Отчёт по лабораторной работе №9

дисциплина: Архитектура компьютера

Веретенников Дмитрий Олегович

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 программ с помощью GDB
3. Самостоятельное выполнение заданий по материалам лабораторной работы

# 3 Теоретическое введение

Отладка — это процесс поиска и исправления ошибок в программе. В общем случае его можно разделить на четыре этапа:

• обнаружение ошибки; • поиск её местонахождения; • определение причины ошибки; • исправление ошибки.

Можно выделить следующие типы ошибок:

• синтаксические ошибки — обнаруживаются во время трансляции исходного кода и вызваны нарушением ожидаемой формы или структуры языка; • семантические ошибки — являются логическими и приводят к тому, что программа запускается, отрабатывает, но не даёт желаемого результата; • ошибки в процессе выполнения — не обнаруживаются при трансляции и вызывают пре- рывание выполнения программы (например, это ошибки, связанные с переполнением или делением на ноль).

Второй этап — поиск местонахождения ошибки. Некоторые ошибки обнаружить доволь- но трудно. Лучший способ найти место в программе, где находится ошибка, это разбить программу на части и произвести их отладку отдельно друг от друга.

Третий этап — выяснение причины ошибки. После определения местонахождения ошибки обычно проще определить причину неправильной работы программы. Последний этап — исправление ошибки. После этого при повторном запуске программы, может обнаружиться следующая ошибка, и процесс отладки начнётся заново.

# 4 Выполнение лабораторной работы

## 4.1 Релазиация подпрограмм в NASM

Ввожу в файл lab9-1.asm программу из листинга 9.1, сохдаю исполняемый файл и проверяю работу программы (рис. 1).

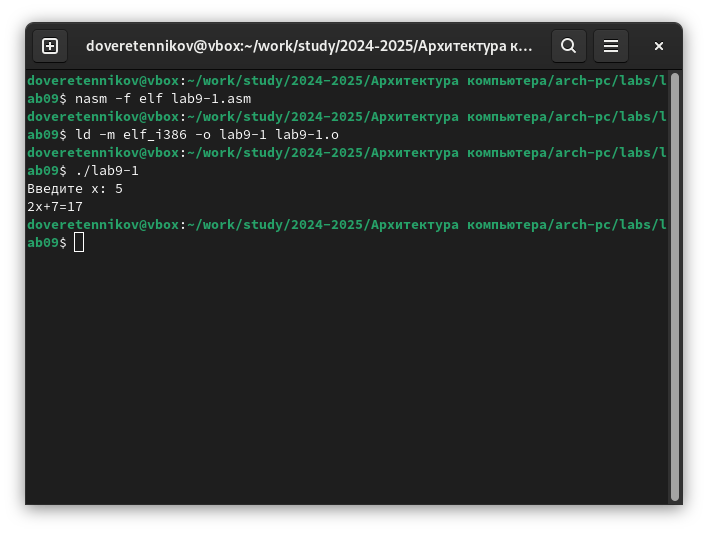


Рис. 1: Создание исполняемого файла и запуск программы

Изменяю текст программы, добавив подпрограмму для вычесления выражения f(g(x)) (рис. 2).



Рис. 2: Изменение программы

Запускаю измененную программу и проверяю правильность ее работы (рис. 3).

|  |
| --- |
| Запуск программы |

Рис. 3: Запуск программы

### 4.1.1 Отладка программ с помощью GDB

Создаю файл lab9-2.asm ввожу в него текст из листинга 9.2, получаю исполняемый файл и загружаю файл в отладчик gdb (рис. 4).

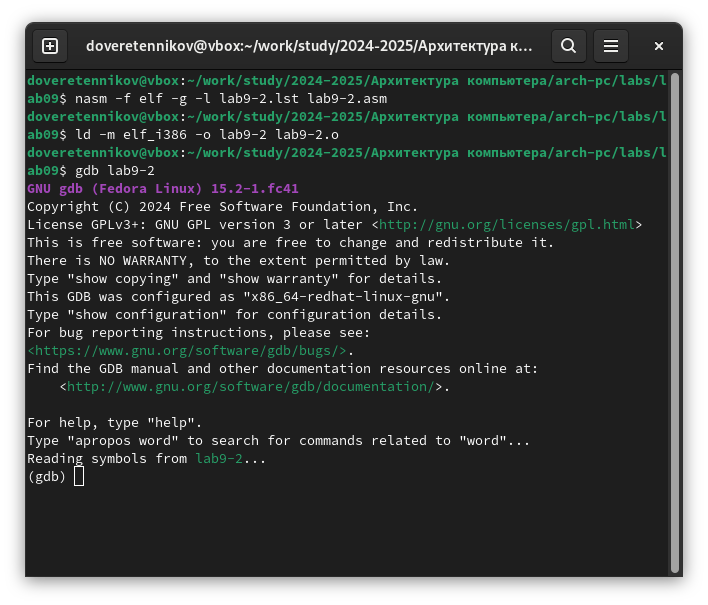


Рис. 4: Получение исполняемого файл и загрузка его в отладчик gdb

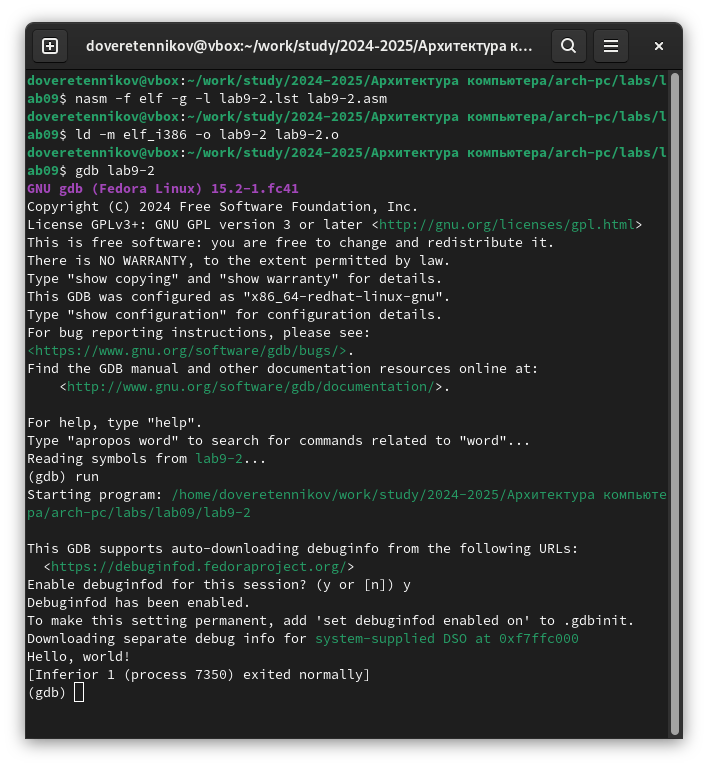


Рис. 5: Отладчик gdb

Устанавливаю брейкпоинт на метку \_start (рис. 5).

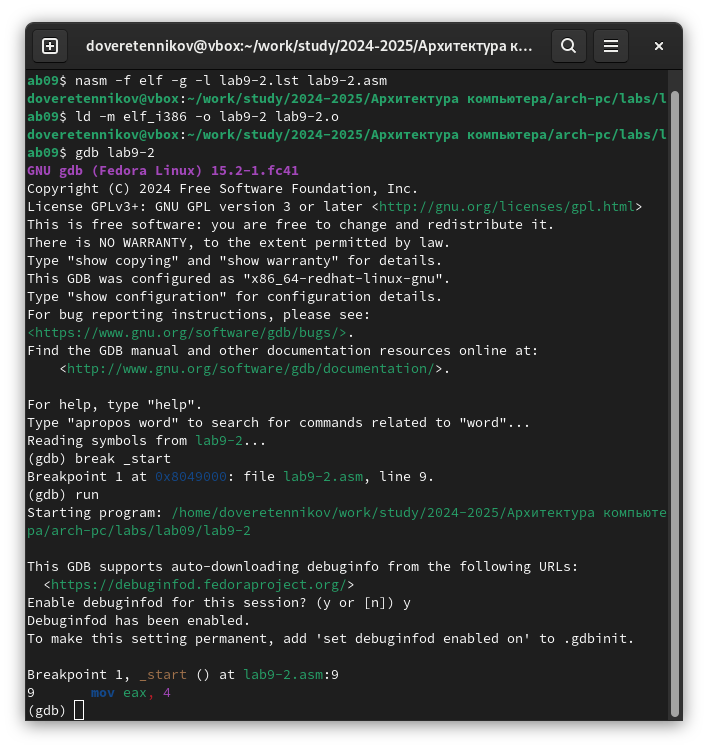


Рис. 6: Установка брейкпоинта

Просматриваю дисассимилированный код программы (рис. 6).

!Дисассимилированный код программы](/home/doveretennikov/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab09/Report/image/Вставленное изображение (7).png){#fig:007}

Переключаюсь на отображение команд с Intel’овским синтаксисом (рис. 7).

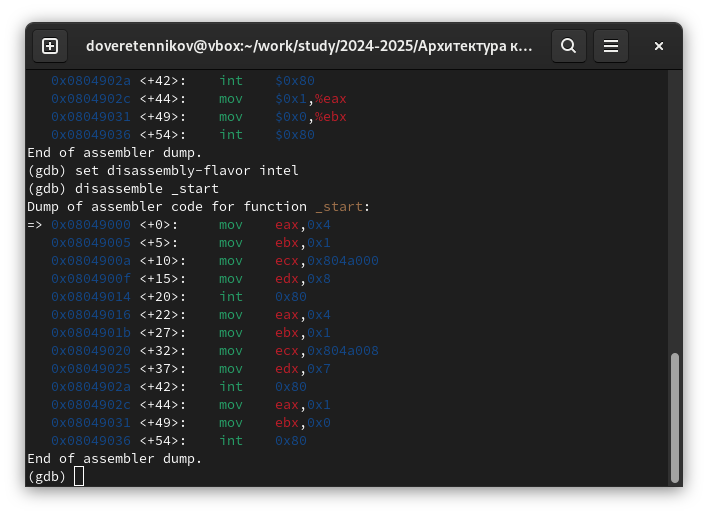


Рис. 7: Переключение

Различия между синтаксисом ATT и Intel заключаются в порядке операндов (ATT - Операнд источника указан первым. Intel - Операнд назначения указан первым), их размере (ATT - pазмер операндов указывается явно с помощью суффиксов, непосредственные операнды предваряются символом $; Intel - Размер операндов неявно определяется контекстом, как ax, eax, непосредственные операнды пишутся напрямую), именах регистров(ATT - имена регистров предваряются символом %, Intel - имена регистров пишутся без префиксов).

Включаю режим псевдографики (рис. 8).

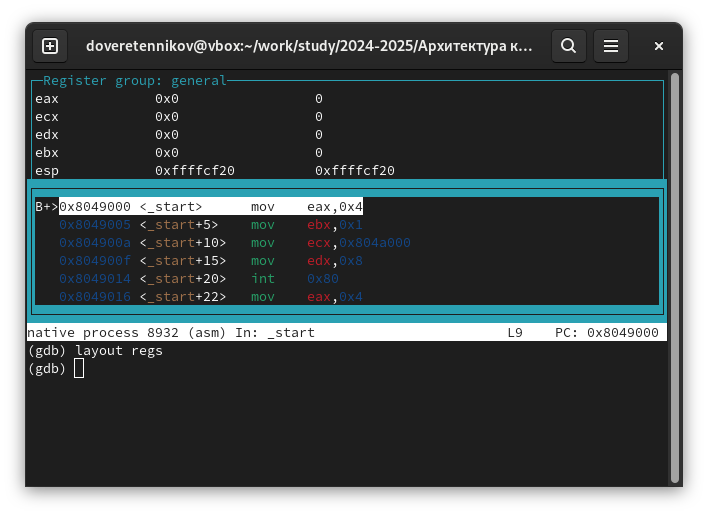


Рис. 8: Режим псевдографики

### 4.1.2 Добавление точек останова

Устанавливаю еще одну точку останова по адресу интрукции и просматриваю информацию о всех установленных точках останова (рис. 9).

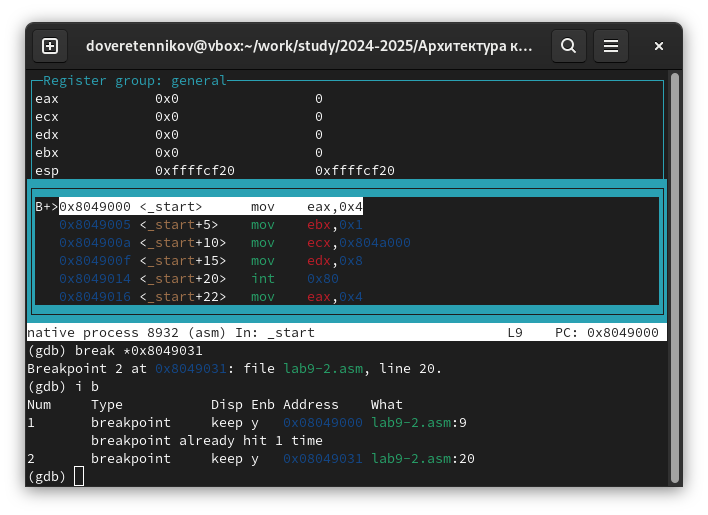


Рис. 9: Информация о точках останова

### 4.1.3 Работа с данными программы в GDB

Просматриваю содержимое регистров (рис. 10).

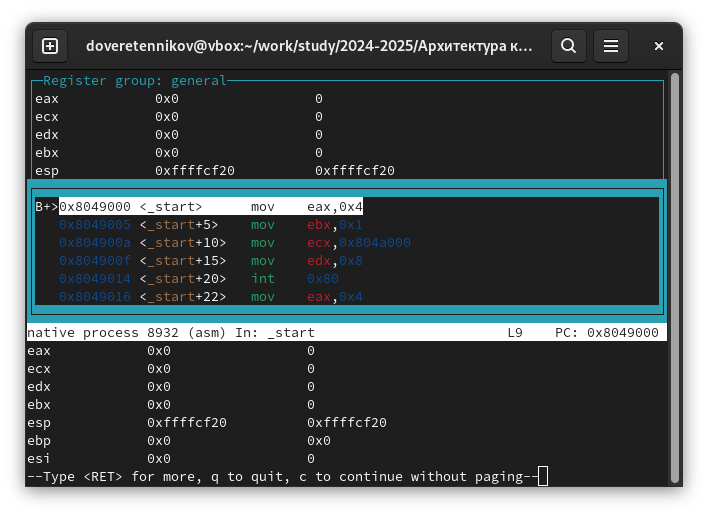


Рис. 10: Содержание регистров

Просматриваю значение переменной msg1 по имени и переменной msg2 по адресу (рис. 11).

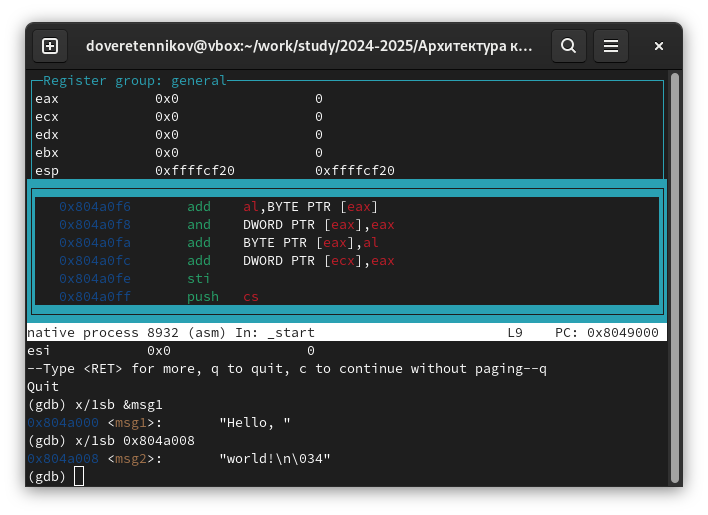


Рис. 11: Значения переменных

Изменяю первый символ переменной msg1 и переменной msg2 (рис. 12).

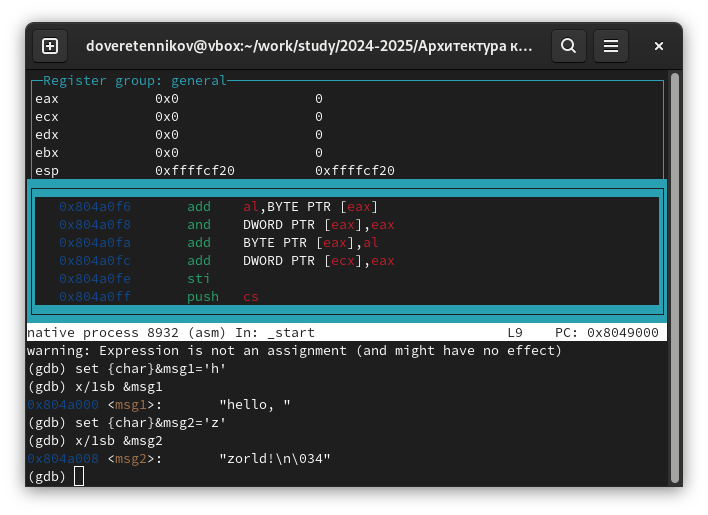


Рис. 12: Изменения первых симолов переменных

Вывожу в различных форматах значение регистра ebx (рис. 13).

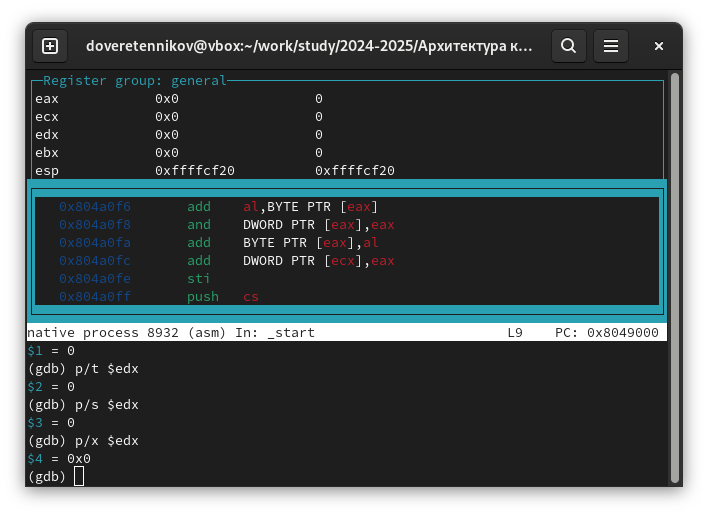


Рис. 13: Вывод ebx в разных форматах

С помощью команды set изменяю значение регистра ebx (рис. 14).

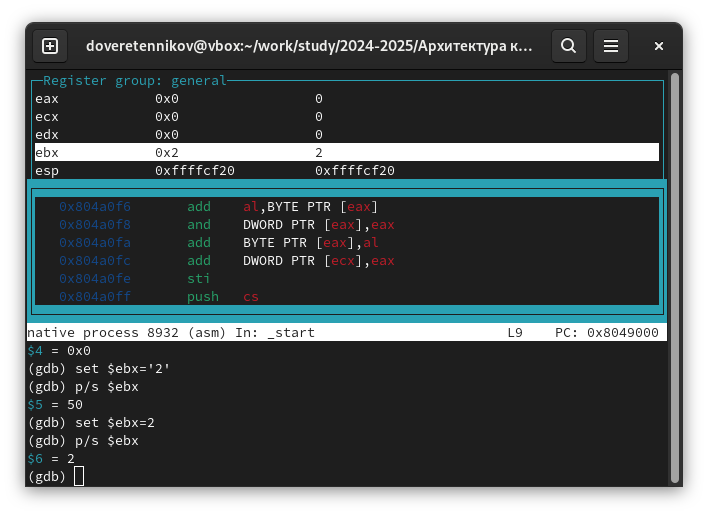


Рис. 14: Изменение значения регистра ebx

### 4.1.4 Обработка аргументов командной строки в GDB

Копирую файл lab8-2.asm в файл с именем lab9-3.asm и создаю исполняемый файл (рис. 15).

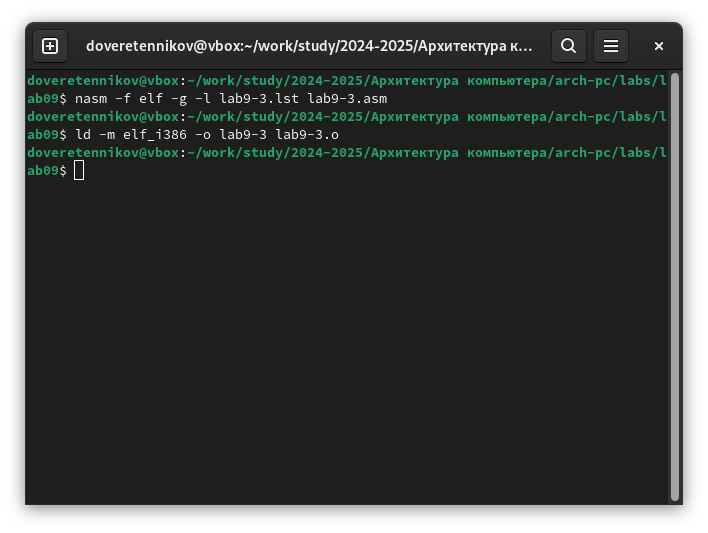


Рис. 15: Создание исполняемого файла

Загружая исполняемый файл в отладчик указываю аргументы, далее устанавливаю точку останова перед первой инструкцией и затем просматриваю позиции стека (рис. 16).

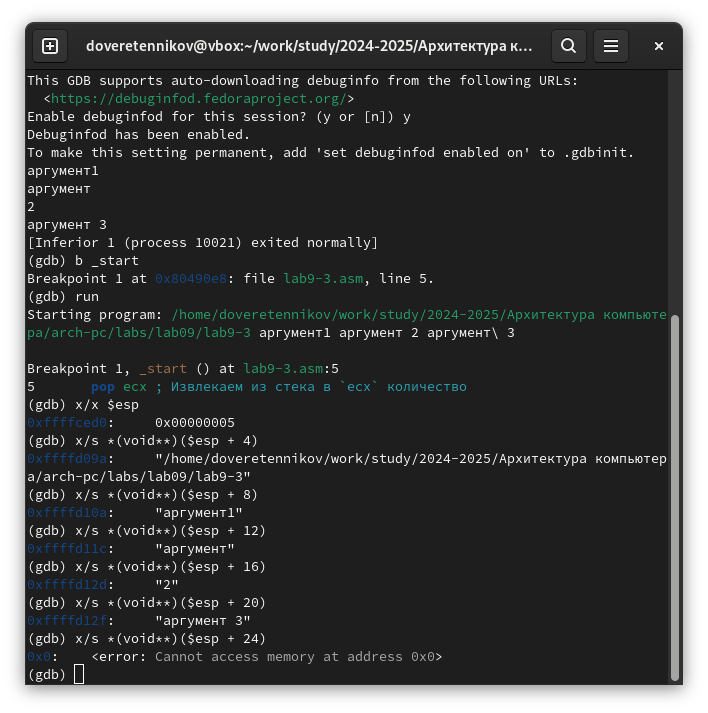


Рис. 16: Загрузка файла, установка точки останова и просмотр позиции стека

## 4.2 Задание для самостоятельной работы

Изменяю программу из 8 лабораторной работы чтобы она вычисляла значения как подпрограмма (рис. 17).

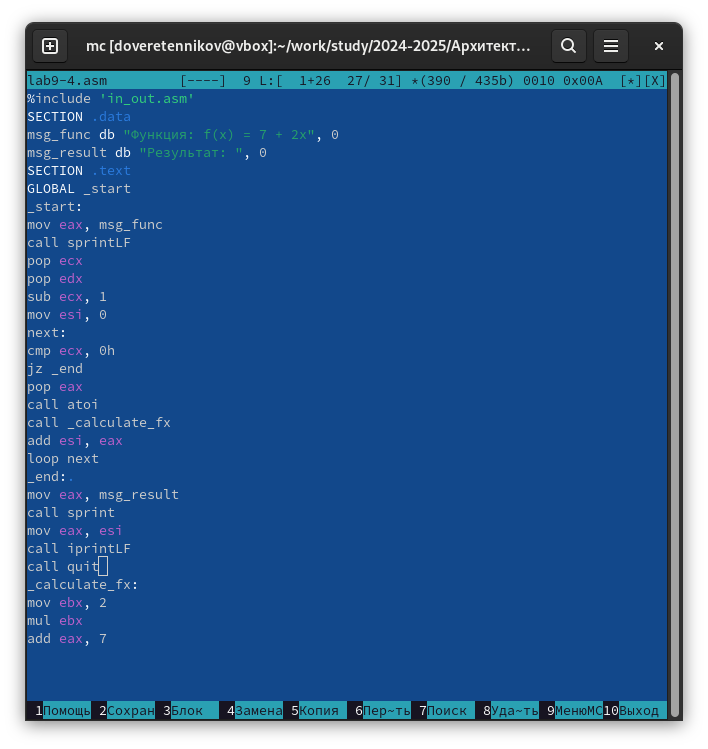


Рис. 17: Измененная программа

# 5 Выводы

Почле выполнения данной лабораторной работы я приобрел навыки написания программ с использованием подпрограмм, а так же познакомился с методами отладки при поомщи GDB и его основными возможностями.

# Список литературы

1. GDB: The GNU Project Debugger. — URL: https://www.gnu.org/software/gdb/.
2. GNU Bash Manual. — 2016. — URL: https://www.gnu.org/software/bash/manual/.
3. Midnight Commander Development Center. — 2021. — URL: https://midnight-commander. org/.
4. NASM Assembly Language Tutorials. — 2021. — URL: https://asmtutor.com/.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O’Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: http://www.amazon.com/Learning- bash-Shell-Programming-Nutshell/dp/0596009658.
6. Robbins A. Bash Pocket Reference. — O’Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: https://www.nasm.us/docs.php.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс,
11. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM. — 2021. — URL: https://www.opennet.ru/docs/RUS/nasm/.
13. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВ- Петербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
14. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: http://www.stolyarov.info/books/asm\_unix.
15. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
16. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер,
17. — 1120 с. — (Классика Computer Science).