Отчёт по лабораторной работе №9

Дисциплина: Архитектура компьютера

Бражко Александра Александровна

Содержание

# 1 Цель работы

Приобретение навыков написания программ с использованием подпрограмм. Знакомство с методами отладки при помощи GDB и его основными возможностями.

# 2 Задание

1. Реализация подпрограмм в NASM
2. Отладка программам с помощью GDB
3. Задание для самостоятельной работы

# 3 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы.

Например, в табл. 1 приведено краткое описание стандартных каталогов Unix.

Таблица 1: Описание некоторых каталогов файловой системы GNU Linux

| Имя каталога | Описание каталога |
| --- | --- |
| / | Корневая директория, содержащая всю файловую |
| /bin | Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям |
| /etc | Общесистемные конфигурационные файлы и файлы конфигурации установленных программ |
| /home | Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя |
| /media | Точки монтирования для сменных носителей |
| /root | Домашняя директория пользователя root |
| /tmp | Временные файлы |
| /usr | Вторичная иерархия для данных пользователя |

Более подробно про Unix см. в [1–4].

# 4 Выполнение лабораторной работы

Создаём каталог для выполнения лабораторной работы № 9, переходим в него и создаём файл lab9-1.asm (рис. 1).

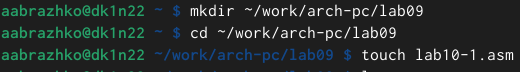


Рис. 1: Создание каталога и файла

Вводим в файл lab9-1.asm текст программы из листинга 9.1 (рис. **¿fig:002?**).

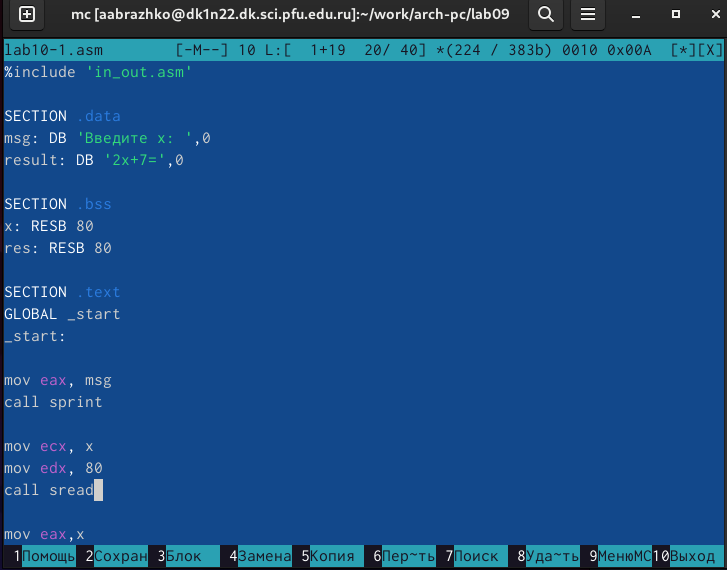


Рис. 2: Ввод листинга

Создаём исполняемый файл и проверяем его работу (рис. 3).

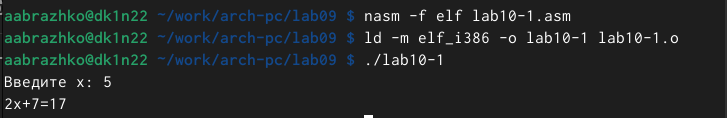


Рис. 3: Создание и проверка

Вносим изменения в программу, чтобы решалось выражение f(g(x)) (рис. 4).

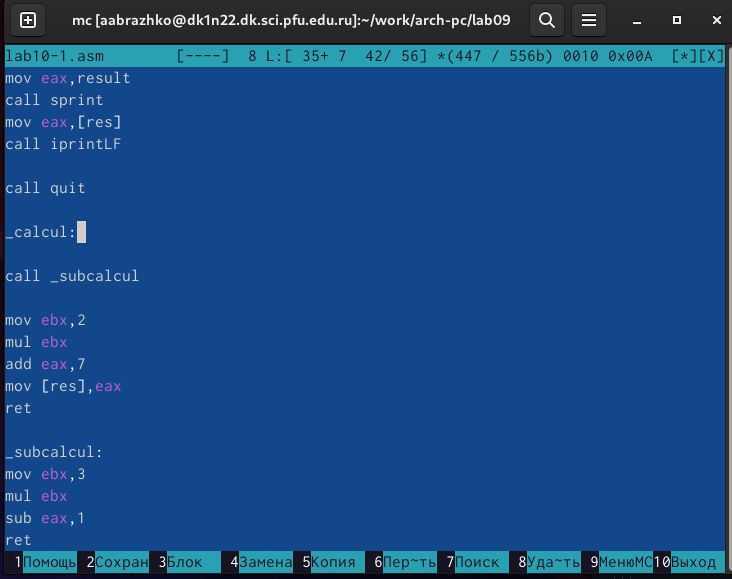


Рис. 4: Изменение программы

Создаём исполняемый файл и проверяем его работу (рис. 5).

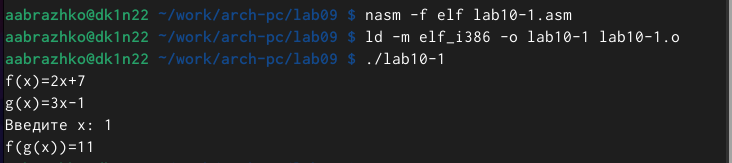


Рис. 5: Создание и проверка

Создаём файл lab9-2.asm (рис. 6).

Рис. 6: Создание файла

Рис. 6: Создание файла

Вводим в файл lab9-2.asm текст программы из листинга 9.2 (рис. 7).

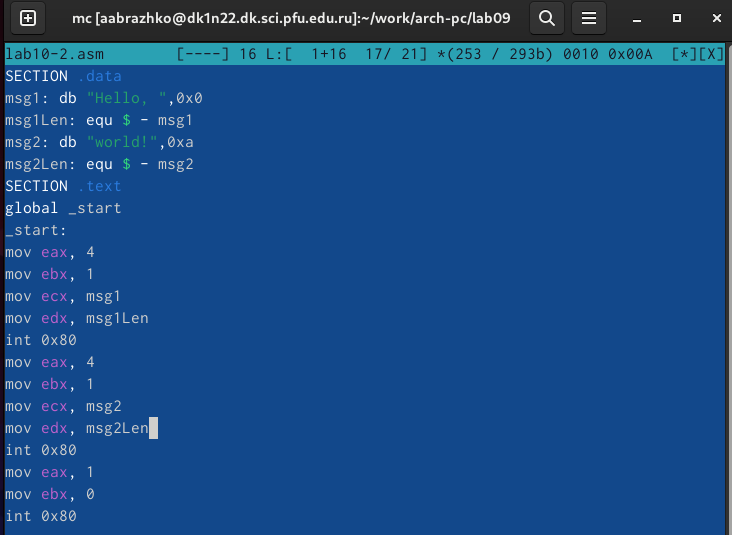


Рис. 7: Ввод листинга

Загружаем и запускаем файл в откладчике gdb (рис. 8).

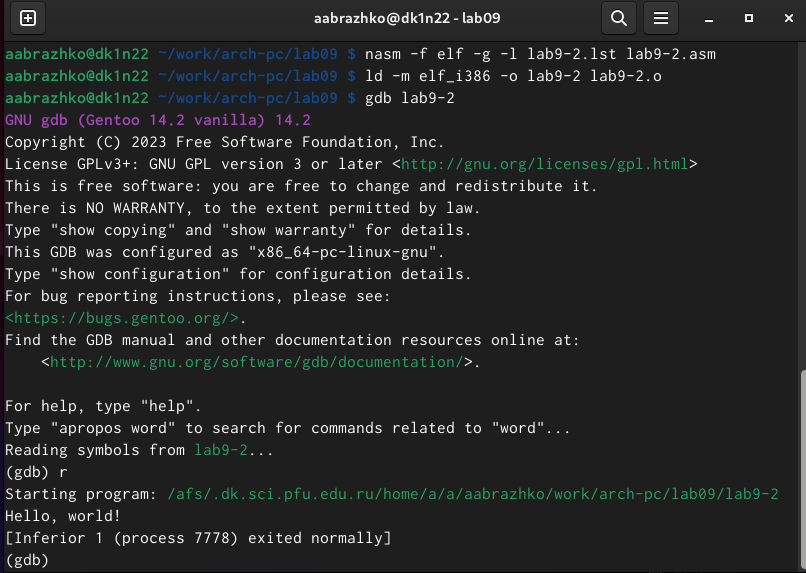


Рис. 8: Запуск и загрузка

Ставим брекпоинт на метку \_start и запускаем программу (рис. 9).

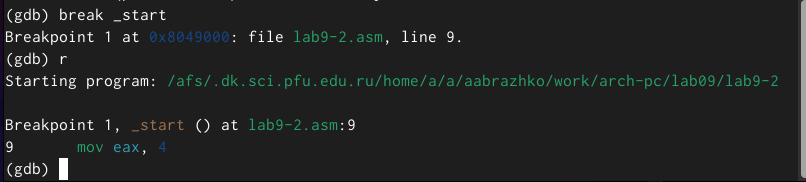


Рис. 9: Запуск

Просматриваем дисассимплированный код программы, начиная с метки (рис. 10).

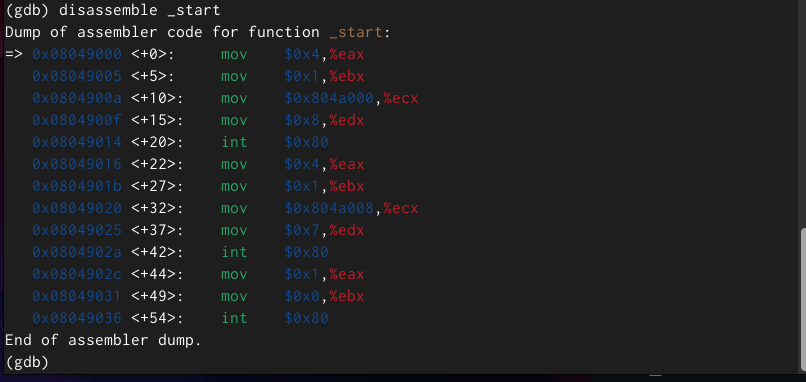


Рис. 10: Просмотр кода

Переключаемся на отображение команд с Intel’овским синтаксисом (рис. 11).



Рис. 11: Переключение

Включим режим псевдографики для более удобного анализа программы (рис. 12).

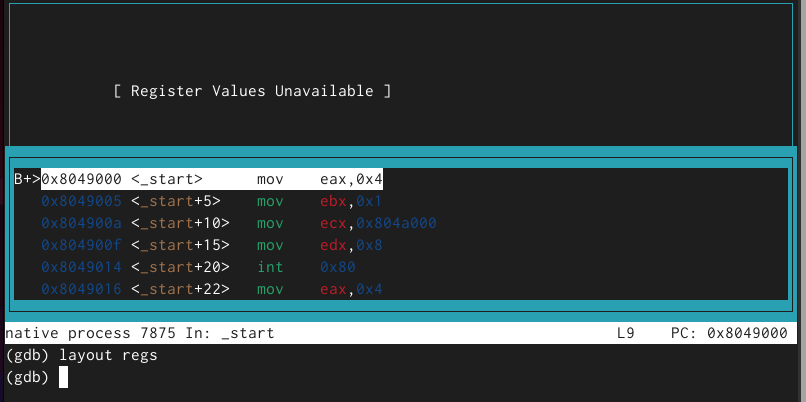


Рис. 12: Режим псевдографики

Просмотрим наличие меток и добавим еще одну метку на предпоследнюю инструкцию (рис. 13).

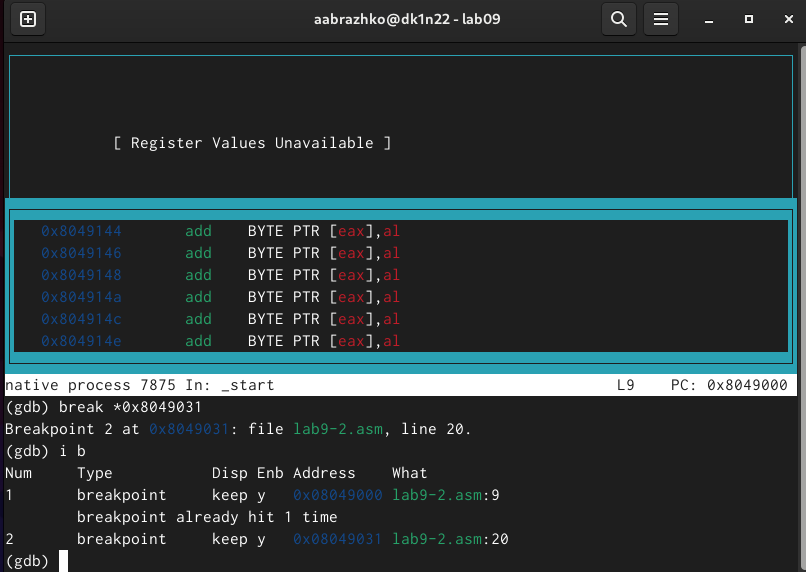


Рис. 13: Просмотр и добавление метки

С помощью команды si мы посмотрим регистры и изменим их (рис. 14, рис. 15).

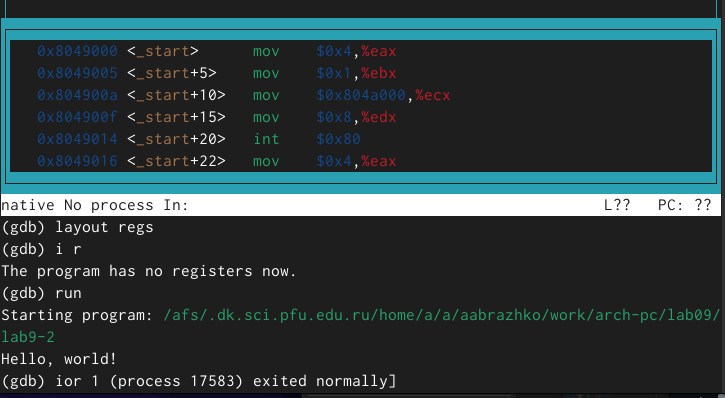


Рис. 14: Просмотр и замена

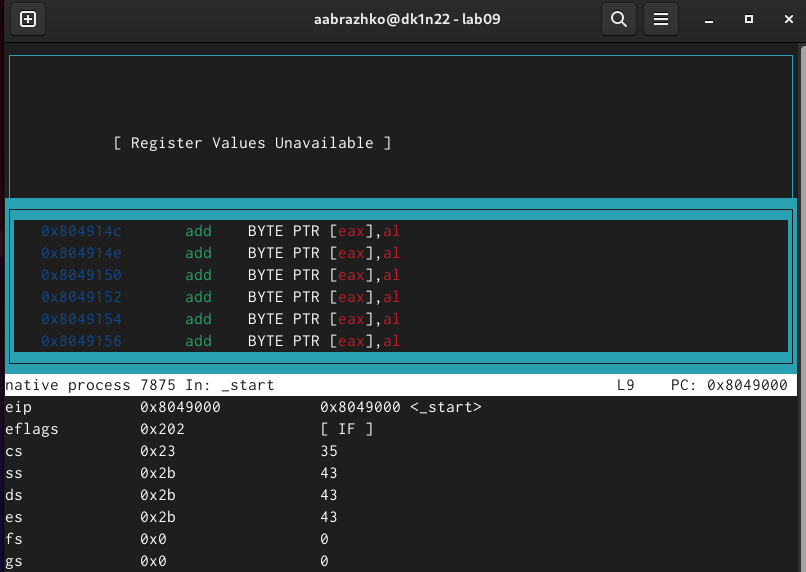


Рис. 15: Просмотр и замена

С помощью команды просмотрим значение переменной msg1 (рис. 16).

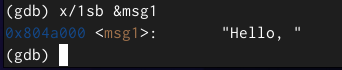


Рис. 16: Просмотр

Просмотрим значение переменной msg2 (рис. 17).

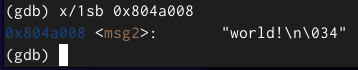


Рис. 17: Просмотр

С помощью команды set изменим значение переменной msg1 (рис. 18).

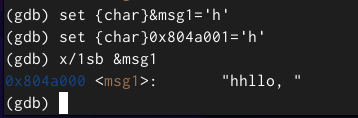


Рис. 18: Изменение значения

С помощью команды set изменим значение переменной msg2 (рис. 19).

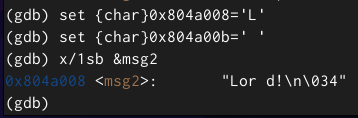


Рис. 19: Изменение значения

Просмотрим значения регистра edx (рис. 20).

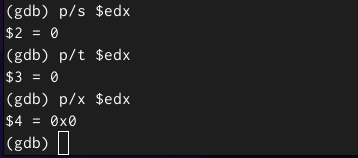


Рис. 20: Просмотр значений

С помощью команды set изменим значения регистра ebx. Команда выводит два разных значения так как в первый раз мы вносим значение 2, а во второй раз регистр равен двум, поэтому и значения разные. (рис. 21).

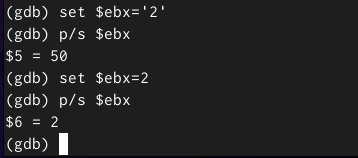


Рис. 21: Изменение значений

Завершаем работу файлов и выходим (рис. 22).

Рис. 22: Завершение и выход

Рис. 22: Завершение и выход

Скопируем файл lab8-2.asm, созданный при выполнении лабораторной работы №8, с программой выводящей на экран аргументы командной строки (Листинг 8.2) в файл с именем lab9-3.asm (рис. 23).

Рис. 23: Копирование

Рис. 23: Копирование

Запускаем файл в отладчике и указываем аргументы (рис. 24).

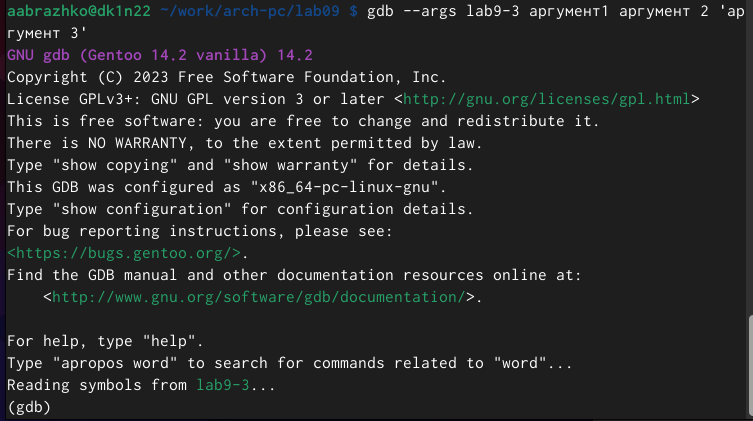


Рис. 24: Запуск

Установим точку останова перед первой инструкцией в программе и запустим её (рис. 25).

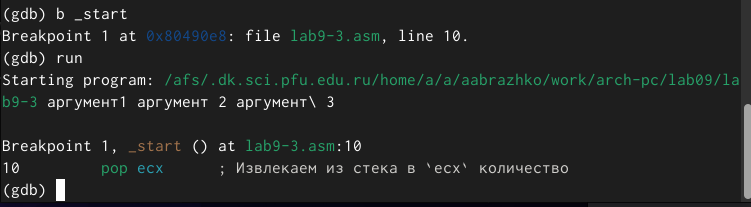


Рис. 25: Установка и запуск

Проверим адрес вершины стека (рис. 26).

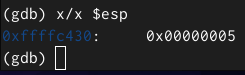


Рис. 26: Проверка

Просмотрим все позиции стека. По первому адрему хранится адрес, в остальных адресах хранятся элементы. Элементы расположены с интервалом в 4 единицы, так как стек может хранить до 4 байт, и для того чтобы данные сохранялись нормально и без помех, компьютер использует новый стек для новой информации. (рис. 27).

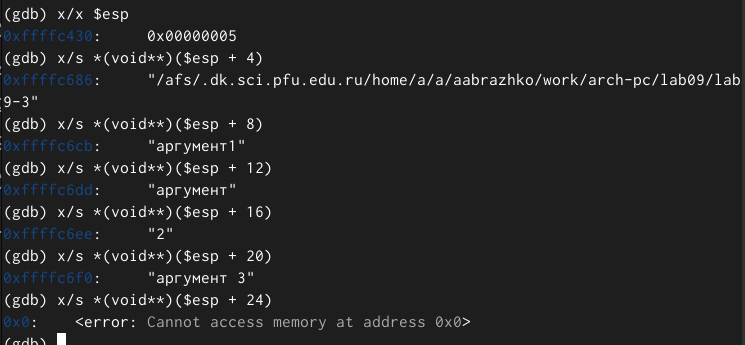


Рис. 27: Просмотр позиций стека

# 5 Выполнение самостоятельной работы

Преобразуем программу из лабораторной работы №8 (Задание №1 для самостоятельной работы), реализовав вычисление значения функции 𝑓(𝑥) как подпрограмму. (рис. 28).

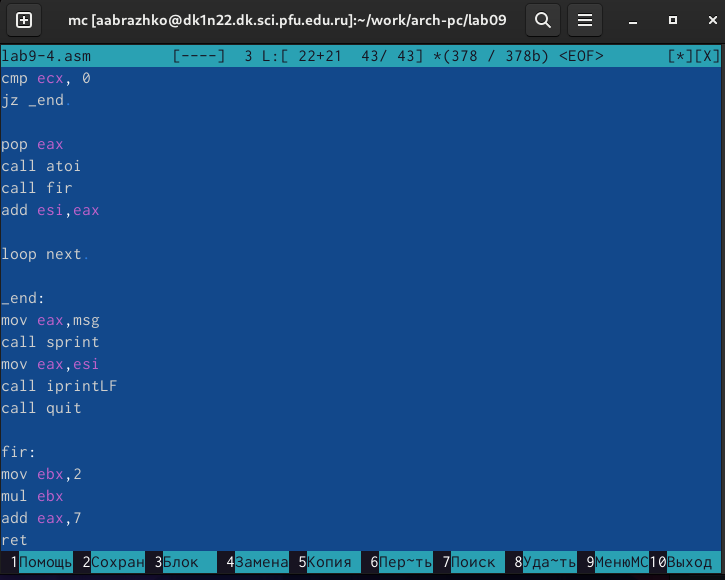


Рис. 28: Программа

Создаём исполняемый файл и проверяем его работу (рис. 29).

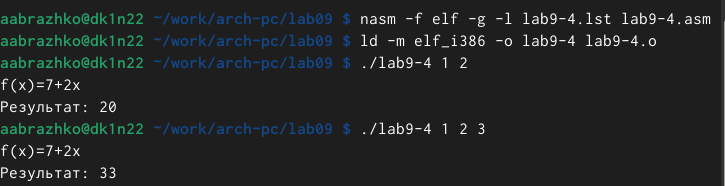


Рис. 29: Создание и проверка

Переписываем программу из листинга 9.3 (рис. 30).

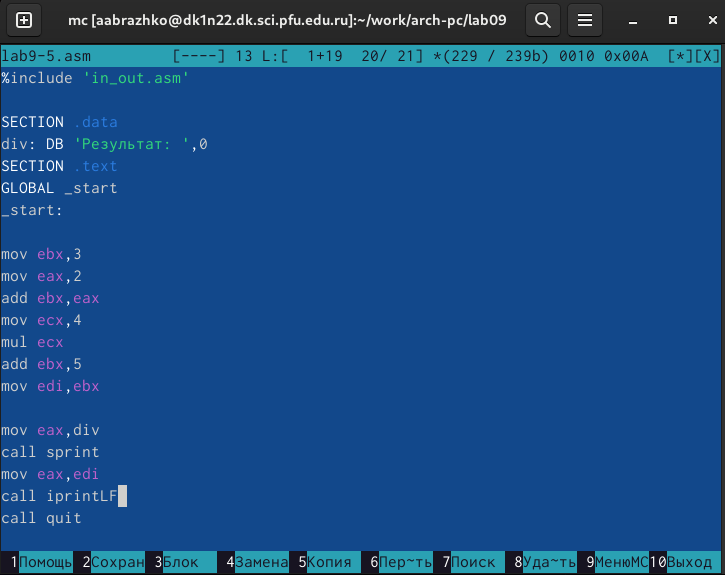


Рис. 30: Программа

Создаём исполняемый файл и проверяем его работу. Программа должна вывести 10, что является ошибкой, так как должно быть 25 (рис. 31).

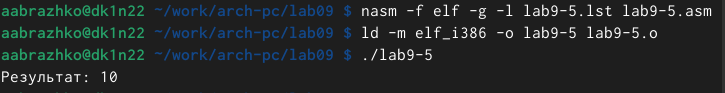


Рис. 31: Создание и проверка

# 6 Выводы

Я приобрела навыки написания программ с использованием подпрограмм. Познакомилась с методами отладки при помощи GDB и его основными возможностями.

# Список литературы

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.

2. Robbins A. Bash Pocket Reference. O’Reilly Media, 2016. 156 с.

3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.

4. Newham C. [Learning the bash Shell: Unix Shell Programming](http://www.amazon.com/Learning-bash-Shell-Programming-Nutshell/dp/0596009658). O’Reilly Media, 2005. 354 с.