**Лабораторная работа №3**

**Библиотеки**

**Цель работы:** Получение практических навыков в работе со статическими и динамическими библиотеками.

**Функционал для библиотек:**

Любая библиотека из данной лабораторной работы должна содержать следующий функционал:

* Итеративную функцию бинарного поиска;
* Рекурсивную функцию бинарного поиска;
* Массив целых чисел размером 1024 элемента.

Реализацию функций можно взять [отсюда](https://rosettacode.org/wiki/Binary_search#C), а массив сгенерировать [тут](https://onlinetools.com/integer/create-integer-array).

**Постановка задачи для Windows:**

**Проект <Ваши Инициалы>Lib:**

Должен содержать код для создания статической библиотеки с заданным функционалом (см. выше). Код должен быть кроссплатформенным, так как будет применяться и для Linux/macOS (Для тех кто не берёт код из указанного выше источника).

**Проект Lab-03a:**

Разработать клиентское приложение которое будет использовать статическую библиотеку. Код также должен быть кроссплатформенным, так как и в этом случае, код будет применяться для Linux/macOS. Приложение должно быть консольным и может быть вызывано в виде:

***Lab-03a <Имя вызываемой функции> [Искомое число]***

Если искомое число не задано, то оно запрашивается для ввода с консоли. В результате выполнения этого приложения должен быть выведен один из следующих результатов:

* Не указана вызываемая функция!
* <Имя вызванной функции>: Заданное число не найдено!
* <Имя вызванной функции>: Число <Х> найдено на позиции <Y>!

Проверить факт подключения библиотеки через Process Explorer.

***Вопрос:*** *Является ли это возможным? Почему?*

**Проект <Ваши Инициалы>DLib:**

Разработать динамическую библиотеку на языке **C** с заданным функционалом (см. выше). Также должна быть реализована функция **DllMain** c логированием событий в некоторый файл (путь к файлу должен быть абсолютным).

Экспорт функций и переменных организовать через **.def** файл так, чтобы:

* Одна из функций экспортировалась без имени;
* Одна из функций экспортировалась с именем отличающимся от оригинального.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

**Проект <Ваши Инициалы>DLib++:**

Разработать динамическую библиотеку на языке **C++** с заданным функционалом (см. выше). Также должна быть реализована функция **DllMain** c логированием событий в некоторый файл (путь к файлу должен быть абсолютным).

Экспорт функций и переменных организовать через **\_\_declspec(dllexport)**. Одной из функций указать модификатор **extern “C”**.

Изучить таблицы экспорта обеих библиотек с помощью утилиты **dumpbin** (данная утилита доступна из Developer версии терминала Visual Studio)и предоставить скриншоты с их содержимым! Имя функции преобразованное компилятором C++ сохраните для дальнейших вызовов этой функции!

Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

**Проект Lab-03b-Explicit:**

Разработать клиентское приложение которое сможет **явно** подключить любую из двух динамических библиотек. Приложение должно быть консольным и может быть вызывано в виде:

***Lab-03b-ex <Подключаемая библиотека> <Имя или номер вызываемой функции> [Искомое число]***

**Информация!** Параметры «Подключаемая библиотека» и «Имя вызываемой функции» должны быть непосредственно значениями передаваемыми в функции **LoadLibrary** и **GetProcAddress**.

Если искомое число не задано, то оно запрашивается для ввода с консоли (для наглядности работы динамической библиотеки поместите этот код непосредственно перед вызовом функции поиска). В результате выполнения этого приложения должен быть выведен один из следующих результатов:

* Не указана загружаемая библиотека!
* Не указана вызываемая функция!
* Загружаемая библиотека не найдена!
* Искомая функция <Имя вызываемой функции> не найдена в библиотеке <Имя загруженной библиотеки>!
* <Имя вызванной функции>: Заданное число не найдено!
* <Имя вызванной функции>: Число <Х> найдено на позиции <Y>!

Изучить таблицу импорта приложения с помощью утилиты **dumpbin** и предоставить скриншот с её содержимым!

Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

***Вопрос:*** *Что оказалось в таблице импорта? Почему?*

Проверить факт подключения библиотеки через Process Explorer. (Для этого не указывайте число в аргументах при запуске приложения и дайте приложению войти в режим ожидания ввода с консоли).

****

**Изображение выглядит как текст, снимок экрана, число, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

**Проект Lab-03b-Implicit:**

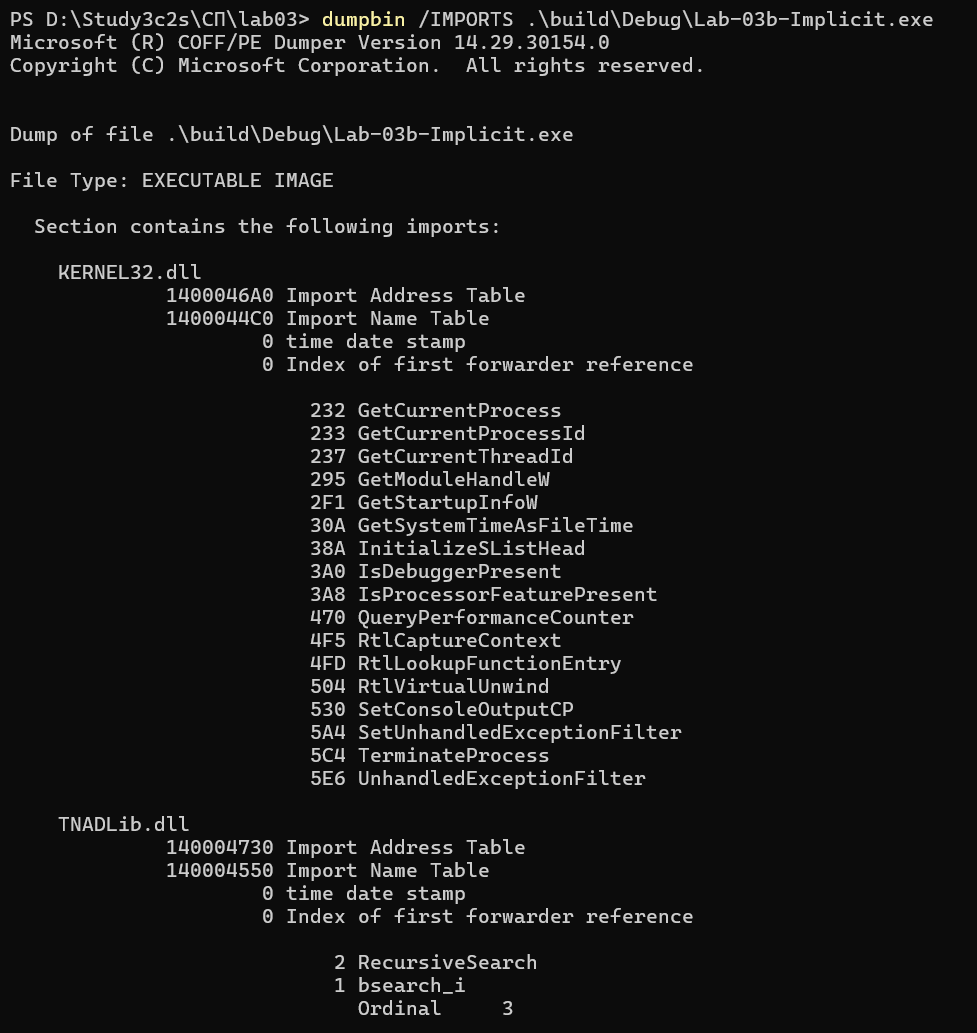
Разработать клиентское приложение которое неявно подключает одну любую из двух динамических библиотек. Приложение должно быть консольным и может быть вызывано в виде:

***Lab-03b-im <Имя вызываемой функции> [Искомое число]***

Если искомое число не задано, то оно запрашивается для ввода с консоли. В результате выполнения этого приложения должен быть выведен один из следующих результатов:

* Не указана вызываемая функция!
* <Имя вызванной функции>: Заданное число не найдено!
* <Имя вызванной функции>: Число <Х> найдено на позиции <Y>!

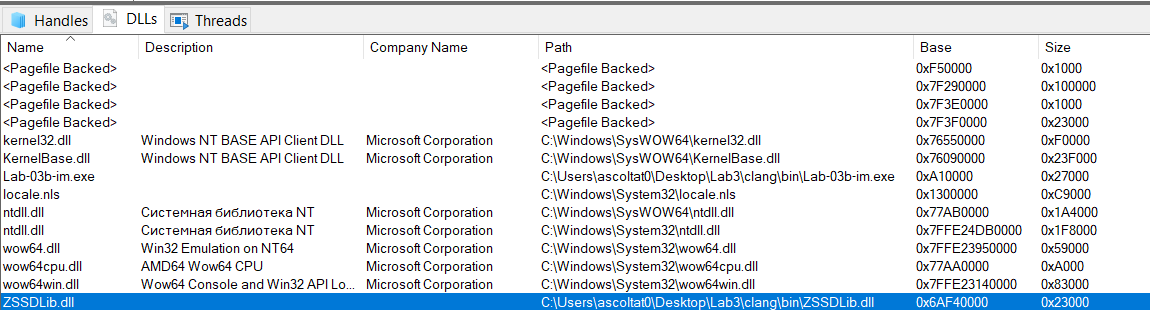
Изучить таблицу импорта приложения с помощью утилиты **dumpbin** и предоставить скриншот с её содержимым!



***Вопрос:*** *Что оказалось в таблице импорта? Почему?*

Проверить факт подключения библиотеки через Process Explorer. (Для этого не указывайте число в аргументах при запуске приложения и дайте приложению войти в режим ожидания ввода с консоли).





**Постановка задачи для Linux:**

**Проект <Ваши Инициалы>Lib:**

Собрать статическую библиотеку из проекта с аналогичным названием на Windows.

**Проект Lab-03a:**

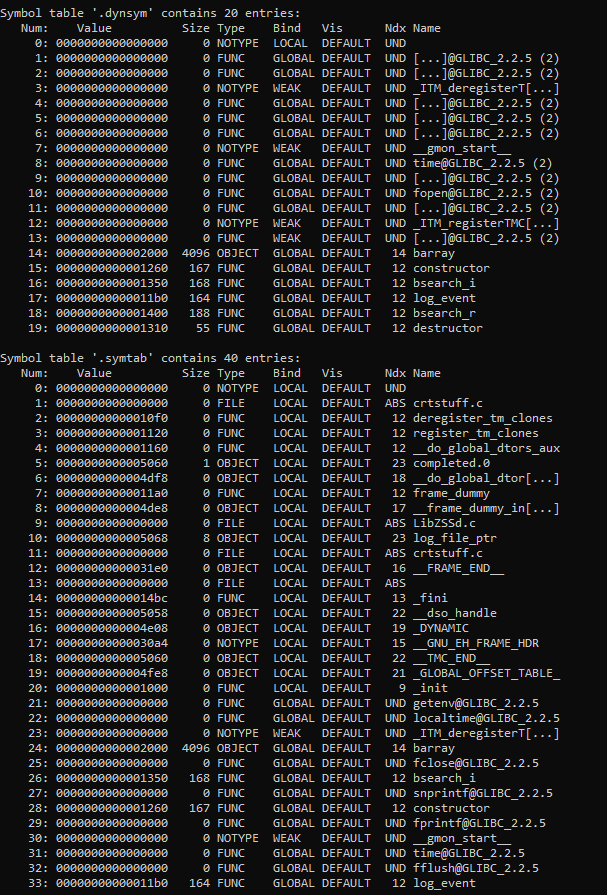
Собрать приложение из проекта с аналогичным названием на Windows. Проверить работоспособность приложения.

**Проект Lib<Ваши Инициалы>d:**

Разработать разделяемую библиотеку на языке **C** с заданным функционалом (см. выше). Также должна быть реализованы функции **constructor** и **destructor** с логированием событий в некоторый файл (путь к файлу должен быть абсолютным).

Просмотреть список экспортируемых символов через утилиту **objdump**/**readelf** и предоставить скриншот с его содержимым!

readelf -s LibZSSd.so



**Проект Lab-03c:**

Разработать клиентское приложение которое сможет **явно** подключать разделяемую библиотеку. Приложение должно быть консольным и может быть вызывано в виде:

***Lab-03с <Подключаемая библиотека> <Имя или номер вызываемой функции> [Искомое число]***

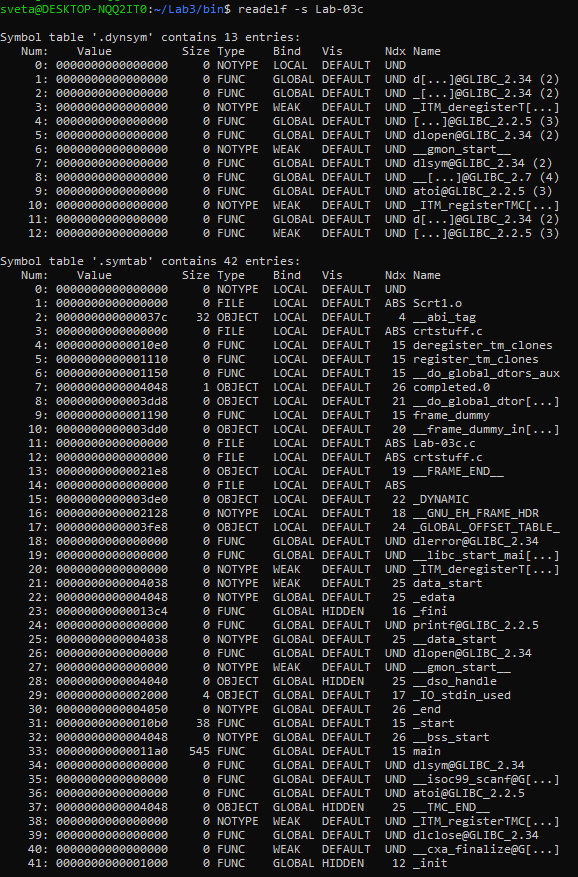
**Информация!** Параметры «Подключаемая библиотека» и «Имя вызываемой функции» должны быть непосредственно значениями передаваемыми в функции **dlopen** и **dlsym**.

Если искомое число не задано, то оно запрашивается для ввода с консоли (для наглядности работы динамической библиотеки поместите этот код непосредственно перед вызовом функции поиска). В результате выполнения этого приложения должен быть выведен один из следующих результатов:

* Не указана загружаемая библиотека!
* Не указана вызываемая функция!
* Загружаемая библиотека не найдена!
* Искомая функция <Имя вызываемой функции> не найдена в библиотеке <Имя загруженной библиотеки>!
* <Имя вызванной функции>: Заданное число не найдено!
* <Имя вызванной функции>: Число <Х> найдено на позиции <Y>!

Просмотреть список импортируемых символов через утилиту **objdump**/**readelf** и предоставить скриншот с его содержимым!

readelf -s Lab-03c



**Дополнительно (совершенно не обязательно, но огромный плюс):**

Написать примеры внедрения кода с использованием динамических библиотек как на Windows, так и на Linux (примеры из лекций сдавать как свои работы бесполезно ^\_^).

**Требования к выполненной работе:**

* Для генерации и сборки проекта использовать как средство CMake, так и clang/clang++ напрямую;
* Для обоих случаев выучить команды (консольные или в CMakeLists.txt) которые используются для сборки библиотек или приложений использующих библиотеки!
* Для обоих случаев подготовить серию скриншотов со сборкой библиотек и приложений!

При использовании CMake должно быть выполнено следующее:

* Сборка должна быть «out-of-source»;
* Для Windows использовать генератор «Visual Studio»/«Ninja» для Linux – «Unix Makefiles», для macOS – «XCode»;
* При сборке проектов всегда должен использоваться clang;
* Все проекты для обеих ОС являются частью одного большого проекта. Добавить в конфигурационные файлы условия сборки отдельных проектов для разных ОС. Конечные файлы генерируются в общий каталог!

**Вопросы для контроля:**

1. Что такое библиотека?
2. Что такое статическая библиотека?
3. Как подключить статическую библиотеку?
4. На каком этапе сборки происходит непосредственно загрузка кода?
5. Что такое динамическая библиотека?
6. Какой механизм лежит в основе работы динамических библиотек?
7. Назовите два способа подключения динамической библиотеки? Кратко поясните порядок подключений.
8. Что такое библиотека импорта?
9. Для чего нужен extern “C”?
10. Функции жизненного цикла динамических библиотек в Windows и Linux?

Сборка

|  |
| --- |
| --------------------------1---------------------------  cd C:\Users\ascoltat0\Desktop\Lab3\clang\ZSSLib  clang -target i386-pc-win32 -I.\include\ -c -o ..\bin\bsearch\_i.o .\src\bsearch\_i.c  clang -target i386-pc-win32 -I.\include\ -c -o ..\bin\bsearch\_r.o .\src\bsearch\_r.c  clang -target i386-pc-win32 -I.\include\ -c -o ..\bin\barray.o .\src\barray.c  llvm-ar cr ..\bin\ZSSLib.lib ..\bin\b\*.o  llvm-ranlib ..\bin\ZSSLib.lib  llvm-ar tv ..\bin\ZSSLib.lib02:35    --------------------------2---------------------------  cd C:\Users\ascoltat0\Desktop\Lab3\clang\Lab-03a  clang -target i386-pc-win32 -I..\ZSSLib\include\ -o ..\bin\Lab-03a.exe .\src\Lab-03a.c ..\bin\ZSSLib.lib   ..\bin\Lab-03a.exe ..\bin\Lab-03a.exe bsearch\_i 1024 ..\bin\Lab-03a.exe bsearch\_r 10    --------------------------3---------------------------  cd C:\Users\ascoltat0\Desktop\Lab3\clang\ZSSDLib  clang -target i386-pc-win32 -shared -I .\include\ -o ..\bin\ZSSDLib.dll .\src\ZSSDLib.c .\src\bsearch\_i.c .\src\bsearch\_r.c .\src\barray.c  clang -target i386-pc-win32 -shared -I .\include\ '-Wl,/def:.\src\ZSSDLib.def' -o ..\bin\ZSSDLib.dll .\src\ZSSDLib.c .\src\bsearch\_i.c .\src\bsearch\_r.c .\src\barray.c  dumpbin -exports ..\bin\ZSSDLib.dll  --------------------------4---------------------------  cd C:\Users\ascoltat0\Desktop\Lab3\clang\ZSSDLib++  clang++ -target i386-pc-win32 -shared -I .\include\ -o ..\bin\ZSSDLib++.dll .\src\ZSSDLib++.cpp .\src\bsearch\_i.cpp .\src\bsearch\_r.cpp .\src\barray.cpp  dumpbin -exports ..\bin\ZSSDLib++.dll  --------------------------5---------------------------  cd C:\Users\ascoltat0\Desktop\Lab3\clang\Lab-03b-ex  clang -target i386-pc-win32 -I .\include\ -L..\bin -o ..\bin\Lab-03b-ex.exe .\src\Lab-03b-ex.c -lZSSDLib -lZSSDLib++ -luser32  ..\bin\Lab-03b-ex ZSSDLib bsearch\_i 10 ..\bin\Lab-03b-ex ZSSDLib recursive\_search 10 ..\bin\Lab-03b-ex ZSSDLib++ bsearch\_r 10 ..\bin\Lab-03b-ex ZSSDLib++ ?bsearch\_r@@YAHPBHHHH@Z 10 ..\bin\Lab-03b-ex ZSSDLib 1 10 ..\bin\Lab-03b-ex ZSSDLib 2 10 ..\bin\Lab-03b-ex ZSSDLib++ 1 10 ..\bin\Lab-03b-ex ZSSDLib++ 2 10  dumpbin -imports ..\bin\Lab-03b-ex.exe06:38  --------------------------6---------------------------  cd C:\Users\ascoltat0\Desktop\Lab3\clang\Lab-03b-im  clang -target i386-pc-win32 -I .\include\ -L..\bin -o ..\bin\Lab-03b-im.exe .\src\Lab-03b-im.c -lZSSDLib -luser32    ..\bin\Lab-03b-im.exe bsearch\_i 10 ..\bin\Lab-03b-im.exe recursive\_search 10  --------------------------LINUX 7---------------------------  cd Lab3/ZSSLib  clang -I./include/ -c -o ../bin/bsearch\_i.o ./src/bsearch\_i.c clang -I./include/ -c -o ../bin/bsearch\_r.o ./src/bsearch\_r.c clang -I./include/ -c -o ../bin/barray.o ./src/barray.c llvm-ar cr ../bin/ZSSLib.a ../bin/b\*.o llvm-ranlib ../bin/ZSSLib.a  --------------------------8---------------------------  cd .. cd Lab-03a  clang -I../ZSSLib/include/ -o ../bin/Lab-03a ./src/Lab-03a.c ../bin/ZSSLib.a  ../bin/Lab-03a ../bin/Lab-03a bsearch\_i 1024 ../bin/Lab-03a bsearch\_r 10 |
| --------------------------9---------------------------  cd .. cd LibZSSd  clang -target x86\_64-pc-linux -I./include/ -o ../bin/LibZSSd.o -fPIC -c ./src/LibZSSd.c clang -target x86\_64-pc-linux -shared -o ../bin/[LibZSSd.so](https://libzssd.so/) ./src/LibZSSd.c  --------------------------10---------------------------  cd .. cd Lab-03c  clang -o ../bin/Lab-03c ./src/Lab-03c.c -ldl ../bin/Lab-03c ../bin/[LibZSSd.so](https://libzssd.so/) bsearch\_i 10  cd Lab3/bin  readelf -s [LibZSSd.so](https://libzssd.so/)  --------------------------СMAKE---------------------------  cd C:\Users\ascoltat0\Desktop\Lab3  cmake -B out -G "Visual Studio 17 2022" -T ClangCL  cmake --build ./out |