МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет информационных технологий и робототехники

Кафедра программного обеспечения информационных систем и технологий

**Отчет по лабораторной работе № 8**

по дисциплине: «Системное программирование»

на тему: «Создание и использование библиотек в Linux»

Выполнила: ст. гр. 10702221

Багапов Д.М.

Матвейчук К.А.

Яцук З.Р.

Проверил: Давыденко Н.В.

Минск 2023

**ЛАБОРАТОРНАЯ РАБОТА # 8**

**Создание и использование библиотек в Linux   
Цель работы :**

Изучить и закрепить на практике создание и использование статически и динамически подключаемых библиотек в операционных системах семейства Linux.

**Индивидуальные задания первой лабораторной:**

**Задание 1:**

Произвести рефакторинг проекта предыдущей лабораторной работы по

автосборке проекта: вынесите код функций бизнес логики в отдельную

статическую библиотеку. Для автосборки проекта используйте утилиту

make, обязательно.

При помощи программы ldd проверим зависимости, образовавшиеся между

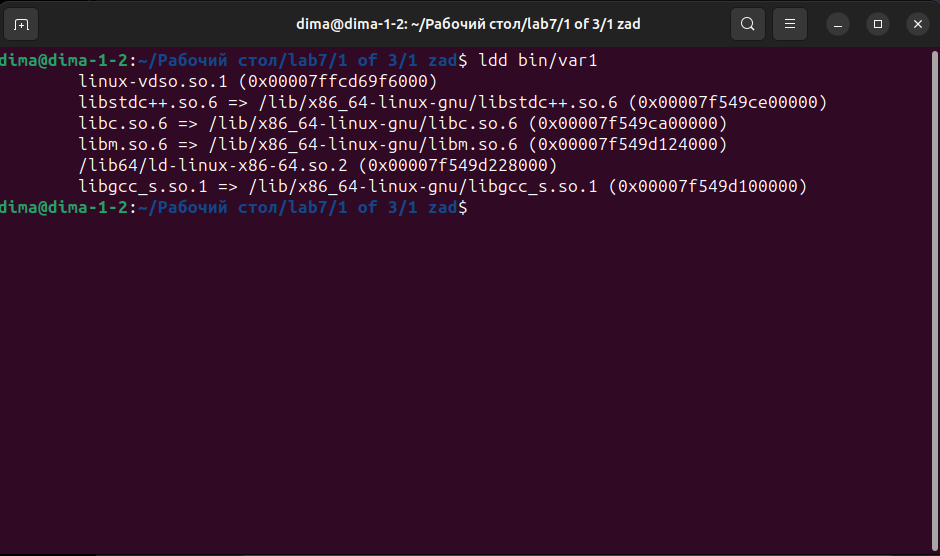
библиотеками.

Сравните сумму размеров объектных файлов проекта и исполняемого файла.

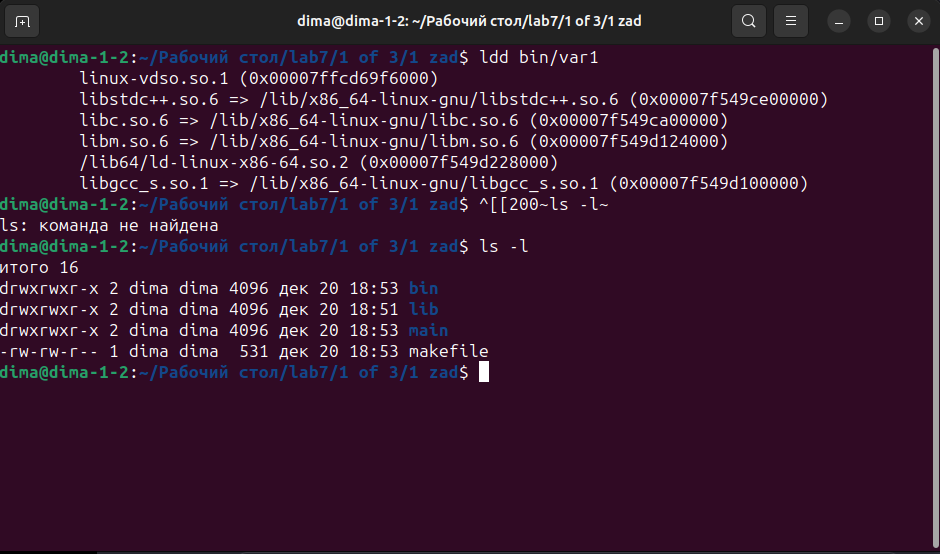
Результаты отобразите скриншотами.

Копию исполняемого файла расположите на «Рабочем столе», запустите программу. Какой будет результат?

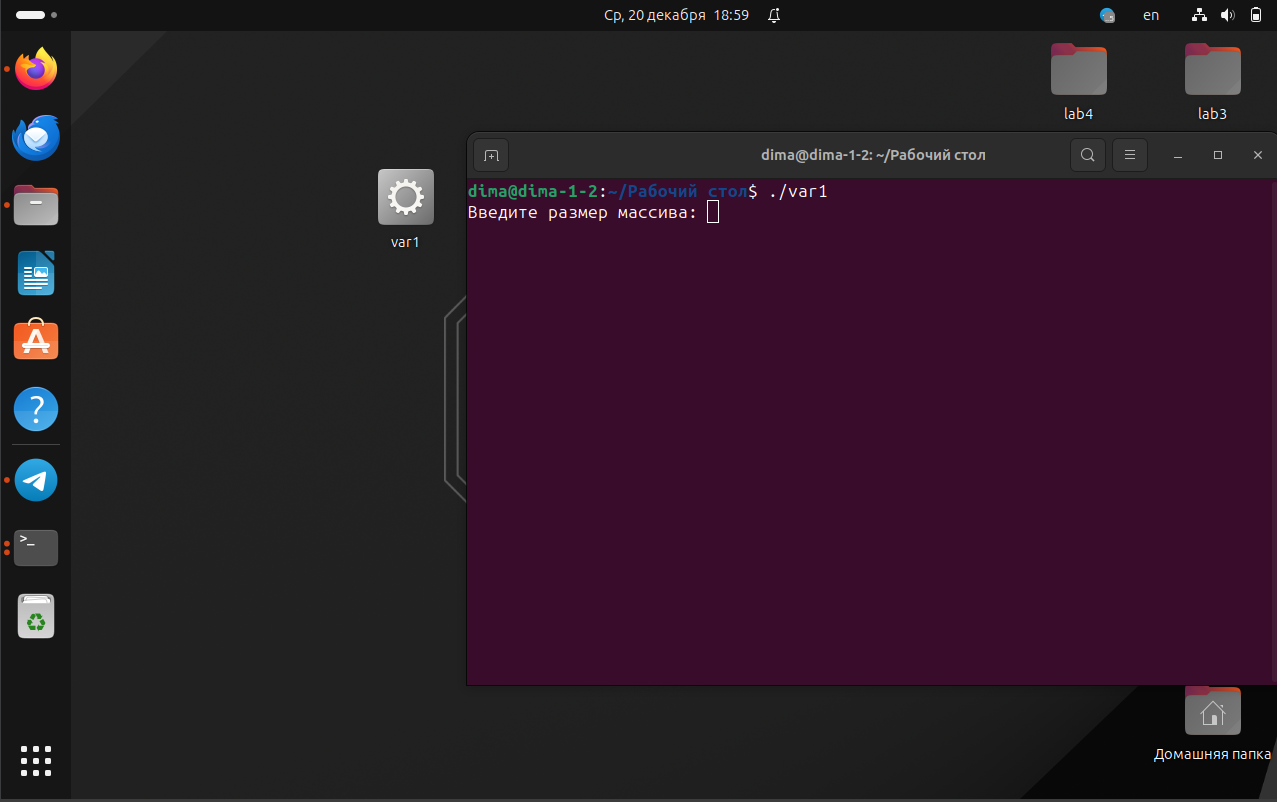
**Выполнение:**



**Проверка зависимостей успешна**



**Вывод размеров файлов**



**Файл успешно скопирован на рабочий стол и выполняется**

**Задание 2:**

Произвести рефакторинг проекта предыдущей лабораторной работы по

автосборке проекта: вынесите код функций бизнес логики в отдельную

статическую библиотеку. Для автосборки проекта используйте утилиту

make, обязательно.

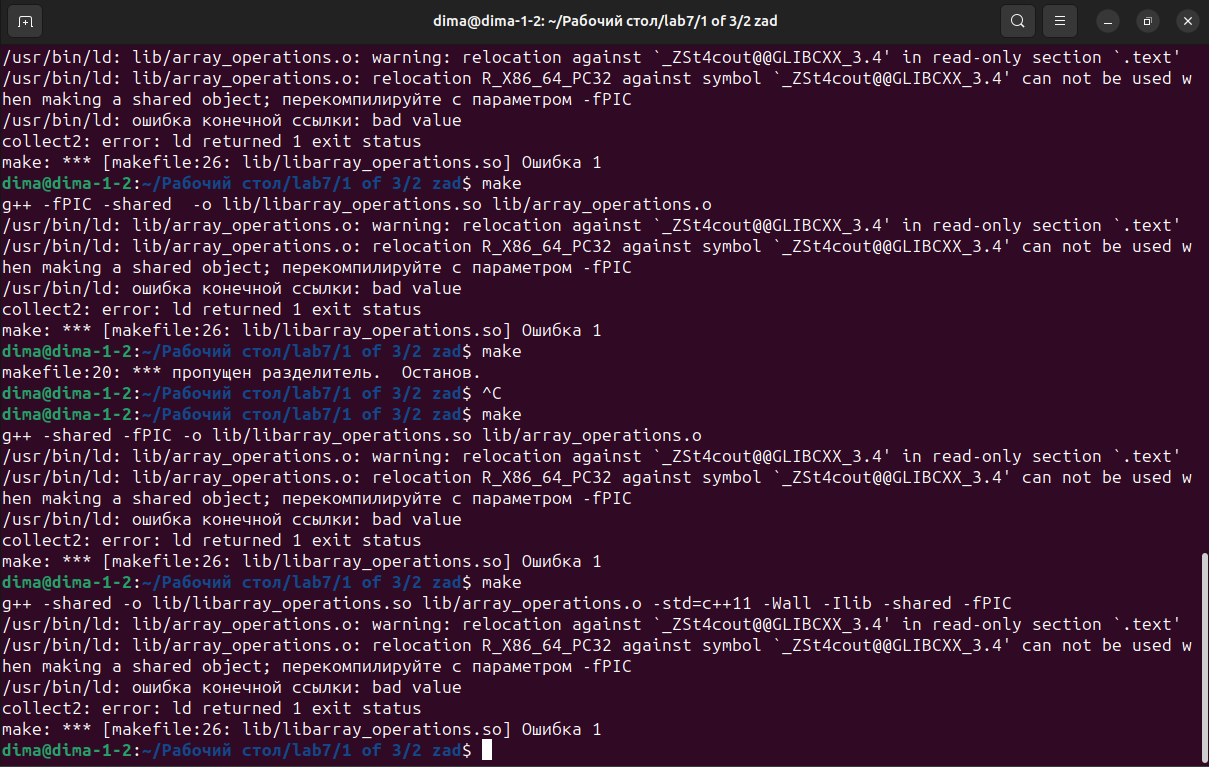
При помощи программы ldd проверим зависимости, образовавшиеся между

библиотеками.

Сравните сумму размеров объектных файлов проекта и исполняемого файла.

Результаты отобразите скриншотами.

Копию исполняемого файла расположите на «Рабочем столе», запустите программу. Какой будет результат?

 **Неудача**

В ходе выполнения этого задания я столкнулся с ошибкой обойти которую у меня так и не вышло, почему-то флаг –fPIC никак не хотел применяться, что бы я не делал. Прошу понять и простить

**Задание 3:**

Произвести рефакторинг проекта предыдущей лабораторной работы по

автосборке проекта: вынесите код функций бизнес логики в отдельную

динамическую библиотеку с динамическим вызовом. Для автосборки проекта

используйте утилиту make, обязательно.

Создание и использование библиотек в Linux

Operating Systems and System Programming | 21

При помощи программы ldd проверим зависимости, образовавшиеся между

библиотеками.

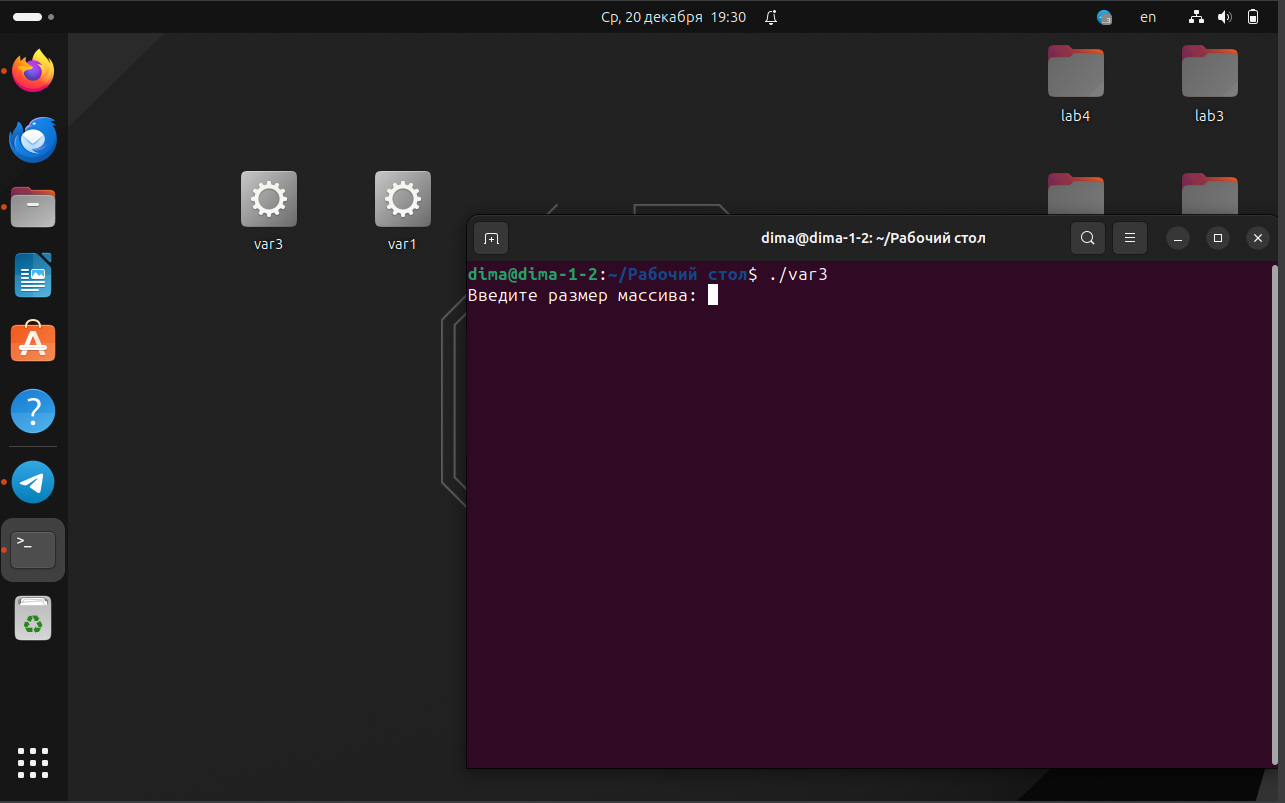
Сравните сумму размеров объектных файлов проекта и исполняемого файла.

Результаты отобразите скриншотами.

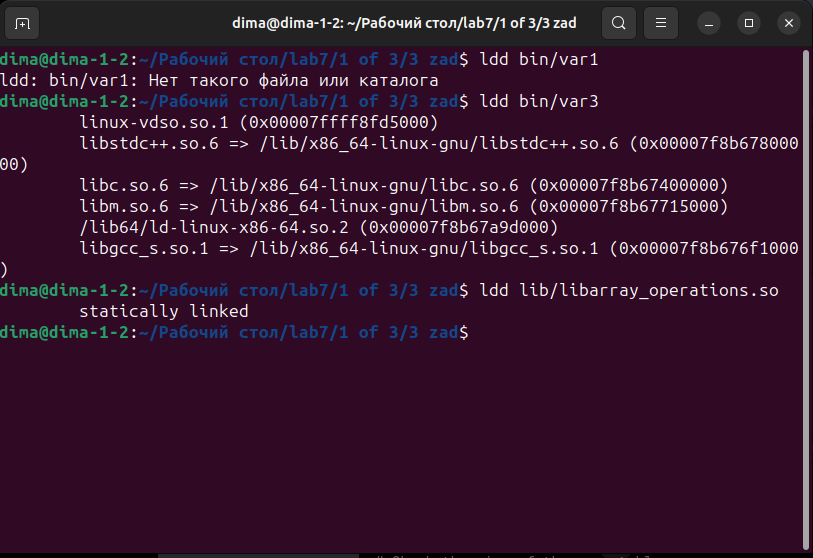
Копию исполняемого файла расположите на «Рабочем столе», запустите программу. Какой будет результат? Если возникла проблема, решите ее и поясните каким способом.

Сравните результаты с предыдущими упражнения, сделайте вывод.

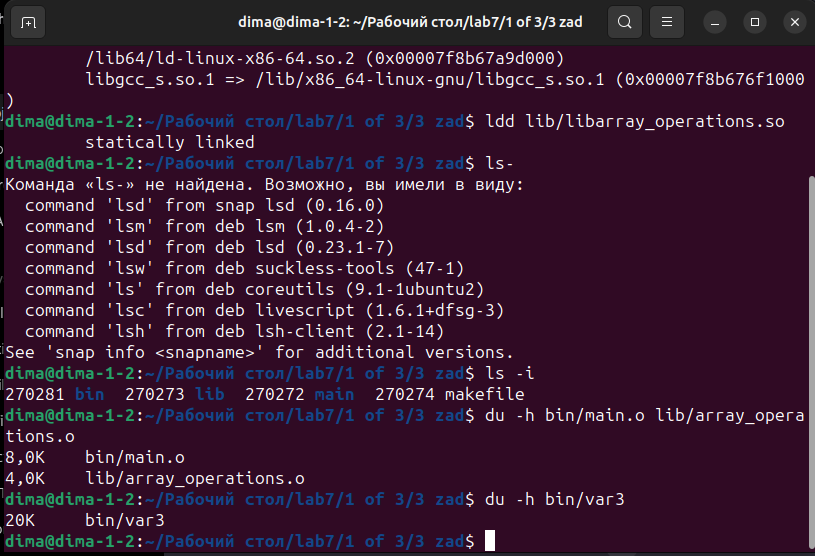
**Выполнение:**



Успешное копирование на рабочий стол и выполнение



Зависимости библиотек



Размеры файлов до и после компиляции

**Индивидуальные задания первой лабораторной:**

Для проверки остаточных знаний учеников после рождественских каникул,

учитель младших классов решил начинать каждый урок с того, чтобы задавать

каждому ученику пример из таблицы умножения, но в классе, к примеру, 15

человек, а примеры среди них не должны повторяться. В помощь учителю

разработайте универсальную библиотечную функцию, которая будет выводить на экран 15 (или N) случайных примеров из таблицы умножения (к примеру, от 2\*2 до 9\*9, потому что задания по умножению на 1 и на 10 – слишком

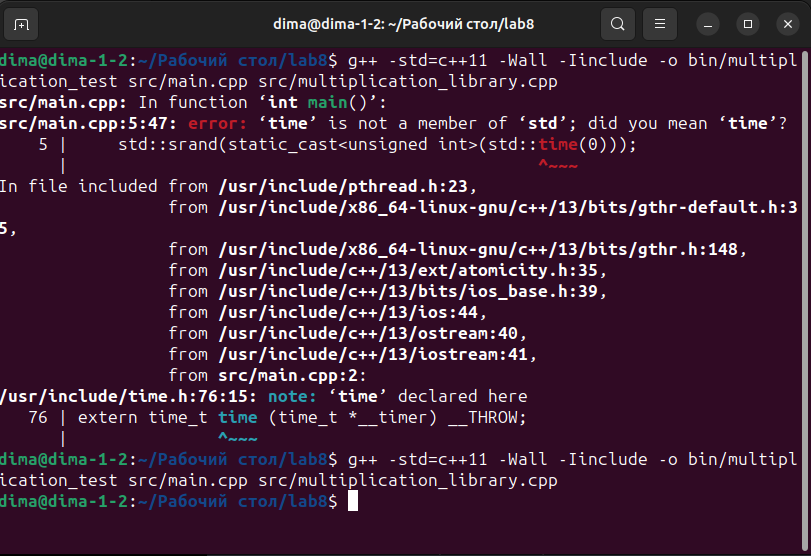
просты). При этом среди 15 примеров не должно быть повторяющихся (примеры 2\*3 и 3\*2 и им подобные пары считать повторяющимися).

Алгоритм поиска данных комбинаций должен быть эффективным!!!

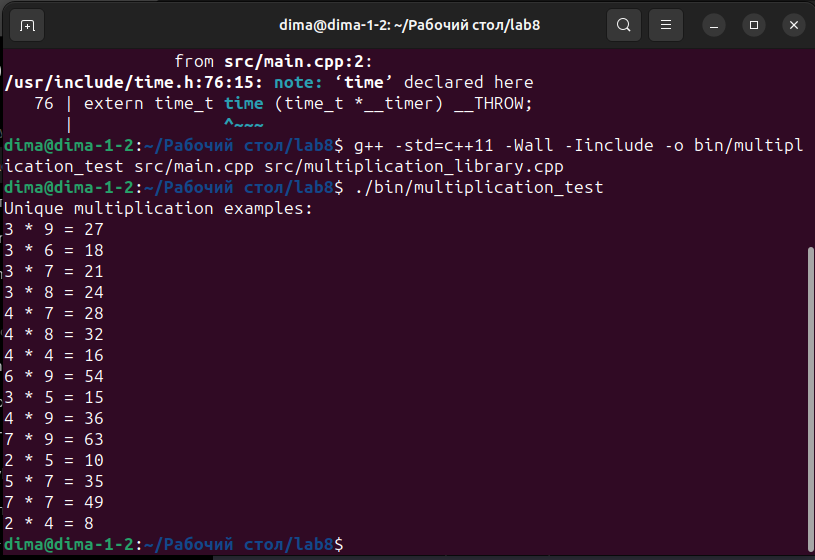
Напишите тестовую программу, которая бы демонстрировала работоспособность данной библиотечной функции. Произвести рефакторинг предыдущей лабораторной работы: вынесите код

функций бизнес логики в отдельную библиотеку

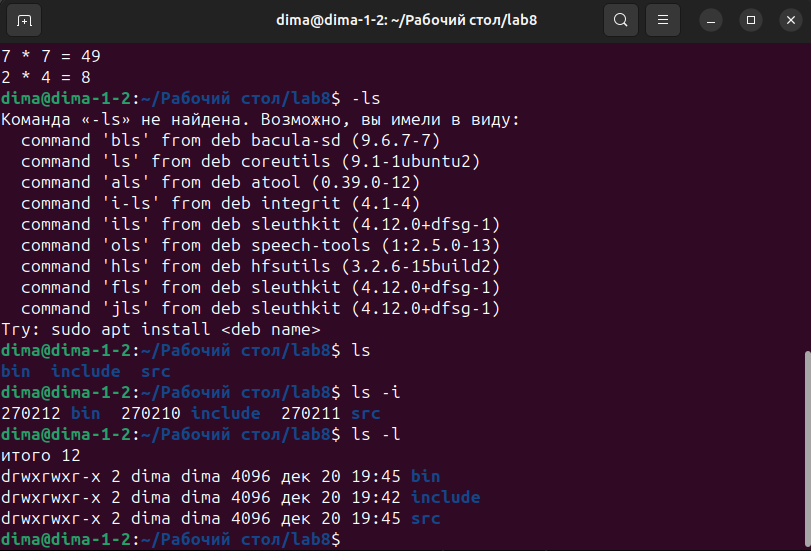
**Выполнение:**



Успешная компиляция



Успешное выполнение



Правила бизнес логики учтены

**Контрольные вопросы:**

**1) Что такое библиотеки и для чего они используются?**

- Библиотеки представляют собой совокупность функций и ресурсов, предназначенных для повторного использования в различных программах. Они упрощают разработку, обеспечивают структурирование кода и сокращают объем повторяющегося кода.

**2) Чем отличается заголовочные файлы (файлы с расширением .h) от файлов библиотек?**

- Заголовочные файлы содержат объявления функций, типов данных и другие декларации, необходимые для использования функциональности библиотеки. Файлы библиотек содержат скомпилированный код, который используется в программе.

**3) Опишите существующие типы библиотек и принципы их использования.**

- Существуют статические и динамические библиотеки. Статические включают свой код непосредственно в исполняемый файл, а динамические подгружаются программой во время выполнения.

**4) Опишите преимущества и недостатки каждого из типа библиотек.**

- Статические библиотеки:

- Преимущества: Простота в использовании, отсутствие зависимостей во время выполнения.

- Недостатки: Увеличение размера исполняемого файла, обновление всей программы при изменении библиотеки.

- Динамические библиотеки:

- Преимущества: Экономия памяти, возможность обновления библиотеки без изменения самой программы.

- Недостатки: Зависимость от наличия правильной версии библиотеки во время выполнения, сложности в управлении версиями.

**5) Как и какими способами подключить библиотеку к программе? Опишите специфику подключения.**

- В C++ можно подключить библиотеку двумя основными способами: статически и динамически. Статическое подключение происходит во время компиляции, а динамическое – во время выполнения.

**6) На какой стадии происходит подключение библиотек (объектных файлов) к основному модулю программы?**

- Подключение библиотек происходит на стадии компиляции и/или линковки. В случае статических библиотек код библиотеки встраивается в исполняемый файл, а для динамических библиотек создается связь (линковка) с библиотекой.

**7) Как создать статическую и динамическую библиотеки и подключить их к программе?**

- Статическая библиотека:

- Создать объектные файлы: `g++ -c -o library.o library.cpp`

- Собрать статическую библиотеку: `ar rcs liblibrary.a library.o`

- Подключить статическую библиотеку к программе в Makefile: `-llibrary`

- Динамическая библиотека:

- Создать объектные файлы: `g++ -fPIC -c -o library.o library.cpp`

- Собрать динамическую библиотеку: `g++ -shared -o liblibrary.so library.o`

- Подключить динамическую библиотеку к программе в Makefile: `-L/path/to/library -llibrary`

**8) Что такое «позиционно-независимый код» (position independent code, PIC)?**

- PIC — это код, который можно загрузить и выполнить в любом месте памяти, независимо от фактической позиции в памяти во время компоновки. Это важно для динамических библиотек, так как они загружаются в разные адреса в памяти.

**9) Какие способы можно использовать для того, чтобы указать местоположение подключаемых динамических библиотек?**

- Местоположение подключаемых динамических библиотек можно указать с использованием переменной окружения `LD\_LIBRARY\_PATH`, в файле конфигурации `/etc/ld.so.conf` или с помощью флага `-rpath` при компиляции.

**10) Если в целевом каталоге присутствует два типа одной и той же библиотеки, какая из них всегда будет подключаться по умолчанию? А как явно подключить вторую?**

- По умолчанию будет использоваться динамическая библиотека. Для явного подключения второй библиотеки можно использовать флаг `-static` при компиляции, чтобы использовать статическую версию.

**11) Какие проблемы могут быть при обновлении совместно используемых библиотек, и что такое «Ад DLL» (DLL HELL)?**

- Проблемы могут включать в себя конфликты версий, где приложение, созданное для одной версии библиотеки, несовместимо с другой версией. "Ад DLL" (DLL Hell) — это ситуация, когда различные программы требуют разные версии одной и той же динамической библиотеки, и установка одной версии может повлиять на работу других программ. Это особенно актуально для операционных систем Windows.