Лабораторная работа №9

Понятие подпрограммы. Отладчик GDB.

Борисенкова София Павловна

Содержание

Список иллюстраций

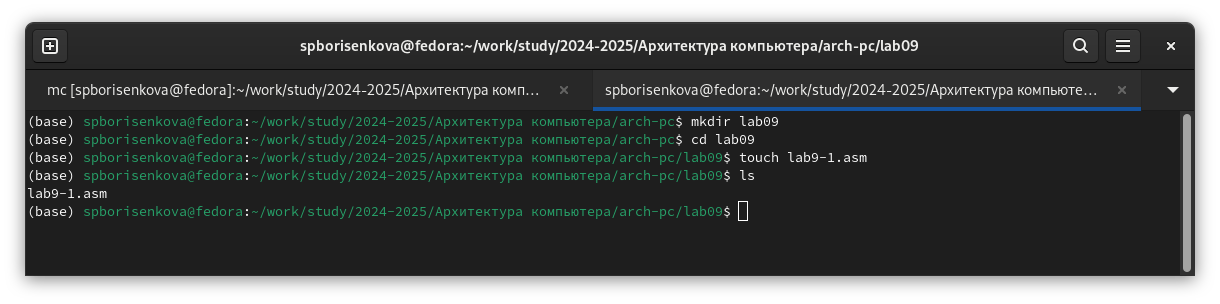
Список таблиц

# 1 Цель работы

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

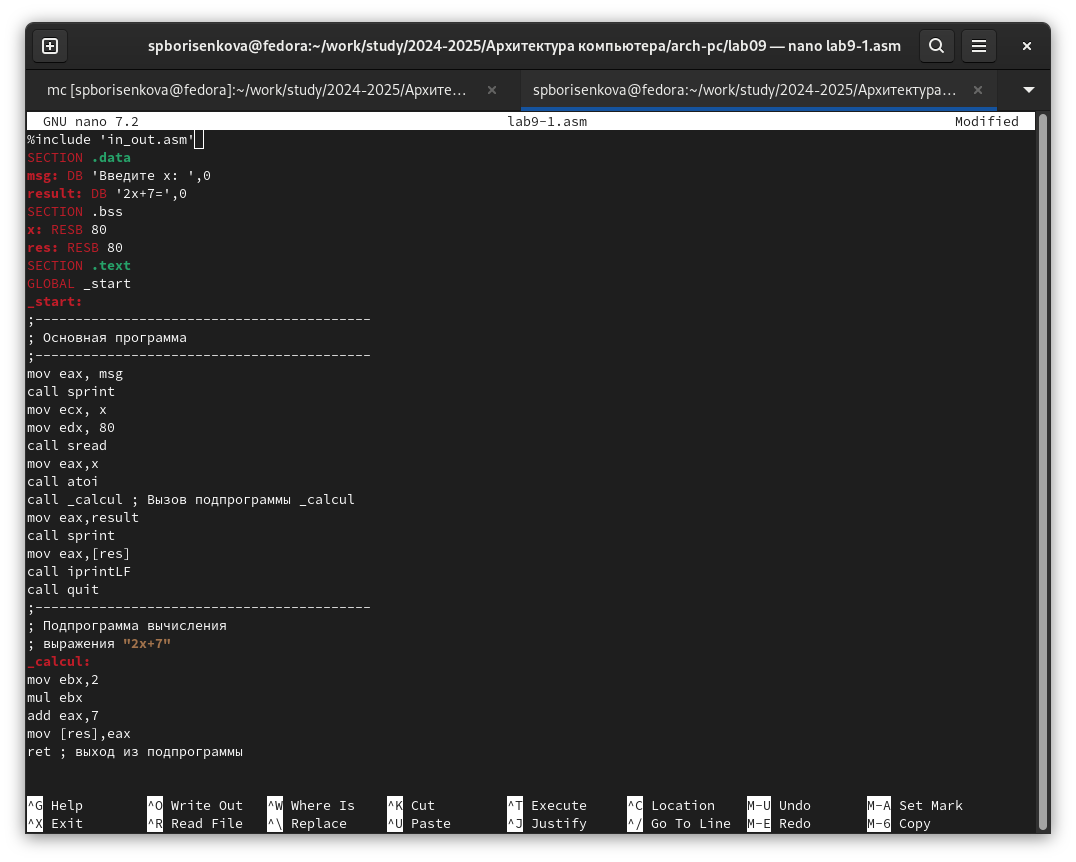
# 2 Выполнение лабораторной работы

Для начала выполнения лабораторной работы создадим рабочую директорию и файл lab9-1.asm (рис. 2.1):



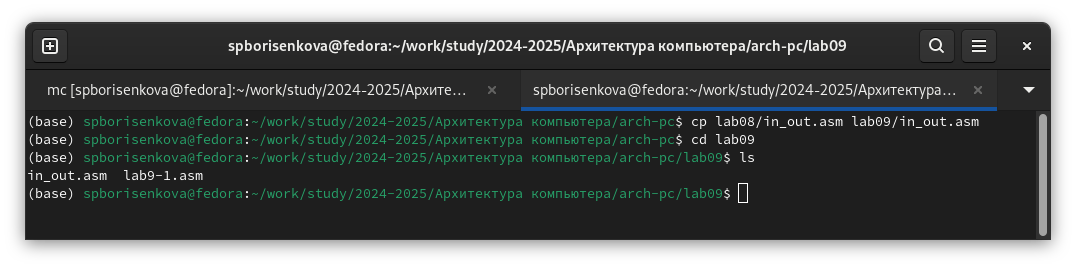
Создание рабочей директории и файла lab9-1.asm

Теперь, вставим в ранее созданный файл из листинга 9.1.



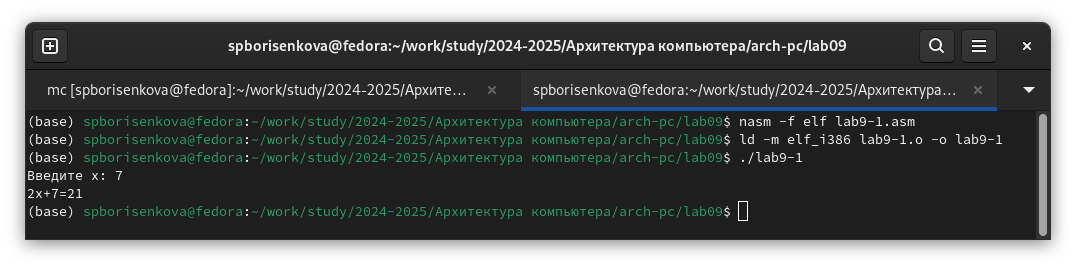
Вставка кода из файла листинга 9.1

Чтобы собрать код, нужен файл in\_out.asm. скопируем его из директории прошлой лабораторной работы (рис. 2.3):



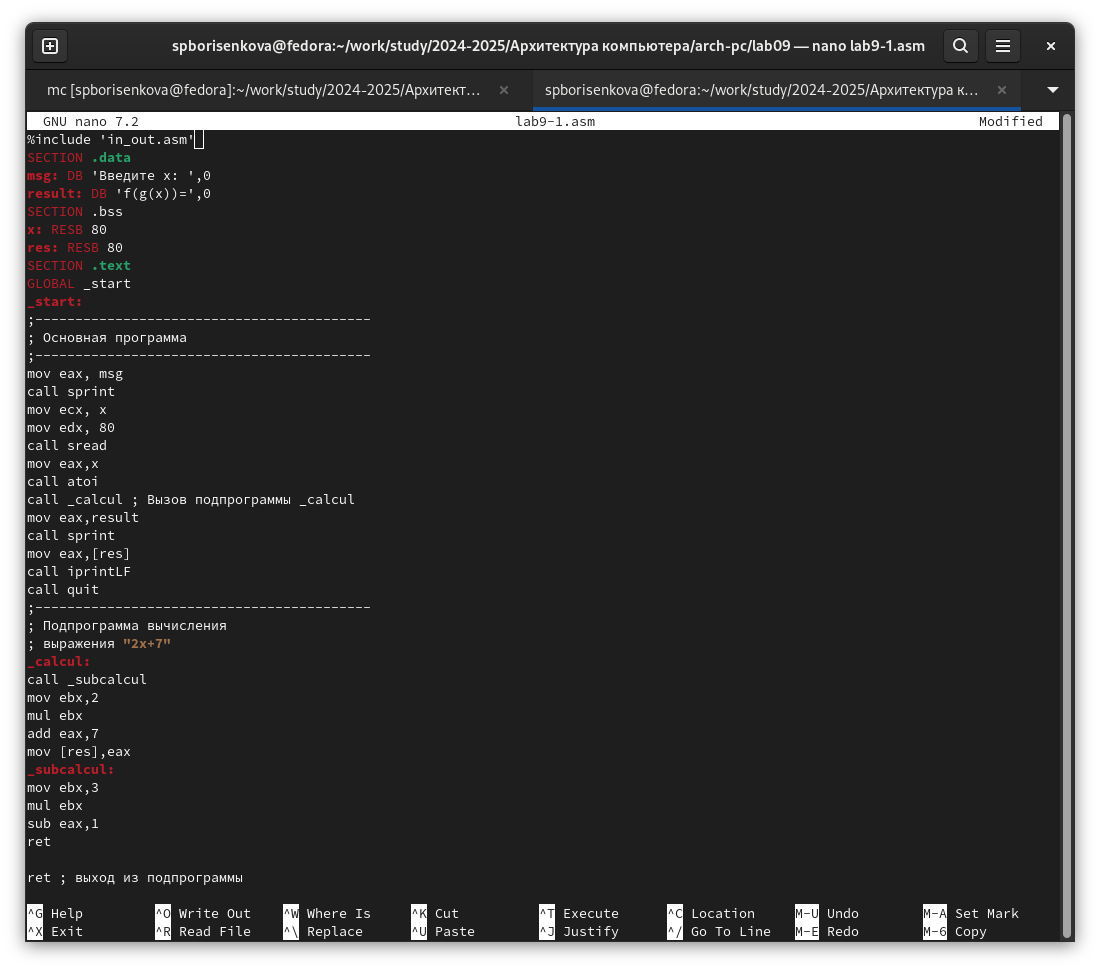
Копирование файла in\_out.asm в рабочую директорию

Теперь соберём программу и посмотрим на результат выполнения. Она работает корректно. Изменим код, добавив подпрограмму subcalcul



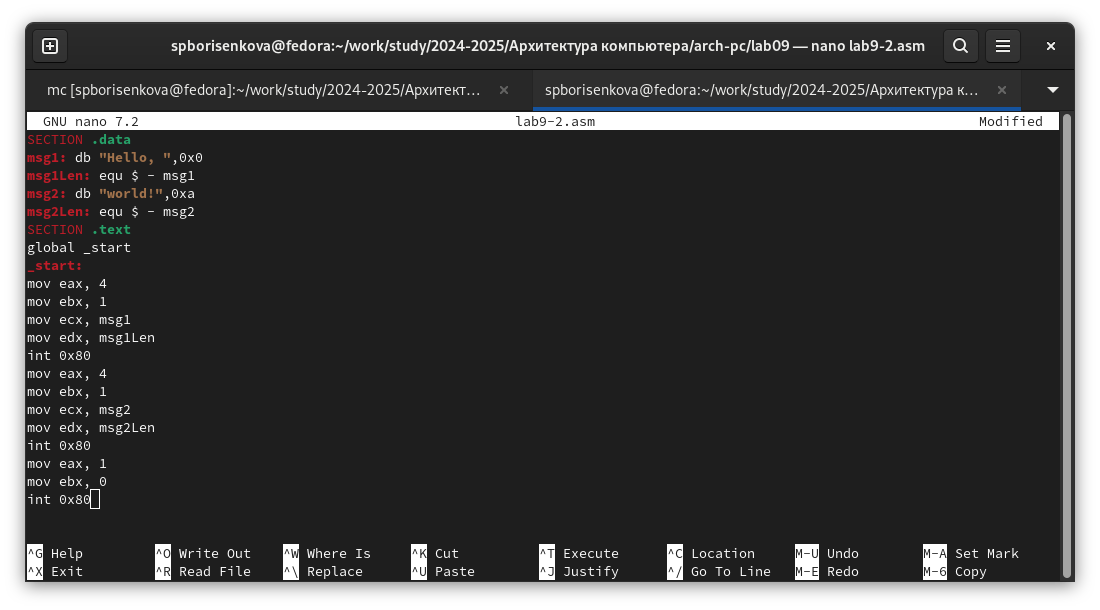
Изменение файла lab9-1.asm

Попробуем собрать программу и запустить её (рис. 2.5):



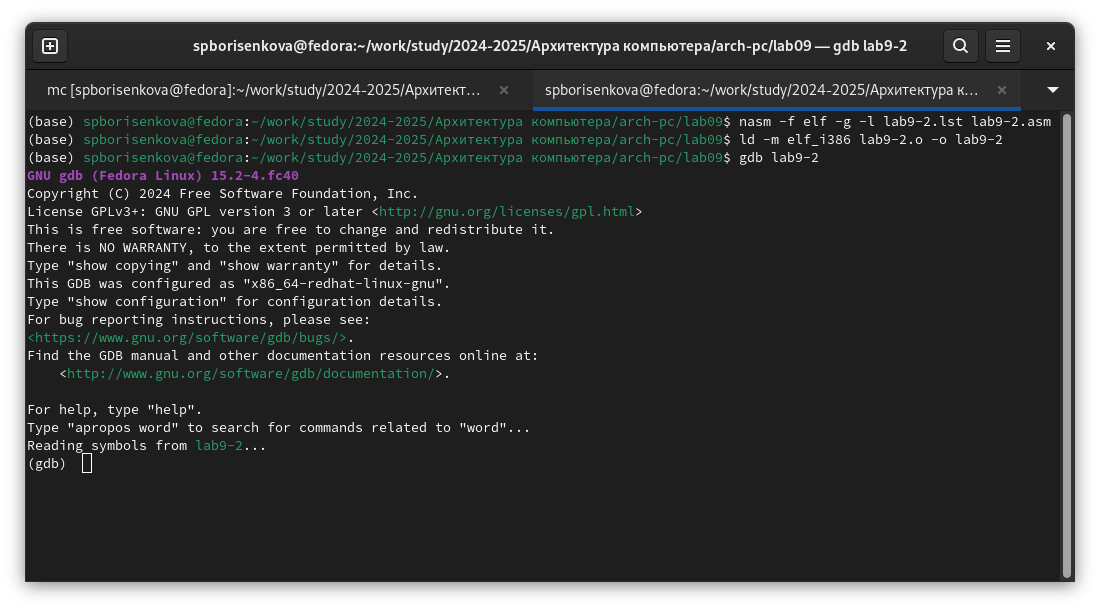
Повторная сборка программы из файла lab9-1.asm и её запуск

Как видим, она корректно выводит значение функции. Создадим второй файл и вставим в него код из файла листинга 9.2 (рис. 2.6):



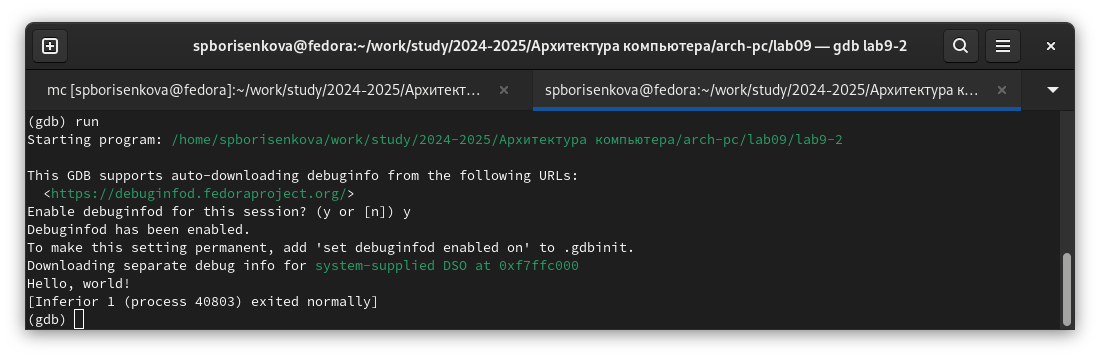
Запись кода из листинга 9.2 в файл lab9-2.asm

Соберём его и откроем исполняемый файл в отладчике (рис. 2.7):



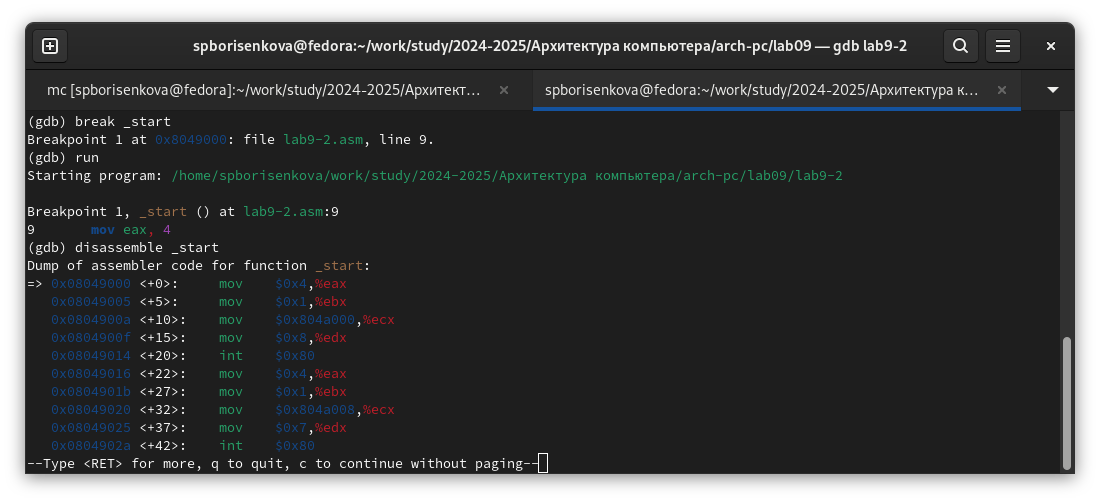
Отладчик gdb

Запустим программу командой run (рис. 2.8):



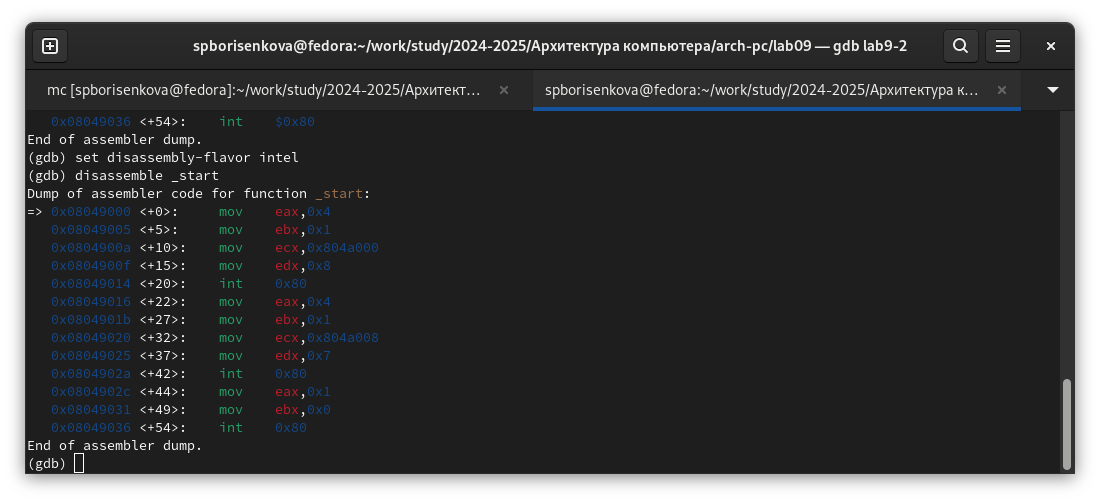
Запуск lab9-1.asm в gdb

Установим брейкпоинт и запустим ещё раз: (рис. 2.9):



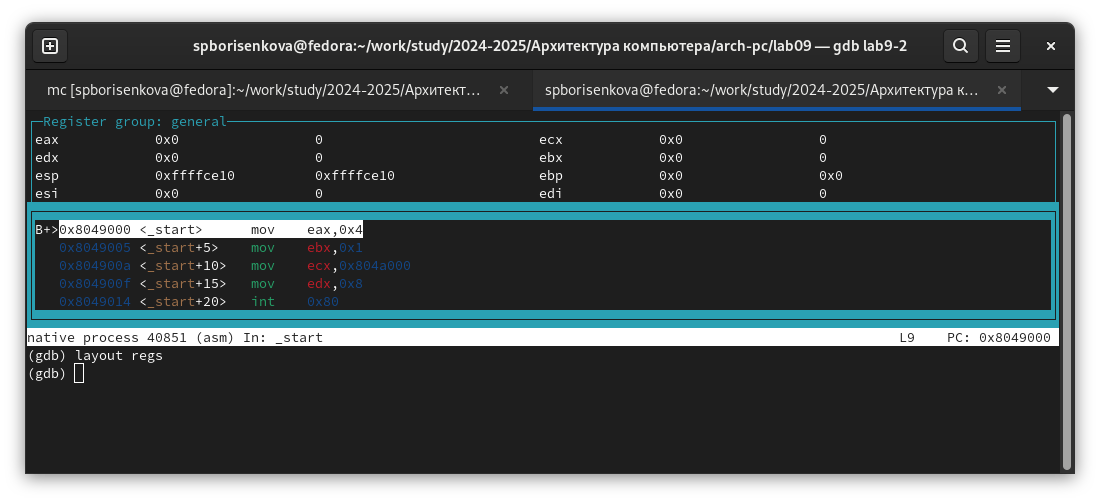
Запуск lab9-1.asm в gdb с точкой останова

Переключимся на отображение команд с синтаксисом intel (рис. 2.10):



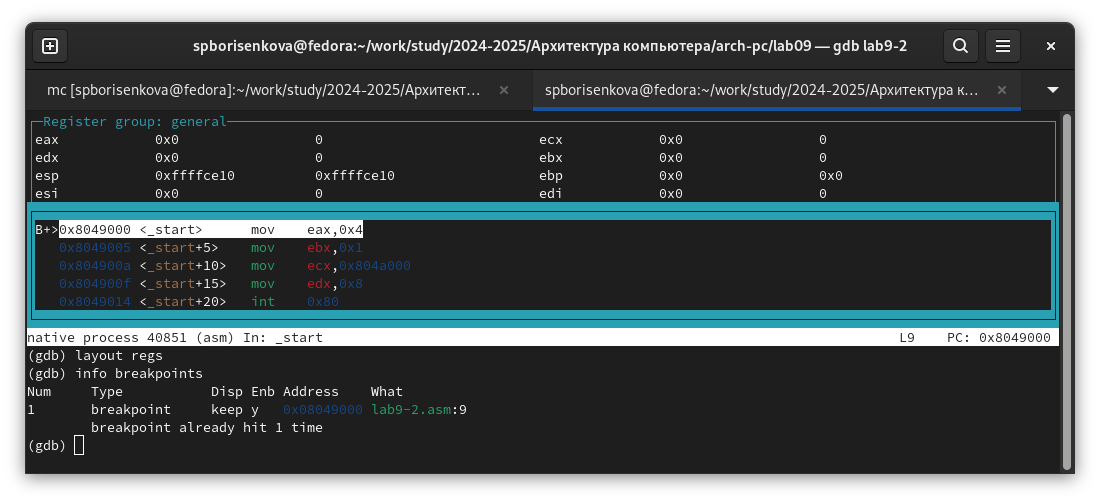
Отображение команд с синтаксисом intel

Главные отличия синтаксиса AT&T и Intel состоят в порядке записи переменных в командах, виде записи числовых констант, и особенностях записиимён регистров. Включим режим псевдографики (рис. 2.11):



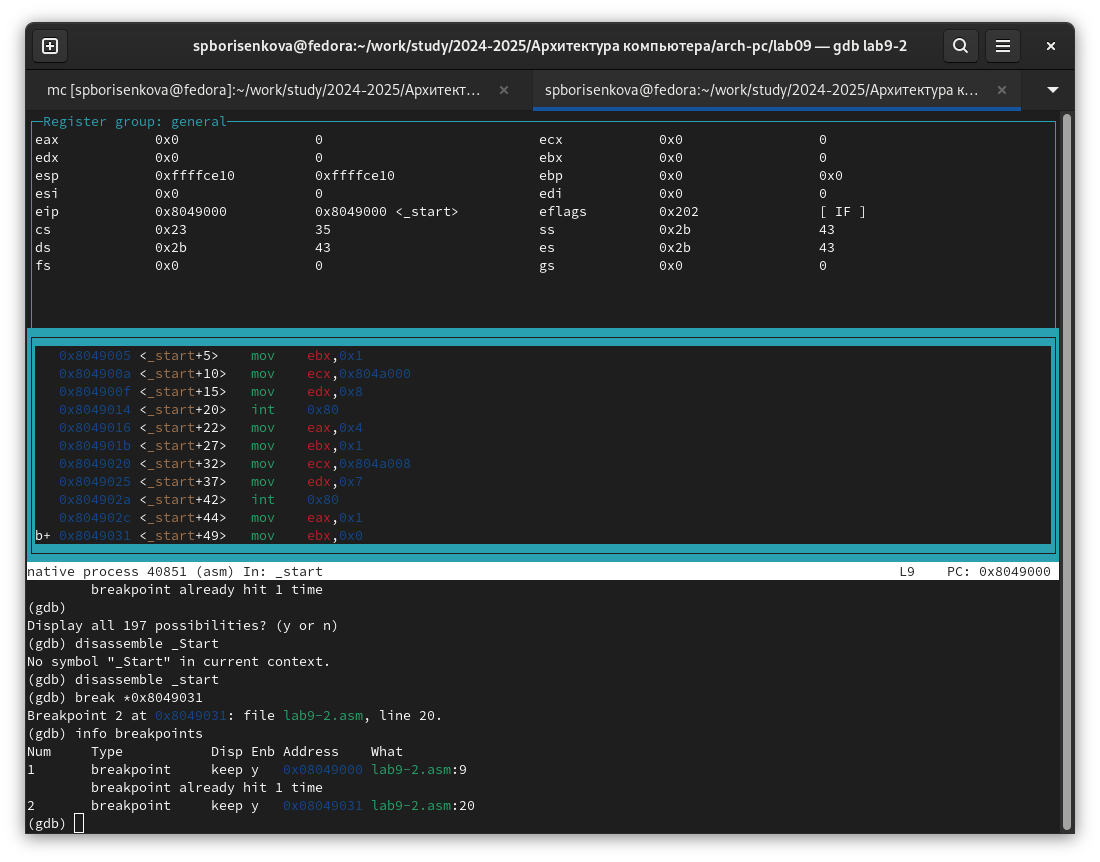
Режим псевдографики

Проверим существующие точки останова командой info breakpoints (рис. 2.12):



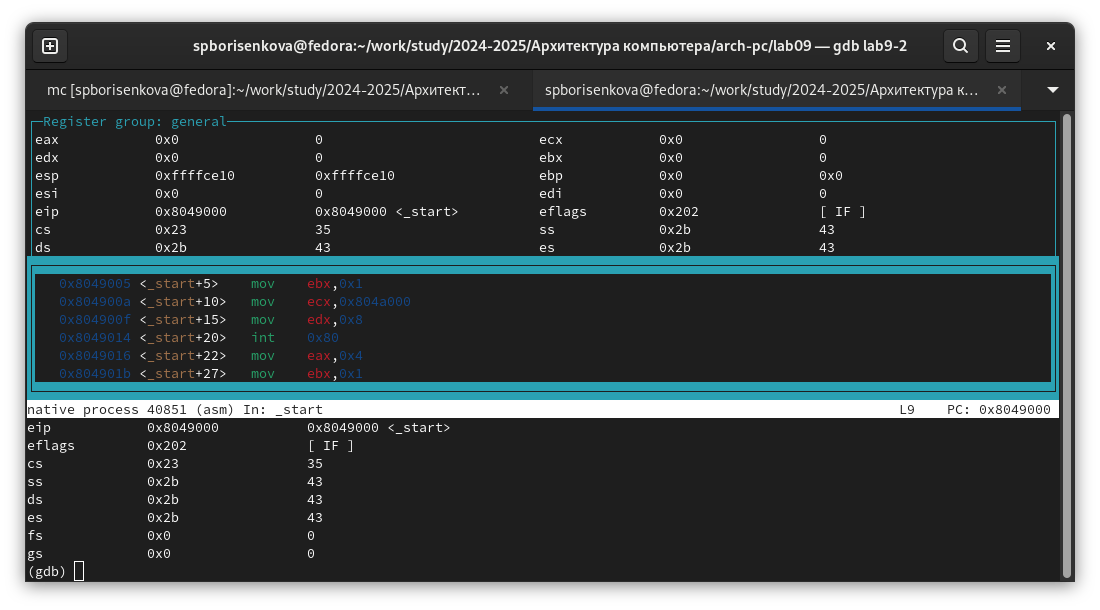
Точки останова

Установим ещё одну по адресу инструкции и опять выведем список точек останова.(рис. 2.13):



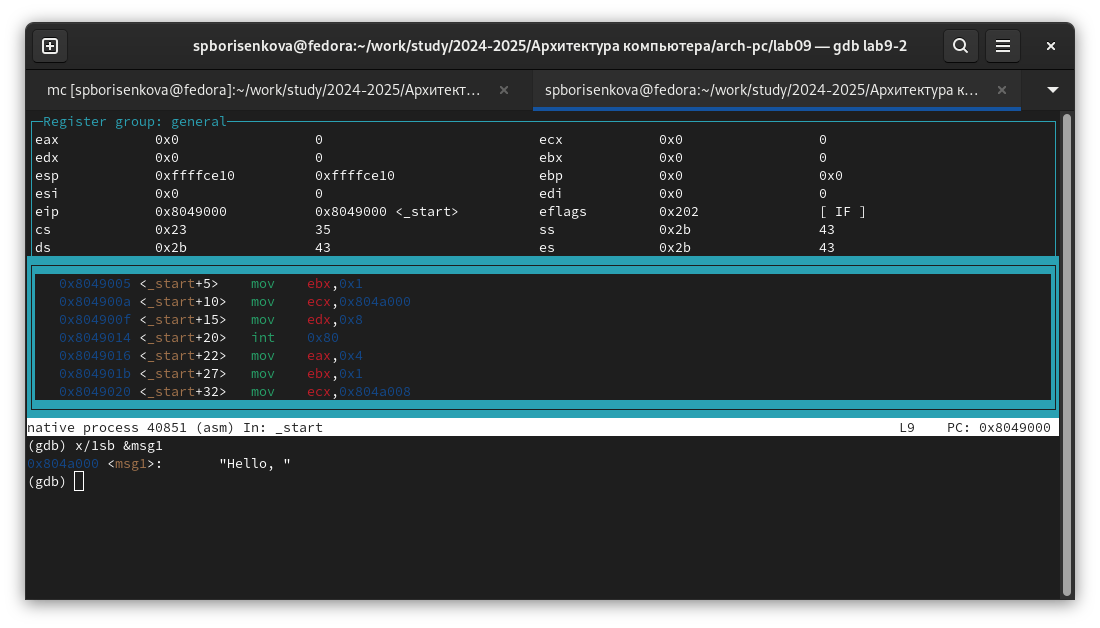
Точки останова 2

Проверим содержание регистров. (рис. 2.14):



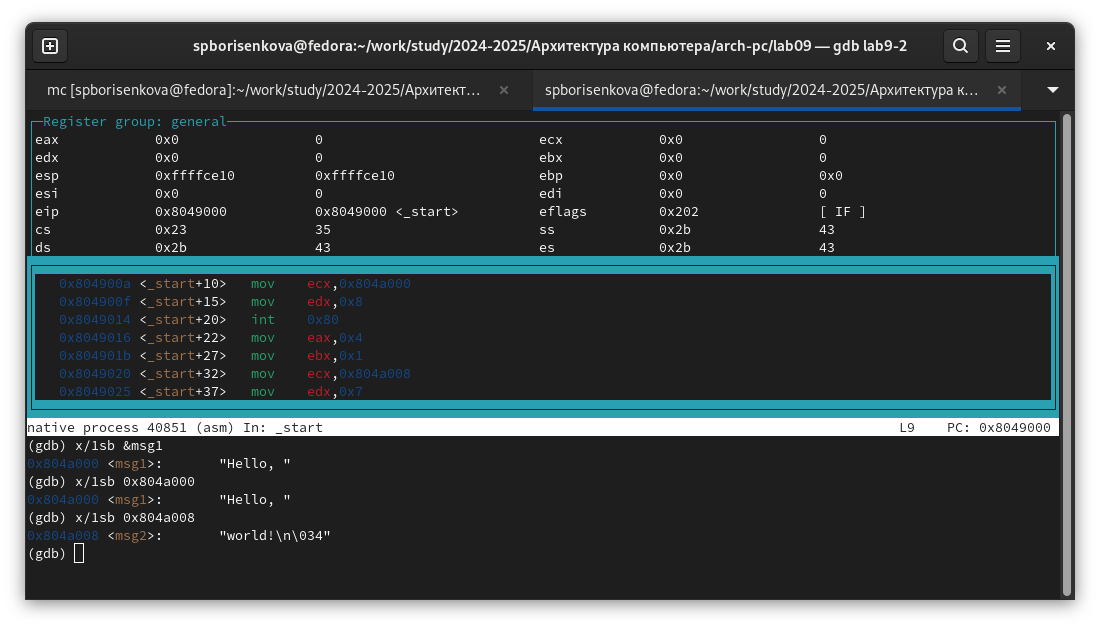
Содержание регистров

Проверим содержание переменной msg1 по имени. (рис. 2.15):



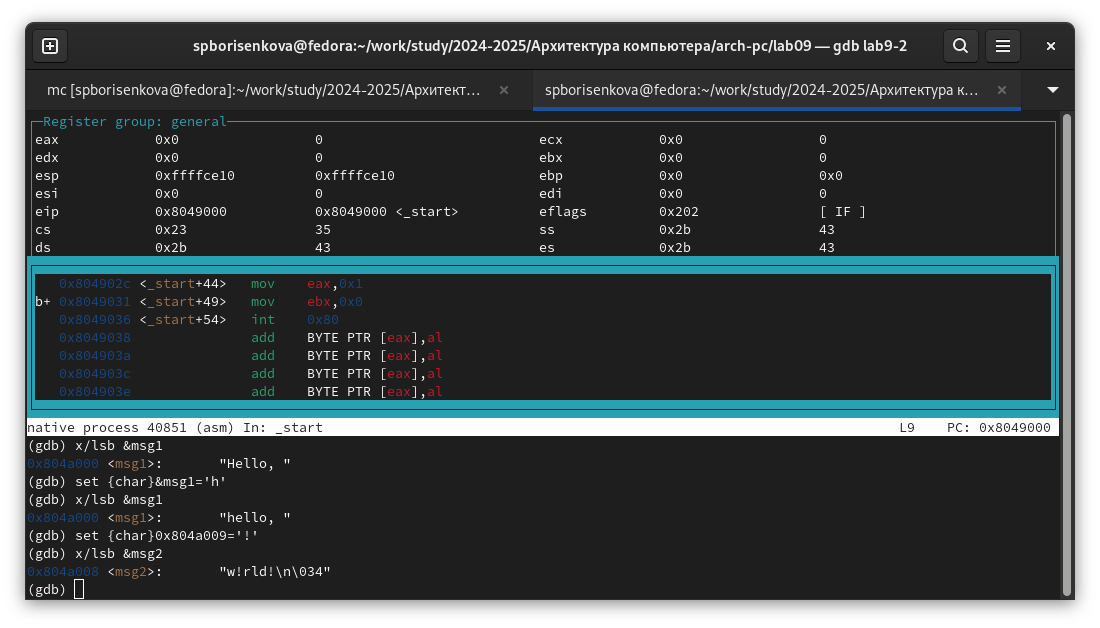
Содержание msg1

Проверим содержание переменной msg2 по адресу. (рис. 2.16):



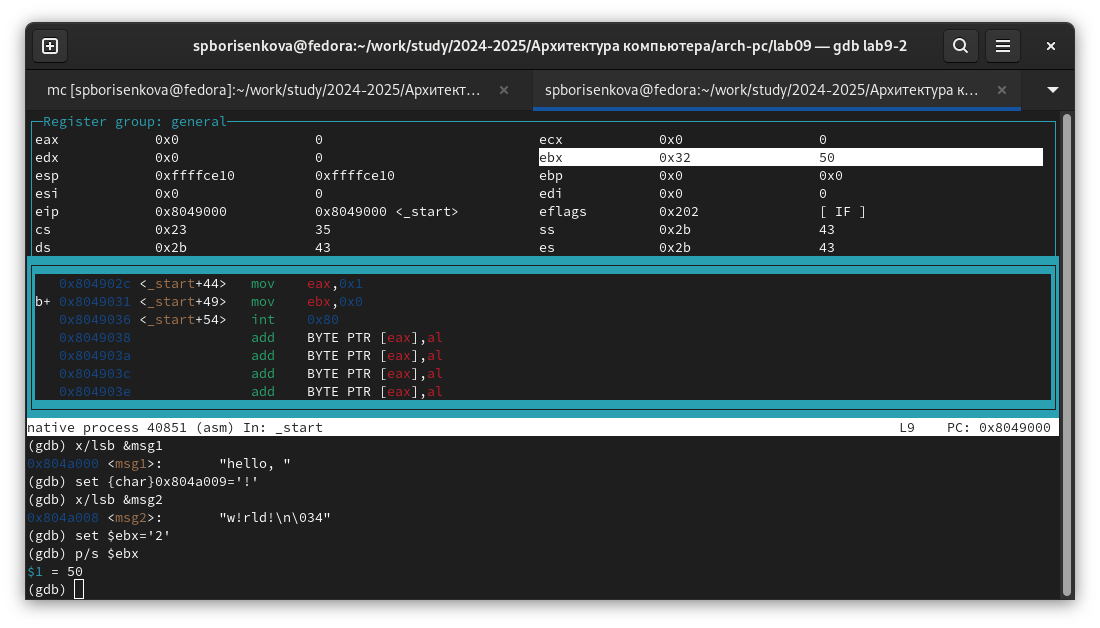
Содержание msg2

Изменим командой set содержание msg1 и msg2. (рис. 2.17):



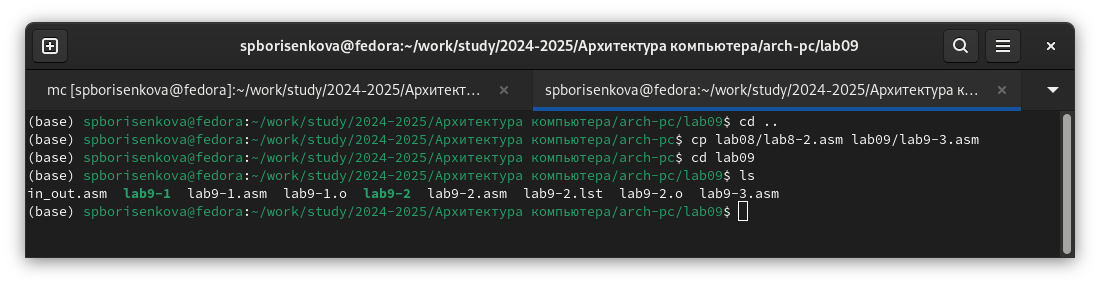
Содержание msg1 и msg2

Изменим командой set содержание ebx. (рис. 2.18):



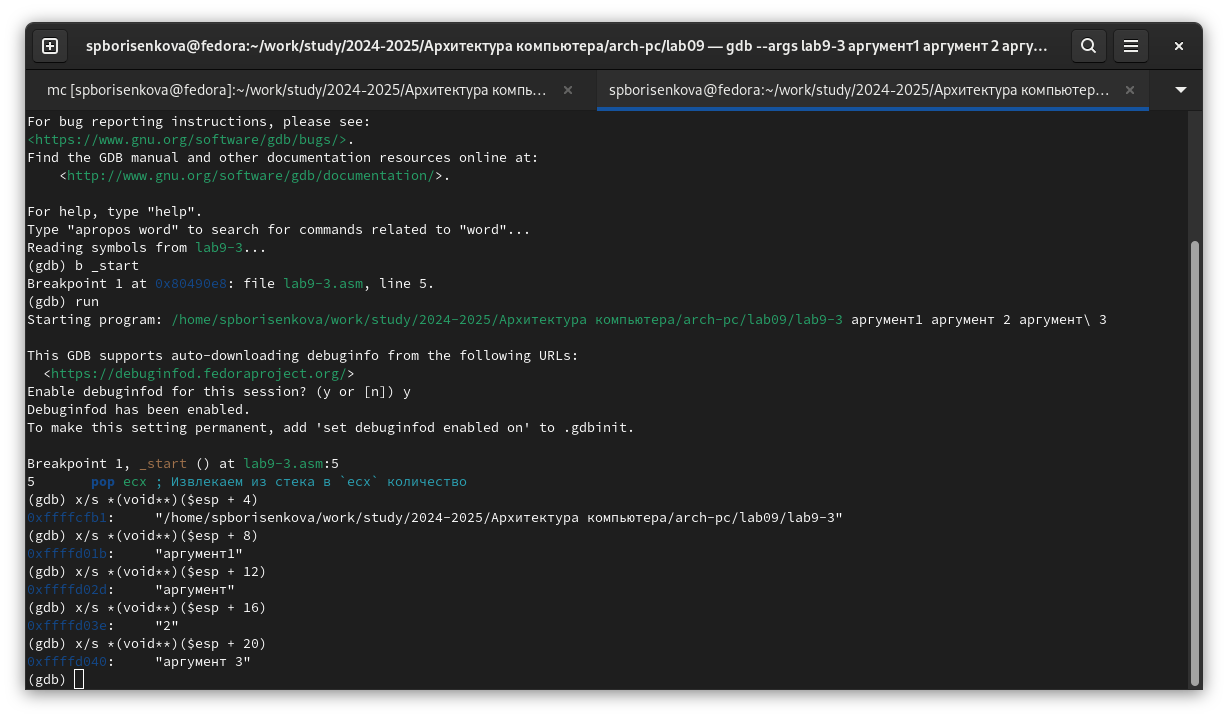
Содержание ebx

Скопируем файл lab8-2.asm в файл с именем lab9-3.asm и создадим исполняемый файл(рис. 2.19):



Копирование файла

Загрузим файл в gdb и исследуем расположение аргументов командной строки в стеке после запуска программы (рис. 2.20):

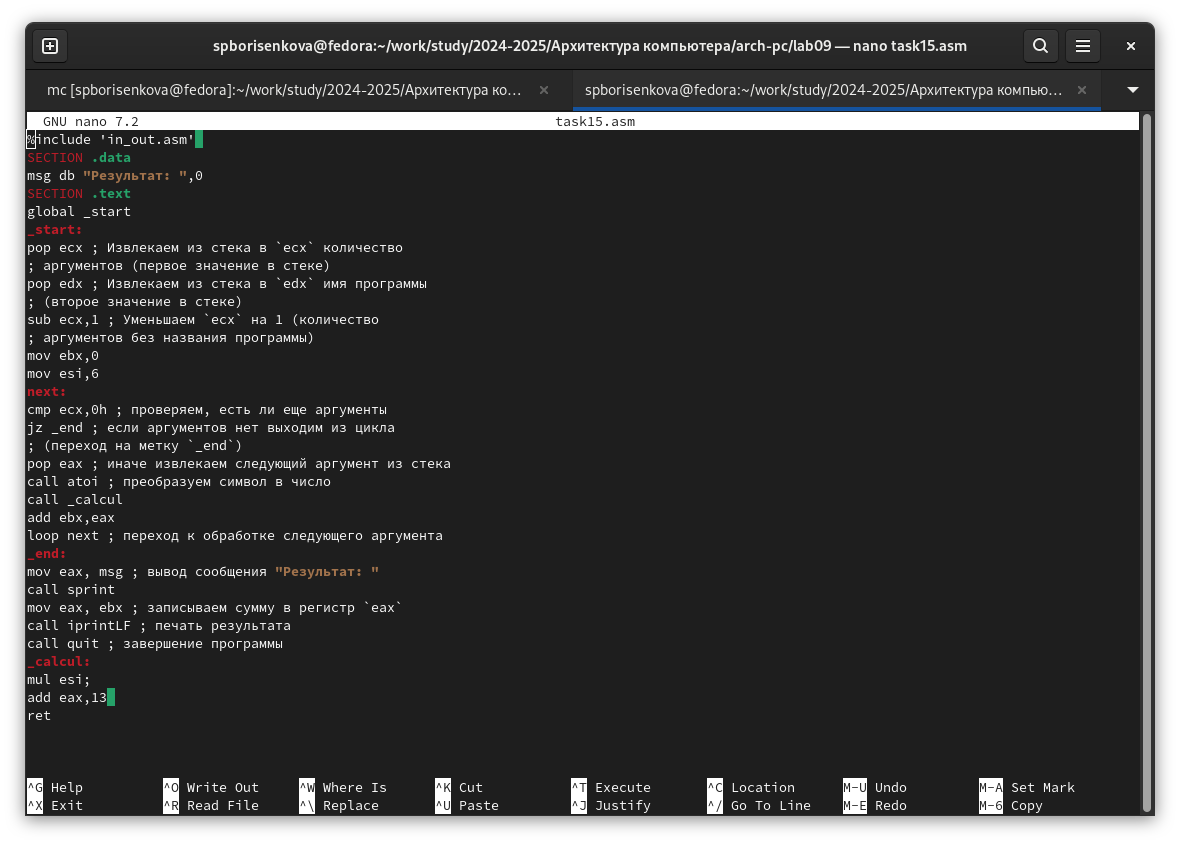


Исследование стека

Как видим, для вывода каждого элемента стека нам нужно менять значение адреса с шагом 4. Это связано с тем, что под каждый элемент выделяется 4 байта

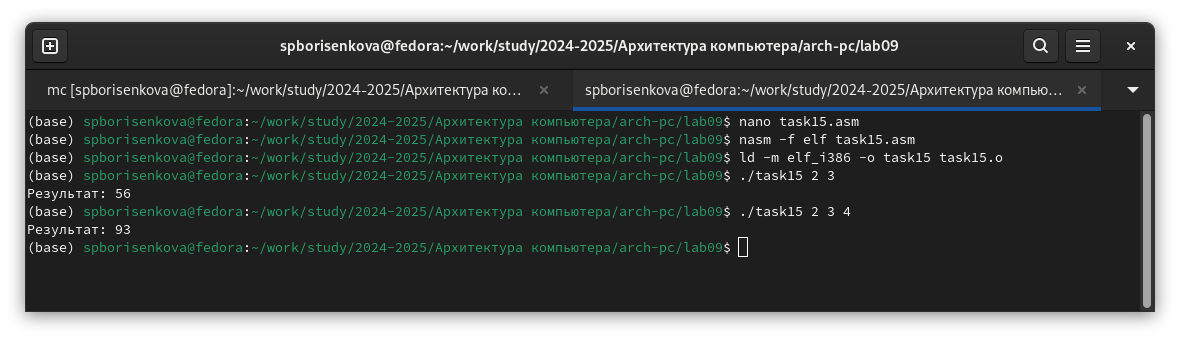
# 3 Выполнение задания для самостоятельной работы

Для выполнения самостоятельной работы создадим файл task15.asm. Скопируем в него программу из работы 8 и изменим соответственно заданию(рис. 3.1):



Код файла самостоятельной работы

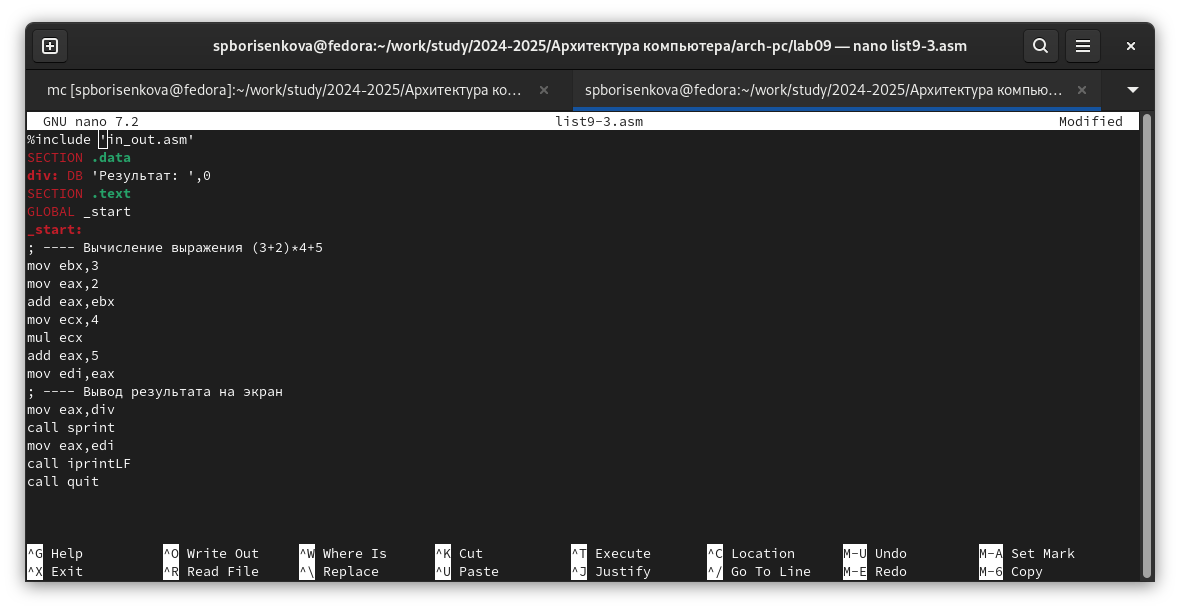
Соберём и запустим программу, вводя различные аргументы (рис. 3.2):



Сборка и запуск программы первого задания самостоятельной работы, а также результат выполнения

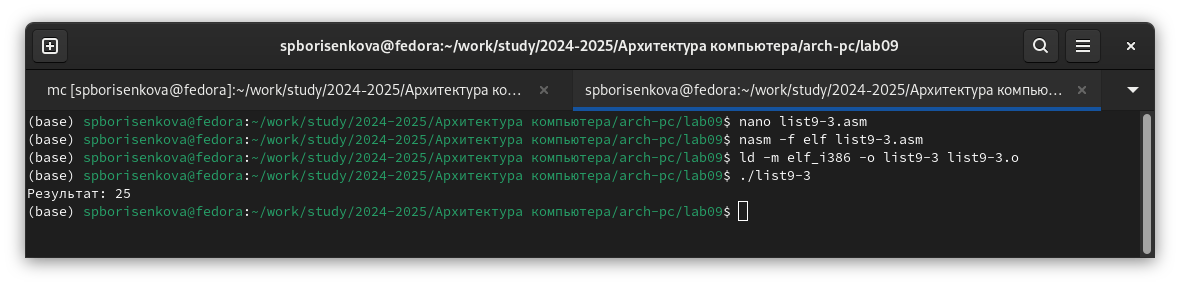
Пересчитав результат вручную, убеждаемся, что программа работает верно

Создадим файл list9-3.asm. Скопируем в него листинг и изменим соответственно заданию(рис. 3.3):



Код файла самостоятельной работы

Соберём и запустим программу (рис. 3.4):



Сборка и запуск программы второго задания самостоятельной работы, а также результат выполнения

Пересчитав результат вручную, убеждаемся, что теперь программа работает верно # Выводы

В результате выполнения лабораторной работы были получены представления о работе подпрограмм, а также было реализовано несколько программ, использующих подпрограммы. Также, были получены навыки работы с базовым функионалом gdb, и с помощью gdb была отловлена ошибка в коде программы