**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

****

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ЛИПЕЦКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

|  |  |
| --- | --- |
| Институт | компьютерных наук |
| Кафедра | автоматизированных систем управления |

ЛАБОРАТОРНАЯ РАБОТА №1

по

«Средства разработки программного обеспечения в ОС Linux»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ПИ-22-1 | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | Первеева Е. Ю. |
| Руководитель | |  |  |  |
| канд. техн. наук, доцент  ученая степень, ученое звание | |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | Кургасов В.В. |

Липецк, 2024 г.

**Оглавление**

[Задание 3](#_Toc179832825)

[Цель работы 3](#_Toc179832826)

[Ход работы 4](#_Toc179832827)

[Вывод 8](#_Toc179832828)

[Контрольные вопросы 8](#_Toc179832829)

# Задание

* Установить гипервизор, изучить понятия: Виртуализация, гипервизоры, изучить создание виртуальных машин.
* Установить 64 серверную ОС на выбор (Ubuntu, Debian)
* Написать программу. Дан одномерный массив из N целочисленных элементов (тип int). Все отрицательные элементы массива заменить их модулем и подсчитать их количество.

# Цель работы

Получить представление о технологиях и средствах разработки программного обеспечения в ОС Linux.

# Ход работы

Для установки серверного Debian в качестве гостевой операционной системы, установим и используем гипервизор VirtualBox. Результат установки представлен на рисунке 1.

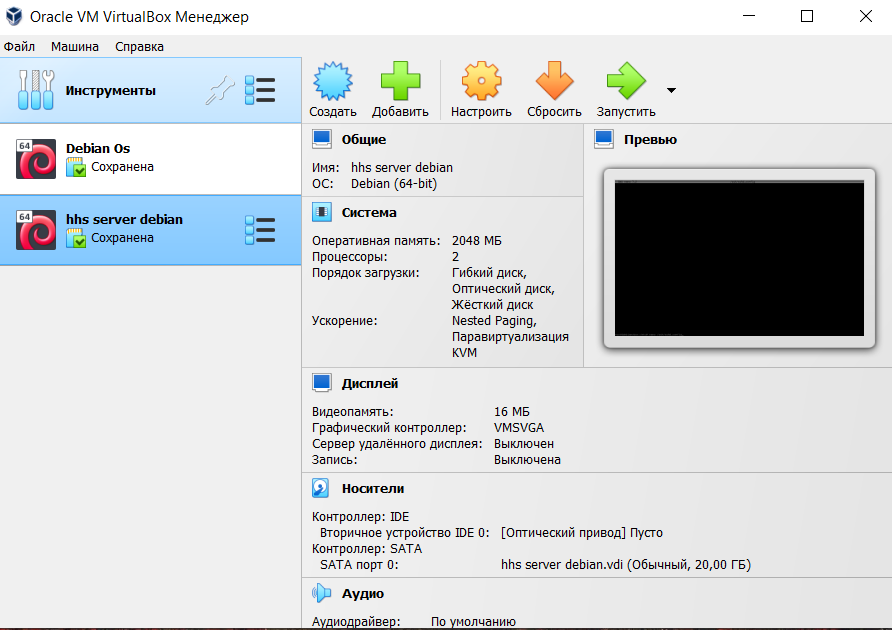


Рисунок 1 - Установка гостевой ОС

После скачивания и настройки запускаем серверный Debian и оказываемся в домашней директории пользователя.

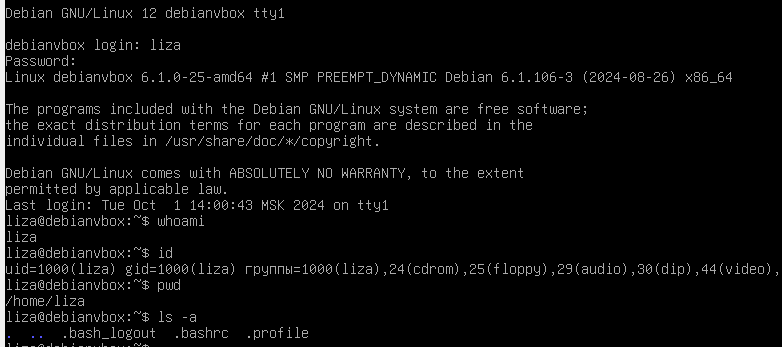


Рисунок 2 - Домашняя директория пользователя

Необходимо написать программу, выполняющую условие: дан одномерный массив из N целочисленных элементов (тип int). Все отрицательные элементы массива заменить их модулем и подсчитать их количество. Для этого используем специализированный текстовый редактор Vim.

Для изменения содержимого файла в текстовом редакторе Vim переходим в режим вставки. Для сохранения файла и выхода из редактора переходим в режим команд и пишем «:wq». Написанная программа на языке C в текстовом редакторе Vim представлена на рисунке 3.

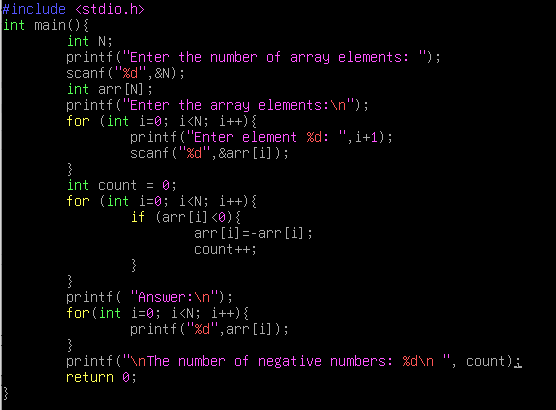


Рисунок 3 - Код требуемой программы

Сохраним программу в файле “script1.c”.

Установим GCC и GDB. **GDB** (GNU Debugger) — это отладчик для программ, написанных на языках программирования, таких как C, C++.

GCC (GNU Compiler Collection) — это компилятор, который используется для компиляции программ, написанных на C, C++.

Для компиляции программы на C через компилятор GCC, написанной в файле “script1.c”, используем команду: gcc -g program.c -o program. Опция -g нужна чтобы включить отладочную информацию для GDB. Эта команда создаст исполняемый файл с именем script1. После компиляции который можно впоследствии запустить. Пример запуска программы представлен на рисунке 4.

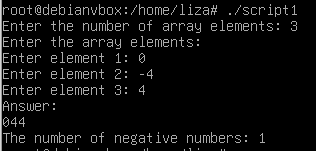


Рисунок 4 - Запуск программы при помощи компилятора GCC

Зайдем в отладчик GDB с исполняемым файлом: gdb ./program и запустим программу через отладчик: (gdb) run. Результат представлен на рисунке 5.

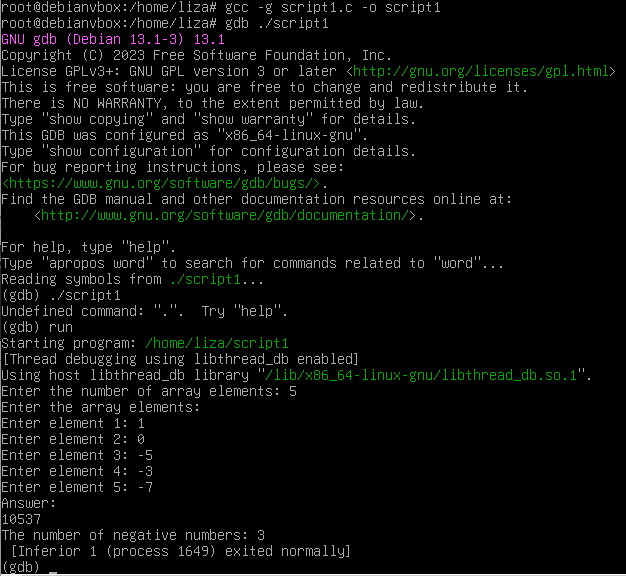


Рисунок 5 - Запуск программы через отладчик GDB

# Вывод

Таким образом, в ходе работы были получены представления о технологиях и средствах разработки программного обеспечения в ОС Linux.

# Контрольные вопросы

**1) Что такое IDE?**  
 IDE (интегрированная среда разработки) — это программный комплекс, который предоставляет разработчикам единое окружение для написания, компиляции, отладки и выполнения кода. IDE обычно включает в себя редактор кода, компилятор или интерпретатор, отладчик и другие инструменты для управления проектами.

**2) Что такое API?**  
 API (интерфейс программирования приложений) — это набор функций, протоколов и инструментов, позволяющий одной программе взаимодействовать с другой. API предоставляет стандартизированный доступ к функциональности другого приложения или сервиса, скрывая детали реализации.

**3) Что такое библиотека в программировании?**  
 Библиотека — это коллекция заранее написанных функций и классов, предназначенных для решения типичных задач, таких как работа с данными, ввод-вывод или математические вычисления. Использование библиотек позволяет разработчикам не писать повторяющийся код с нуля, а применять готовые решения.

**4) Что такое статические и динамические библиотеки?**  
 Статическая библиотека связывается с программой на этапе компиляции и становится её частью. Все функции, необходимые программе, включаются в итоговый исполняемый файл, что увеличивает его размер, но делает программу независимой от внешних библиотек.  
 Динамическая библиотека загружается в память при выполнении программы. Программы используют её функции по мере необходимости, что уменьшает размер исполняемого файла и позволяет обновлять библиотеки отдельно от программы.

**5)Что такое плагин?**  
 Плагин — это расширение для программы, которое добавляет новые функции без изменения её исходного кода. Плагины позволяют модифицировать и дополнять функционал программы, сохраняя её гибкость и расширяемость.

**6)Примеры консольных текстовых редакторов для Linux.**  
 Некоторые из популярных текстовых редакторов, работающих в терминале Linux, включают: **nano**, **vim**, и **emacs**. Эти редакторы позволяют редактировать файлы напрямую через командную строку.

**7)Что делает команда gcc?**  
 Команда gcc — это компилятор, используемый для компиляции исходного кода на языках программирования, таких как C и C++. Она преобразует исходный код в исполняемый файл.

**8)Что делает команда make?**  
 Команда make автоматизирует процесс сборки программ, используя инструкции из файла Makefile. Она компилирует только те файлы, которые были изменены, что ускоряет процесс сборки больших проектов.

**9)Что делает команда gdb?**  
 Команда gdb запускает отладчик, который помогает выявлять ошибки в программах, написанных на C и C++. С помощью GDB можно выполнять программу по шагам, проверять значения переменных и отслеживать поведение программы.

**10)Что такое заголовочный файл и файл реализации?**  
 Заголовочный файл (.h) содержит объявления функций, классов и переменных, которые могут использоваться в других частях программы.

**file.h** — объявление функции void foo();  
 Файл реализации (.c или .cpp) содержит определения функций и классов, описанных в заголовочном файле.

**file.cpp** — реализация функции void foo() { ... }

**11) Что такое единица трансляции? В чем особенность разработки программ из нескольких единиц трансляции?**  
 Единица трансляции — это исходный файл вместе с включёнными в него заголовочными файлами, который компилятор обрабатывает как отдельную сущность во время компиляции. Каждый такой файл после компиляции преобразуется в объектный файл (.o или .obj).

Использование нескольких единиц трансляции делает программу более организованной, ускоряет процесс компиляции и способствует повторному использованию кода, что особенно важно для крупных проектов.

**12) Дайте краткую характеристику каждому этапу трансляции программ, написанных на Си.**

1. Предпроцессинг - на этом этапе компилятор обрабатывает директивы препроцессора, такие как #include, #define и другие. Предпроцессор заменяет макросы, вставляет содержимое заголовочных файлов и выполняет другие текстовые замены. Выходной файл: расширенный исходный код без директив препроцессора.

2. Компиляция - на этапе компиляции расширенный исходный код преобразуется в ассемблерный код. Компилятор проверяет синтаксис программы и генерирует низкоуровневые инструкции, которые соответствуют исходному коду. Выходной файл: файл с ассемблерным кодом

3. Ассемблирование - на этом этапе ассемблерный код преобразуется в объектный код, который представляет собой двоичный код, но ещё не является исполняемым файлом. Объектный файл содержит машинные инструкции, которые понимает процессор. Выходной файл: объектный файл (.o или .obj).

4. Компоновка - на этапе компоновки происходит объединение всех объектных файлов и подключаемых библиотек в единый исполняемый файл. Выходной файл: исполняемый файл (.exe, .out).