1. Реализация функции в виде статической библиотеки. Вызов функции из статической библиотеки.
2. Реализация трех функций, которые вызывают друг друга по цепочке (third->second->first), в виде двух динамических библиотек. Вызов цепочки функции из динамической библиотеки.

# Реализация функции в виде статической библиотеки. Вызов функции из статической библиотеки

Библиотеки позволяют использовать разработанный ранее программный код в различных программах. Таким образом, программист может не разрабатывать часть кода для своей программы, а воспользоваться тем, что входит в состав библиотек.

В языке программирования C код библиотек представляет собой функции, размещенные в файлах, которые скомпилированы в объектные файлы, а те, в свою очередь, объединены в библиотеки. В одной библиотеке объединяются функции, решающие определенный тип задач. Например, существует библиотека математических функций.

У каждой библиотеки должен быть свой заголовочный файл, в котором должны быть описаны прототипы (объявления) всех функций, содержащихся в этой библиотеке. С помощью заголовочных файлов вы "сообщаете" вашему программному коду, какие библиотечные функции есть и как их использовать. Для компиляции проекта достаточно иметь только одну библиотеку.

Исходный код библиотеки состоит из простейшей функции, вычисляющей сумму двух слагаемых, передаваемых как параметры.

Листинг 1.1 Исходный код библиотеки (файл lr6\_1Mysum.c)

int my\_sum(int a, int b)

{

return a+b;

}

Для компиляции проекты так же был создан заголовочный файл mysum.h.

Листинг 1.2 Содержимое файла mysum.h

#ifndef MYSUM\_H

#define MYSUM\_H

int my\_sum(int a, int b);

#endif

Статическая библиотека создается из обычных объектных файлов путем их архивации с помощью утилиты ar. Сначала требуется получить объектные файлы, для этого используется команда cc -c mysum.c. Далее используем утилиту ar для создания статической библиотеки: ar r libmysum.a mysum.o. Объектные файлы не нужны, поэтому их можно удалить: rm \*.o. В результате получим статическую библиотеку mysum.a

Для компиляции проекта со статической библиотекой сначала требуется скомпилировать исходный код программы командой cc -c main.c

Листинг 1.3 Исходный код программы (файл main.c)

#include "./mysum.h"

#include <stdio.h>

#include <stdlib.h>

int main(int argc, char\* argv[])

{

int a = atoi(argv[1]);

int b = atoi(argv[2]);

printf("sum = %d\n", my\_sum(a, b));

}

Далее требуется скомпилировать файл main.o в исполняемый файл, объединив со статической библиотекой libmysum.a. Делается это с помощью следующей команды:

cc -static -o main main.o -L. -lmysum

Опция -o указывает на то, что компилируется исполняемый файл main из объектных файлов.

Помимо объектных файлов проекта в компиляции участвует и библиотека. Об этом свидетельствует вторая часть команды: -L. -lmysum. Здесь опция -L указывает на адрес каталога, где находится библиотека, он и следует сразу за ней. После опции -l записывается имя библиотеки, при этом префикс lib и суффикс .a усекается.

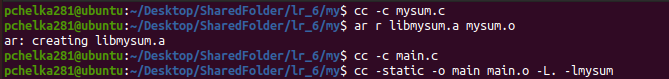


Рисунок 1.1 – Создание библиотеки

Результат запуска скомпилированной программы приведен на рисунке 1.2



Рисунок 1.2 – Результат запуска программы

# Вызов цепочки из трех функций из динамической библиотеки

Объектные файлы для динамической библиотеки компилируются особым образом. Они должны содержать так называемый позиционно-независимый код (position independent code). Наличие такого кода позволяет библиотеке подключаться к программе, когда последняя загружается в память. Это связано с тем, что библиотека и программа не являются единой программой, а значит, как угодно, могут располагаться в памяти относительно друг друга. Компиляция объектных файлов для динамической библиотеки должна выполняться с опцией -fPIC компилятора cc:

cc -c -fPIC library.c

В отличие от статической библиотеки динамическую создают при помощи cc указав опцию -shared:

cc -shared -o liblibrary.so library.o

Использованные объектные файлы можно удалить:

rm library.o

Результат запуска скомпилированной программы приведен на рисунке 2.

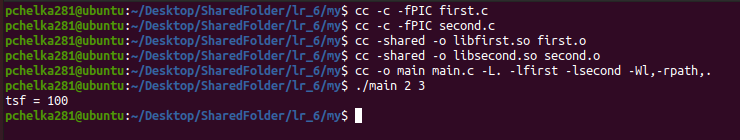


Рисунок 2. – Результат работы приложения

# Вывод

Была реализована функция в виде статической библиотеки. Так же был произведен вызов функции из статической библиотеки.

Для изучения работы с динамическими библиотеками были реализованы три функции, которые вызывают друг друга по цепочке (third->second->first), в виде двух динамических библиотек. Вызов цепочки функции из динамической библиотеки.

Были изучены библиотеки всех видов — статические и динамические. Код первых при компиляции полностью входит в состав исполняемого файла, что делает программу легко переносимой. Код динамических библиотек не входит в исполняемый файл, последний содержит лишь ссылку на библиотеку. Если динамическая библиотека будет удалена или перемещена в другое место, то программа работать не будет. С другой стороны, использование динамических библиотек позволяет сократить размер исполняемого файла. Также если в памяти находится две программы, использующие одну и туже динамическую библиотеку, то последняя будет загружена в память лишь единожды.