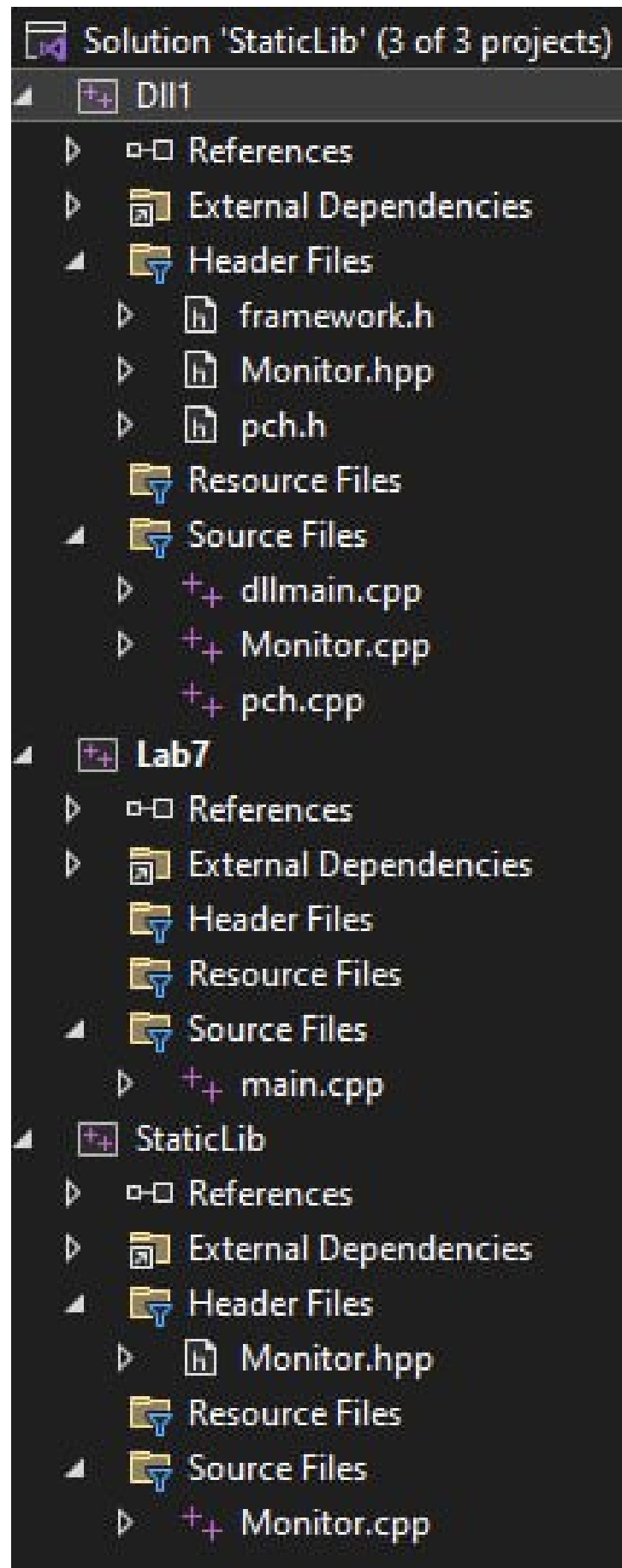
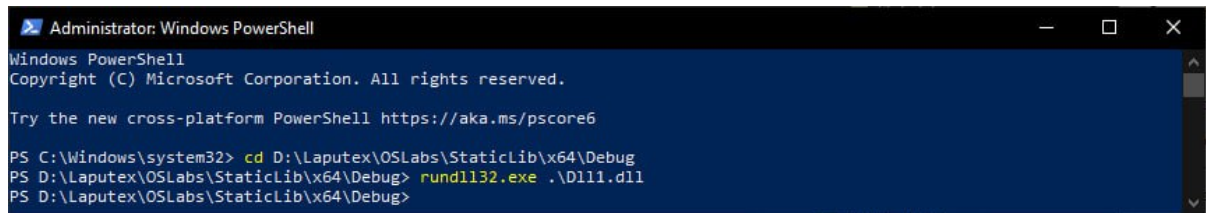


Рис. 3: Вміст директорій



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\Windows\system32> cd D:\Laputex\OSLabs\StaticLib\x64\Debug
PS D:\Laputex\OSLabs\StaticLib\x64\Debug> rundll32.exe .\D111.dll
PS D:\Laputex\OSLabs\StaticLib\x64\Debug>
```

Рис. 4:

Висновок: Я навчився створювати динамічні та статичні бібліотеки на Linux і Windows.