

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Інститут КНІТ
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-22
Коваленко Д.М.

Прийняла:

старший викладач кафедри ПЗ
Грицай О.Д.

«_____» _____ 2022 р.
 Σ = _____

Тема. Статичні та динамічні бібліотеки. WINDOWS та LINUX.

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

Лабораторне завдання

Завдання.

1. Реалізувати лабораторну роботу №5 (згідно варіанту) у вигляді статичної та динамічної бібліотеки в ОС WINDOWS.
2. Запустити створену динамічну бібліотеку з командної стрічки (cmd.exe) за допомогою rundll32.exe.
3. Створити окрему програму і реалізувати статичний зв'язок між програмою та бібліотекою із п. 1.
4. Створити окрему програму і реалізувати динамічний зв'язок між програмою та бібліотекою із п. 1.
5. Експортувати головну функцію бібліотеки під іншим іменем із п. 1.
6. Реалізувати лабораторну роботу №6 у вигляді статичної та динамічної (поділюваної) бібліотеки в ОС LINUX.
7. Створити окрему програму і реалізувати статичний зв'язок між програмою та бібліотекою із п. 2
8. Створити окрему програму і реалізувати динамічний зв'язок між програмою та бібліотекою із п. 2.
9. Порівняти результати виконання програми та роботи бібліотек під ОС Windows та Linux.
10. Результати виконання роботи відобразити у звіті.

2. Обчислити суму елементів заданого масиву (кількість елементів >10000, елементи рандомні)
(Синхронізація: семафор, спімблокування)

Хід роботи

WINDOWS

main.cpp

```
#include <iostream>
#include <Windows.h>
#include "libSum.h"
#ifdef RUNTIME
#include "dllSum.h"
#else
typedef int (__cdecl* DLLSUMUP)(int*, int);
#endif

int main() {
    int len;
```

```

std::cout << "Enter array len: ";
std::cin >> len;
int* arr = new int[len];
std::srand(static_cast<unsigned int>(std::time(nullptr)));
for (int i = 0; i < len; i++) arr[i] = rand() % 10;

int sum = libSumUp(arr, len);
std::cout << "Sum from static lib: " << sum << std::endl;

#ifdef RUNTIME
int dllSum = dllSumUp(arr, len);
std::cout << "Sum from dynamic lib (using load-time linking): " << dllSum <<
std::endl;
#else
HINSTANCE hinstLib;
DLSUMUP dllSumUp;
BOOL fFreeResult, fRunTimeLinkSuccess = FALSE;

hinstLib = LoadLibrary(TEXT("dllSum.dll"));

if (hinstLib != NULL) {
    dllSumUp = (DLSUMUP)GetProcAddress(hinstLib, "dllSumUp");

    if (NULL != dllSumUp) {
        fRunTimeLinkSuccess = TRUE;
        int dllSum = (dllSumUp)(arr, len);
        std::cout << "Sum from dynamic lib (using run-time linking): " <<
dllSum << std::endl;
    }

    fFreeResult = FreeLibrary(hinstLib);
}

if (!fRunTimeLinkSuccess)
std::cout << "Failed to link on runtime" << std::endl;
#endif
}

```

libSum.h

```

#pragma once
namespace sum {
    int sumUp(int* arr, int len);
}

```

libSum.cpp

```

#include "pch.h"
#include "framework.h"
#include "libSum.h"

namespace sum {
    int sumUp(int* arr, int len) {
        int sum = 0;
        for (int i = 0; i < len; i++) sum += arr[i];
        return sum;
    }
}

```

dllSum.h

```
#pragma once

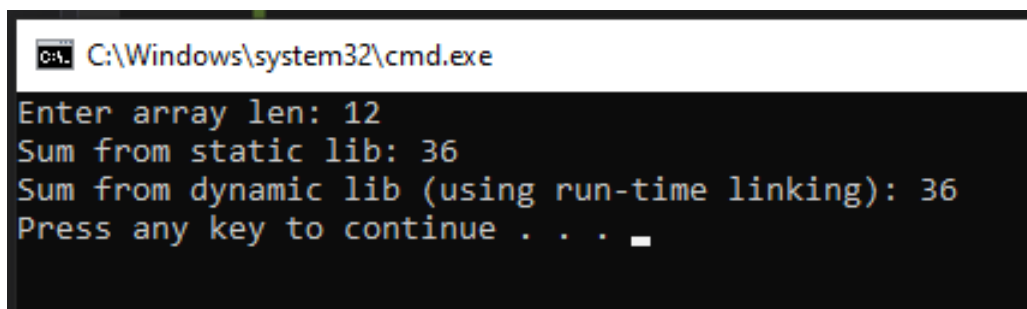
#ifndef DLLSUM_EXPORTS
#define DLLSUM_API __declspec(dllexport)
#else
#define DLLSUM_API __declspec(dllimport)
#endif

extern "C" DLLSUM_API int dllSumUp(int* arr, int len);
```

dllSum.cpp

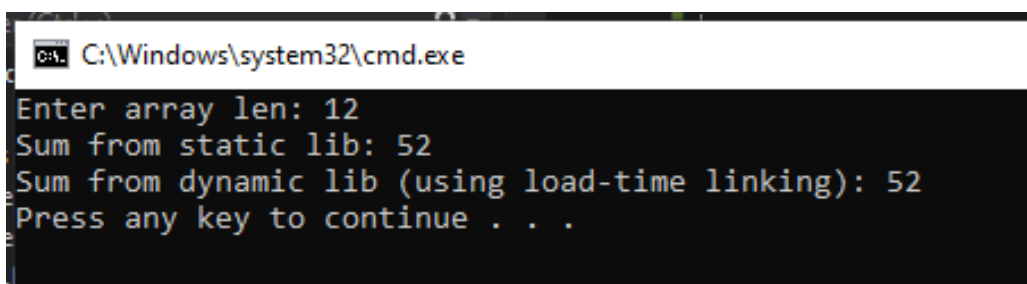
```
#include "pch.h"
#include "dllSum.h"

int dllSumUp(int* arr, int len) {
    int sum = 0;
    for (int i = 0; i < len; i++) sum += arr[i];
    return sum;
}
```

A screenshot of a Windows command prompt window. The title bar shows 'C:\Windows\system32\cmd.exe'. The prompt is 'C:\>'. The user has entered 'Enter array len: 12'. The program outputs 'Sum from static lib: 36' and 'Sum from dynamic lib (using run-time linking): 36'. It then prompts 'Press any key to continue . . .'.

```
C:\> C:\Windows\system32\cmd.exe
Enter array len: 12
Sum from static lib: 36
Sum from dynamic lib (using run-time linking): 36
Press any key to continue . . .
```

Рис. 1: Виконання програми (динамічний зв'язок з динамічною бібліотекою)

A screenshot of a Windows command prompt window. The title bar shows 'C:\Windows\system32\cmd.exe'. The prompt is 'C:\>'. The user has entered 'Enter array len: 12'. The program outputs 'Sum from static lib: 52' and 'Sum from dynamic lib (using load-time linking): 52'. It then prompts 'Press any key to continue . . .'.

```
C:\> C:\Windows\system32\cmd.exe
Enter array len: 12
Sum from static lib: 52
Sum from dynamic lib (using load-time linking): 52
Press any key to continue . . .
```

Рис. 2: Виконання програми (статичний зв'язок з динамічною бібліотекою)

LINUX

aSum.h

```
extern "C" int soSumUp(int* arr, int len);
```

aSum.cpp

```
#include "aSum.h"
```

```
int sumUp(int* arr, int len) {  
    int sum = 0;  
    for (int i = 0; i < len; i++) sum += arr[i];  
    return sum;  
}
```

soSum.h

```
int soSumUp(int* arr, int len);
```

soSum.cpp

```
#include "soSum.h"
```

```
int soSumUp(int* arr, int len) {  
    int sum = 0;  
    for (int i = 0; i < len; i++) sum += arr[i];  
    return sum;  
}
```

main.cpp

```
#include <iostream>  
#include "aSum.h"  
#define RUNTIME  
#ifndef RUNTIME  
#include "soSum.h"  
#else  
#include <dlfcn.h>  
int (*soSumUp)(int*, int);  
#endif  
  
int main() {  
    int len;  
    std::cout << "Enter array len: ";  
    std::cin >> len;  
    int* arr = new int[len];  
    srand(static_cast<unsigned int>(time(nullptr)));  
    for (int i = 0; i < len; i++) arr[i] = rand() % 10;  
  
    int aSum = sumUp(arr, len);  
    std::cout << "Sum from static lib: " << aSum << std::endl;  
  
#ifndef RUNTIME  
    int soSum = soSumUp(arr, len);  
    std::cout << "Sum from dynamic lib (using load-time linking: " << soSum <<  
    std::endl;  
#else  
    void* lib;  
    lib = dlopen("./libSum.so", RTLD_LAZY);  
    if (!lib)  
        std::cout << "Failed to link on runtime" << std::endl;  
  
    soSumUp = (int (*)(int*, int))dlsym(lib, "soSumUp");  
  
    int soSum = (*soSumUp)(arr, len);
```

```

    std::cout << "Sum from dynamic lib (using run-time linking: " << soSum <<
    std::endl;

    dlclose(lib);
#endif
}

```

run.sh

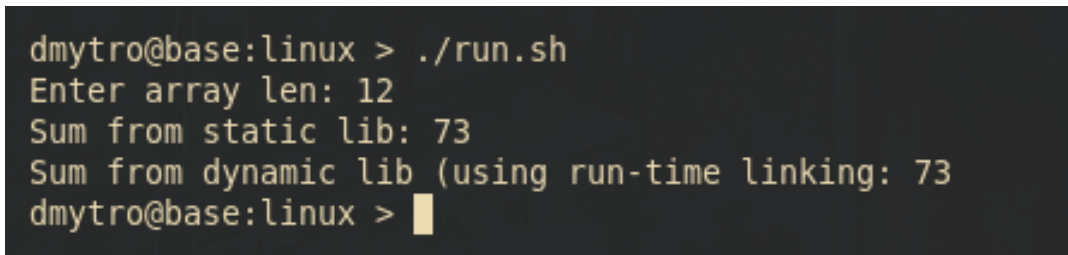
```

#!/bin/sh
export LD_LIBRARY_PATH=.

g++ -c main.cpp
g++ -c aSum.cpp
g++ -fPIC -c soSum.cpp
ar rc libSum.a aSum.o
g++ -shared -o libSum.so soSum.o
g++ main.o libSum.a libSum.so -ldl
./a.out

rm *.a *.o *.so *.out

```

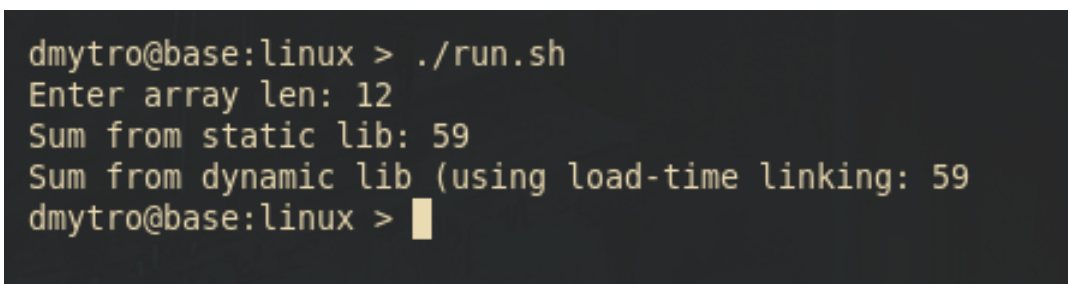


```

dmytro@base:linux > ./run.sh
Enter array len: 12
Sum from static lib: 73
Sum from dynamic lib (using run-time linking: 73
dmytro@base:linux >

```

Рис. 3: Виконання програми (динамічний зв'язок з динамічною бібліотекою)



```

dmytro@base:linux > ./run.sh
Enter array len: 12
Sum from static lib: 59
Sum from dynamic lib (using load-time linking: 59
dmytro@base:linux >

```

Рис. 4: Виконання програми (статичний зв'язок з динамічною бібліотекою)

Висновок

Під час виконання лабораторної роботи я ознайомився з статичними та динамічними бібліотеками в операційних системах WINDOWS та LINUX. Навчився реалізовувати статичні та динамічні бібліотеки