Отчёт по лабораторной работе №9

Понятие подпрограммы. Отладчик GDB

Грачева Мария Валерьевна

Содержание

# 1 Цель работы

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

# 2 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

Наиболее часто применяют следующие методы отладки: • создание точек контроля значений на входе и выходе участка программы (например, вывод промежуточных значений на экран — так называемые диагностические сообщения); • использование специальных программ-отладчиков.

GDB (GNU Debugger — отладчик проекта GNU) работает на многих UNIX-подобных системах и умеет производить отладку многих языков программирования. GDB предлагает обширные средства для слежения и контроля за выполнением компьютерных программ.

Установить точку останова можно командой break (кратко b). Типичный аргумент этой команды — место установки.

Информацию о всех установленных точках останова можно вывести командой info

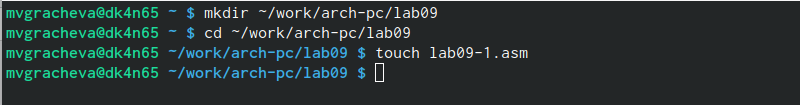
Для того чтобы сделать неактивной какую-нибудь ненужную точку останова, можно воспользоваться командой disable

Обратно точка останова активируется командой enable

*Подпрограмма* — это, как правило, функционально законченный участок кода, который можно многократно вызывать из разных мест программы. В отличие от простых переходов из подпрограмм существует возврат на команду, следующую за вызовом.

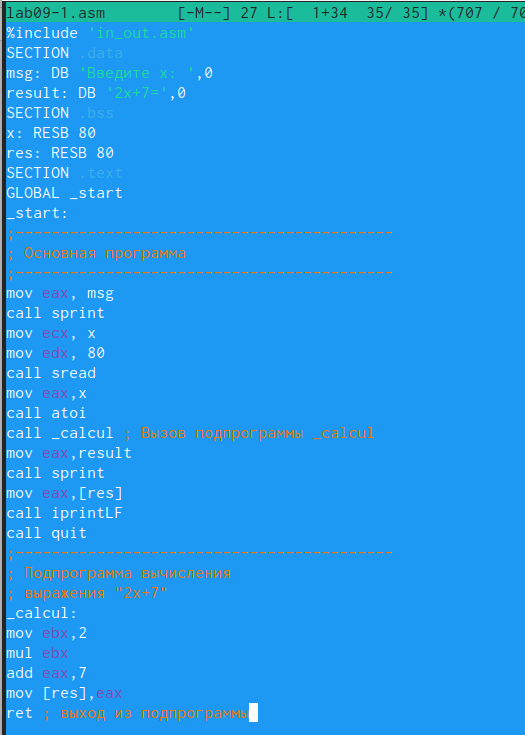
# 3 Выполнение лабораторной работы

Создаю каталог и файл lab09-1 (рис. ??).



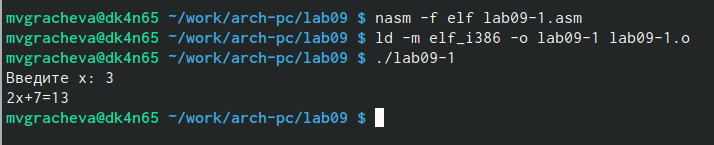
Создание каталога и файл lab09-1

Ввожу программу (рис. ??).



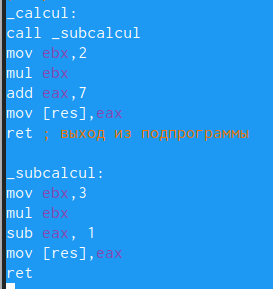
Введение программы

Проверяю работу файла (рис. ??).



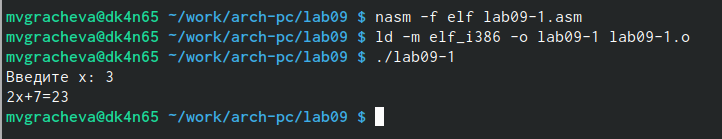
Проверка работы

Изменяю текст программы, добавив подпрограмму \_subcalcul в подпрограмму \_calcul, для вычисления выражения 𝑓(𝑔(𝑥)), где 𝑥 вводится с клавиатуры, 𝑓(𝑥) = 2𝑥 + 7, 𝑔(𝑥) = 3𝑥 − 1. (рис. ??)



Изменение текста

Проверяю его работу (рис. ??).



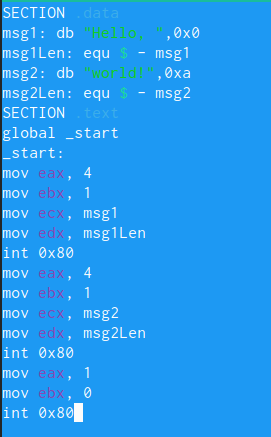
Проверка работы 2

Создаю файл lab09-2 (рис. ??).

Создание файла lab09-2

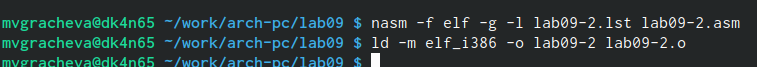
Создание файла lab09-2

Ввожу текст программы (рис. ??).



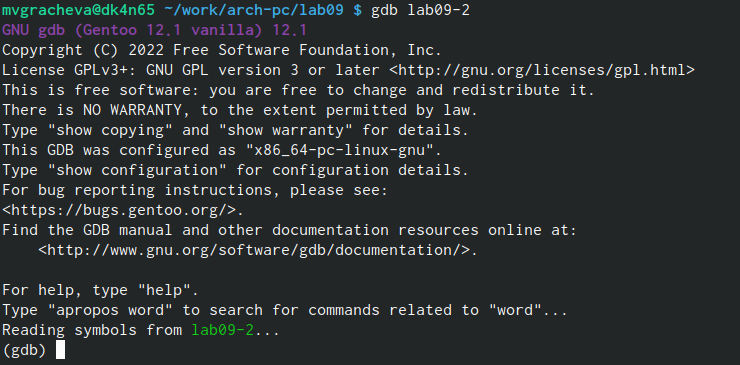
Введение программы

Создаю исполняемый файл (рис. ??).



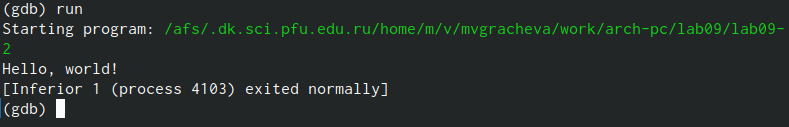
Создание исполняемого файла

Загружаю исполняемый файл в отладчик gdb (рис. ??).



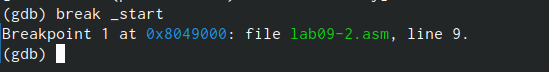
Загрузка исполняемого файла в отладчик gdb

Проверяю работу программы (рис. ??).



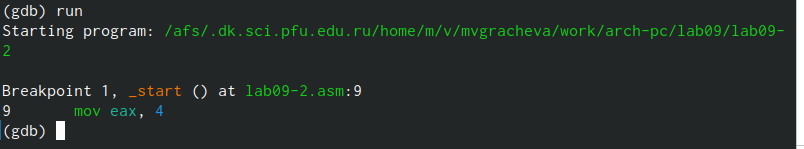
Проверка работы 3

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



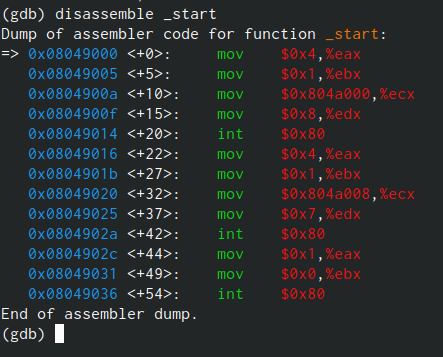
Установка брейкпоинта

Запускаю (рис. ??).



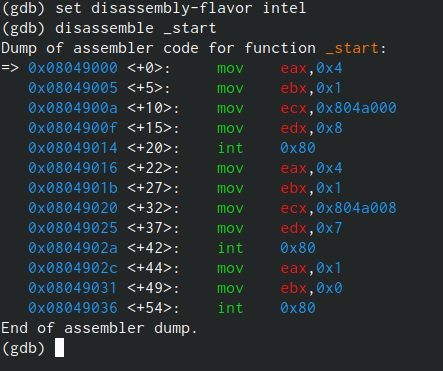
Запуск программы

Смотрю дисассимилированный код программы(рис. ??).



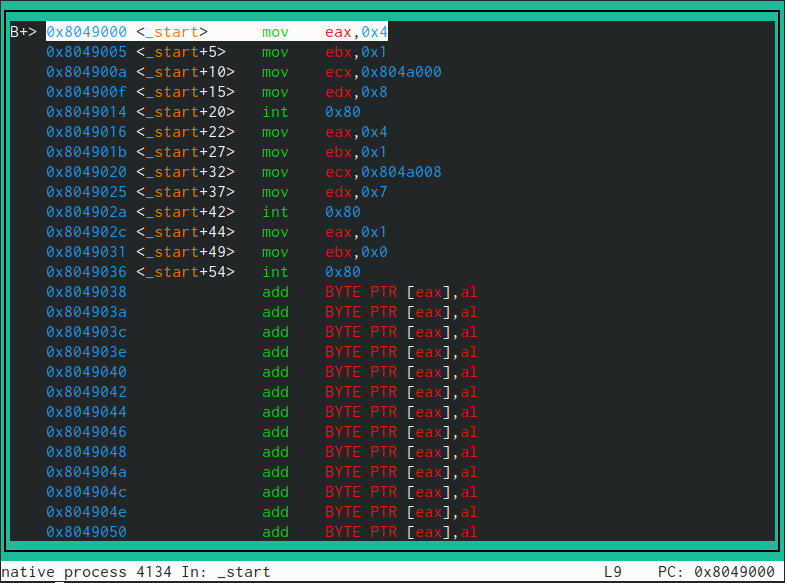
Дисассимилированный код программы

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

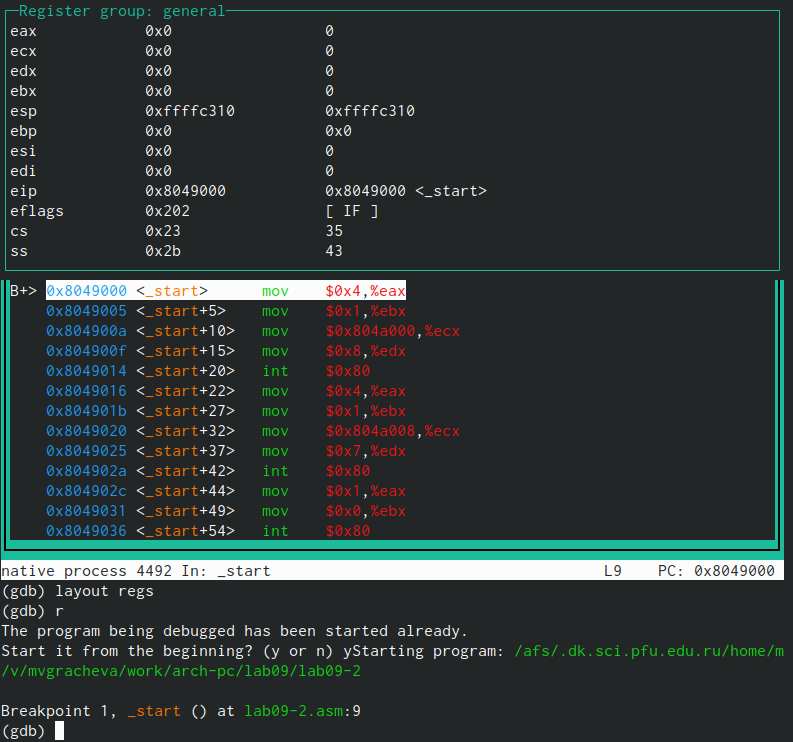


Отображение команд с Intel’овским синтаксисом

Включаю режим псевдографики для более удобного анализа программы (рис. ??), (рис. ??).

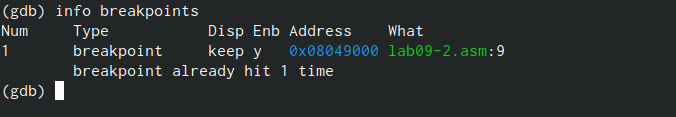


Режим псевдографики 1



Режим псевдографики 2

Ввожу команду info breakpoints (рис. ??).

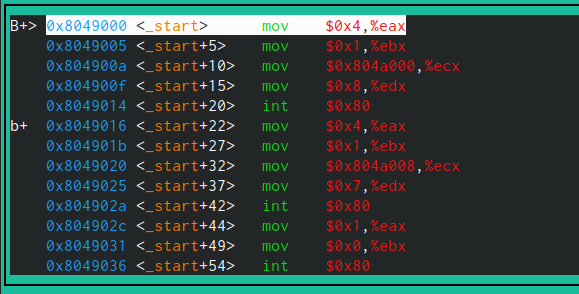


Команда info breakpoints

Устанавливаю новую точку останова (рис. ??), (рис. ??).

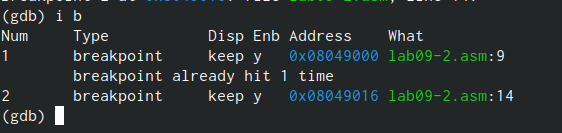
Точка останова 1

Точка останова 1



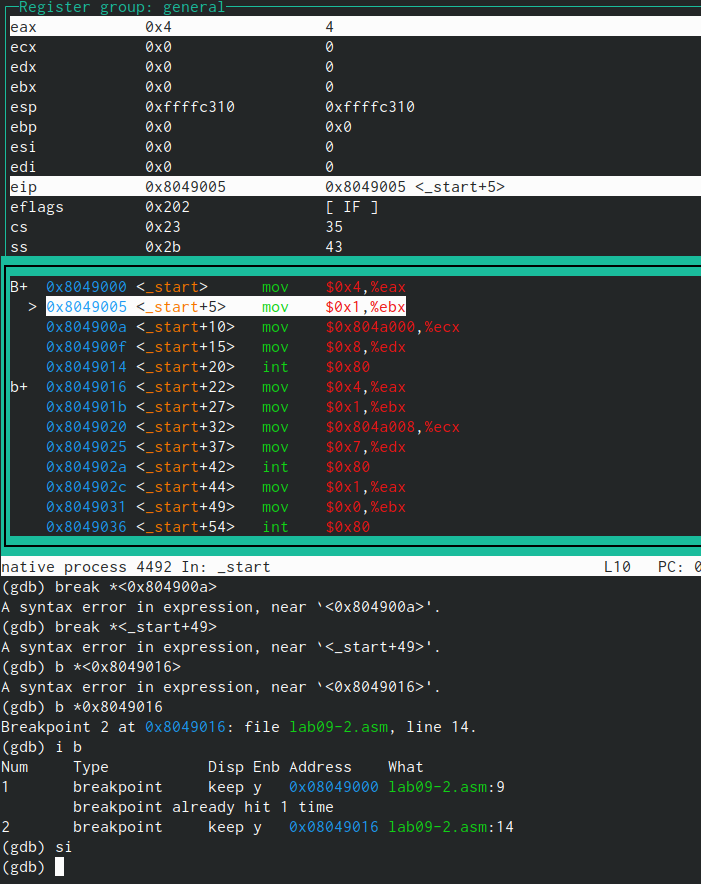
Точка останова 2

Смотрю инфу про все установленные точки (рис. ??).

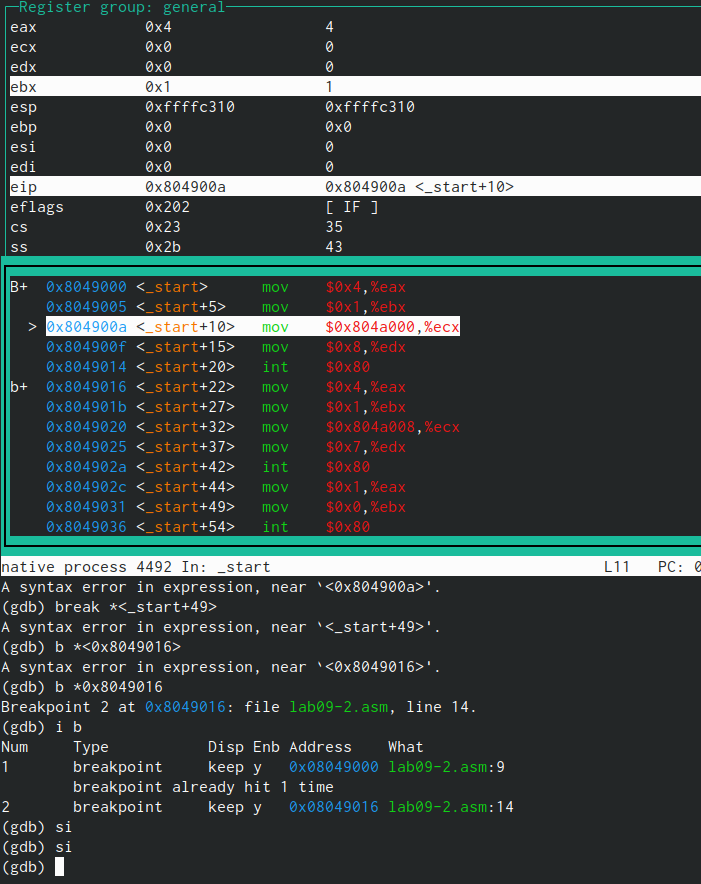


Команда info breakpoints 2

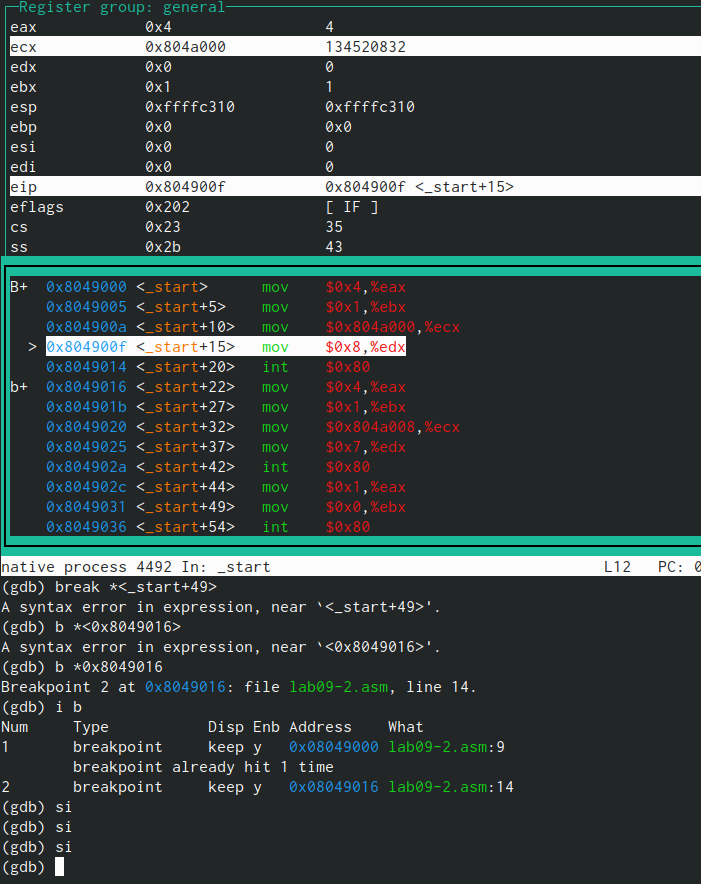
Выполняю пять раз комнду si (рис. ??), (рис. ??), (рис. ??), (рис. ??), (рис. ??).



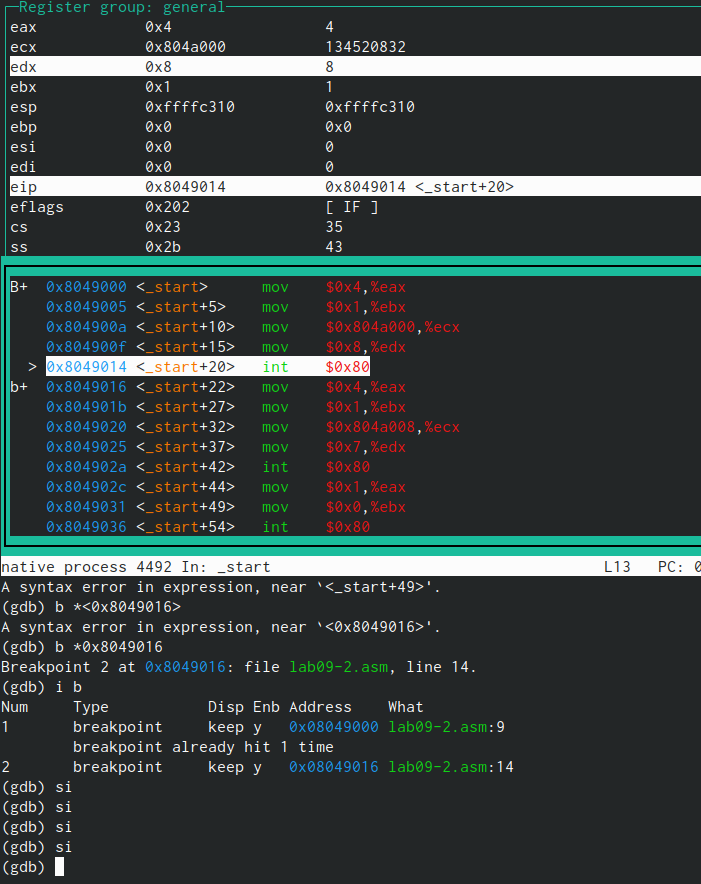
Выполнение команды si 1



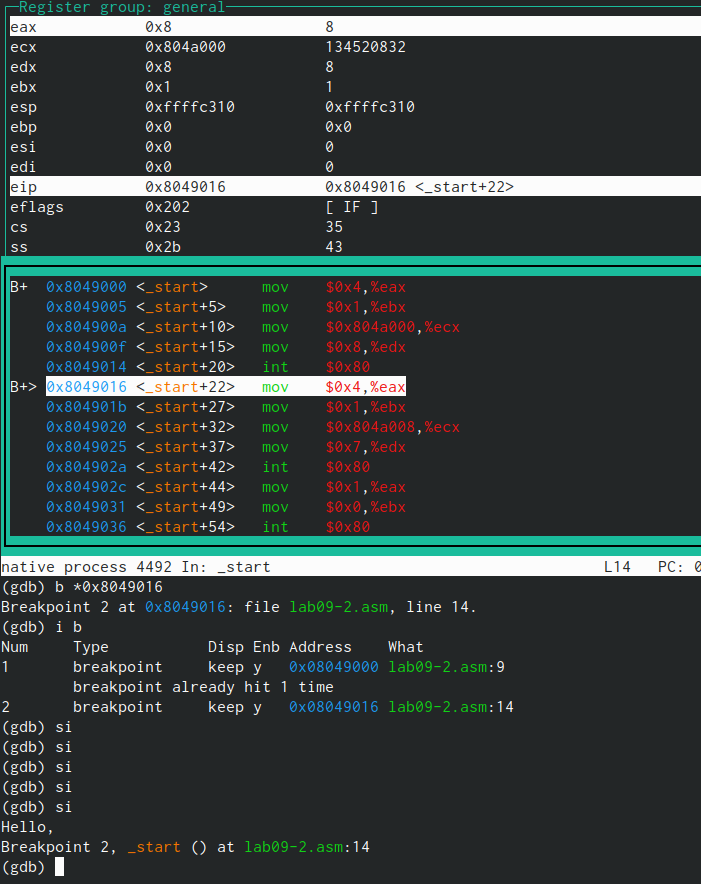
Выполнение команды si 2



Выполнение команды si 3

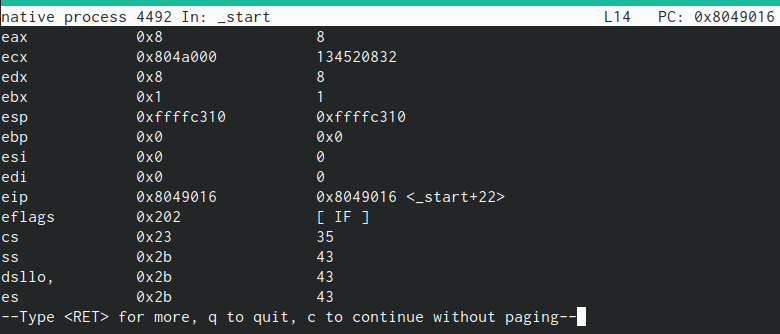


Выполнение команды si 4



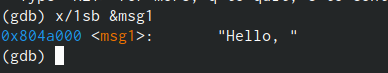
Выполнение команды si 5

Смотрю содержимое регистров с помощью команды info registers (рис. ??).



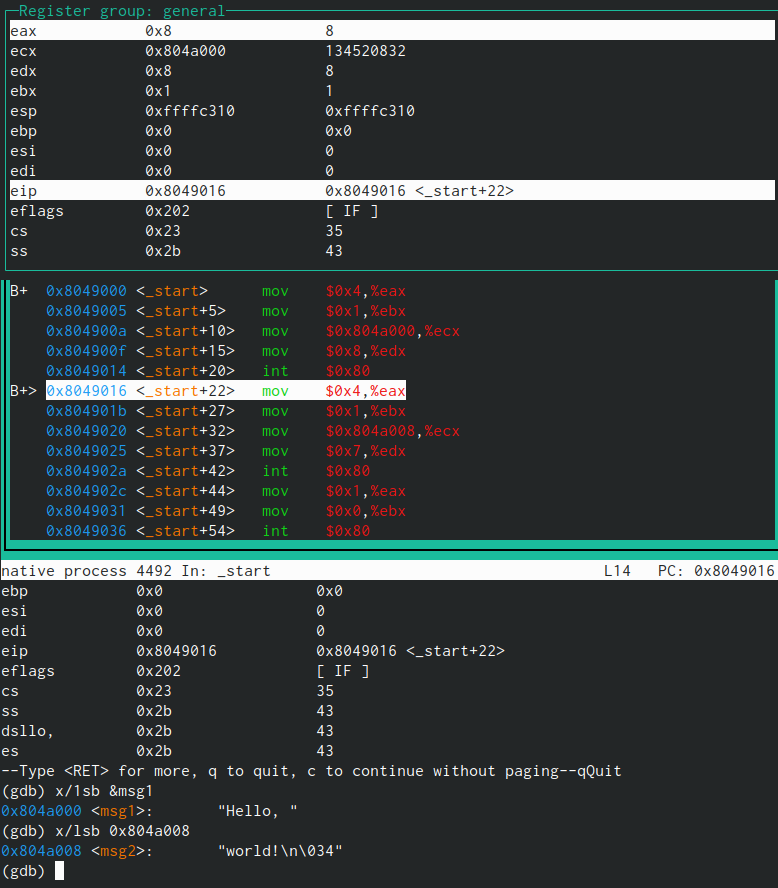
Команда info registers

Смотрю значение переменной msg1 по имени(рис. ??).



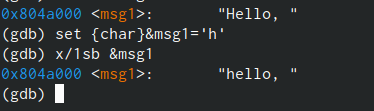
Значение переменной msg1 по имени

Смотрю значение переменной msg2 по адресу (рис. ??).

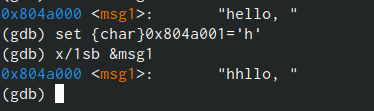


Значение переменной msg2 по адресу

Заменяю первый (рис. ??) и второй символы в первой переменной (рис. ??).

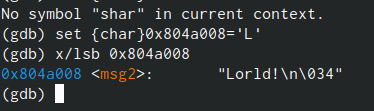


Замена символа 1



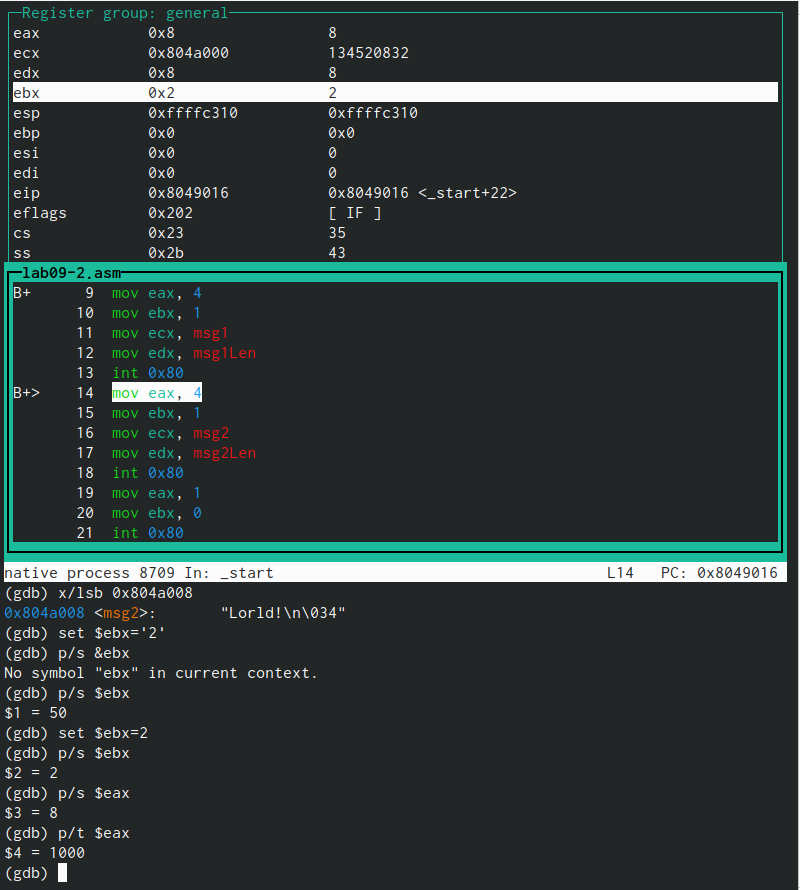
Замена символа 2

Заменяю первый символ во второй переменной(рис. ??)



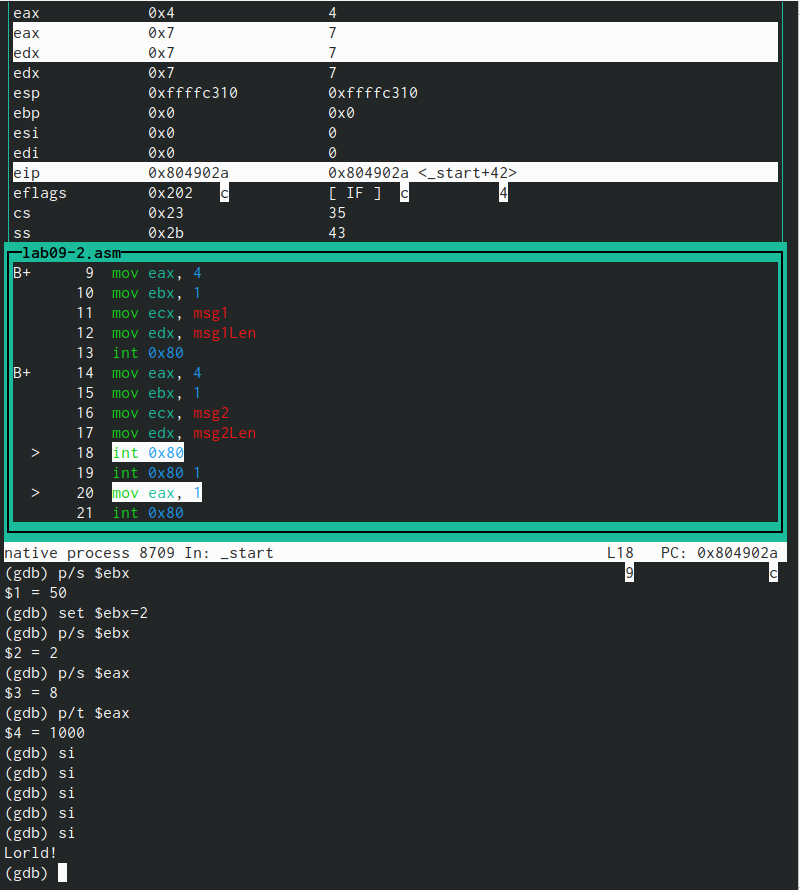
Замена символа 3

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



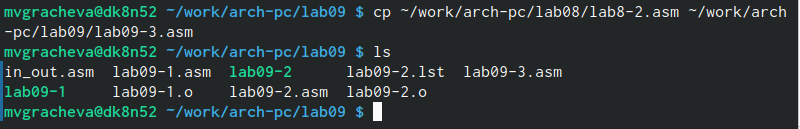
Изменение значение регистра ebx

Завершаю выполнение программы с помощью команды si (рис. ??).



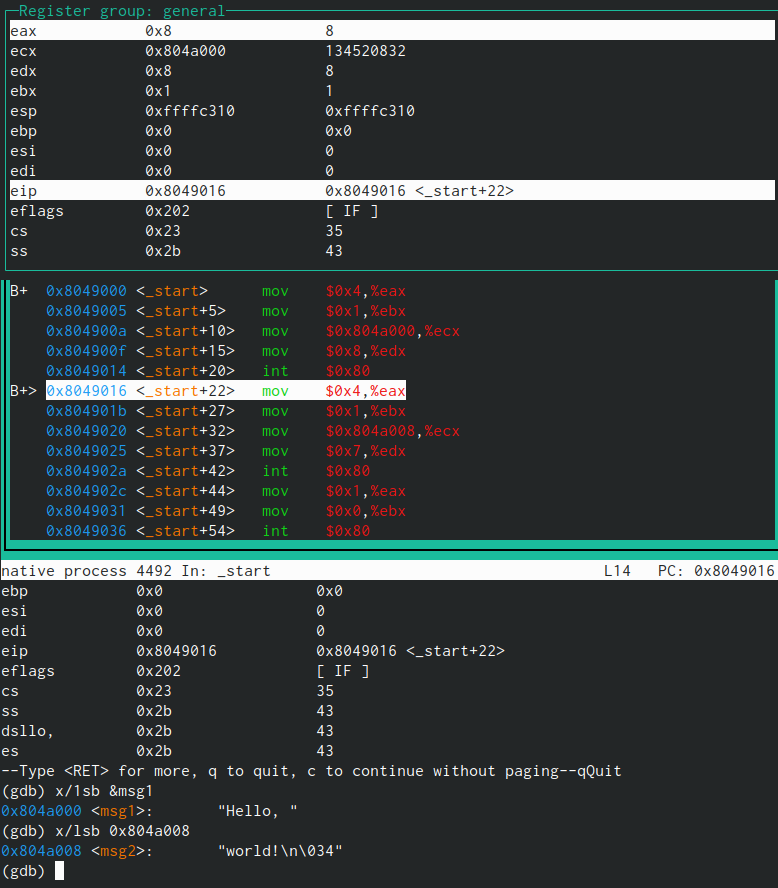
Завершение программы

Скопировала файл lab8-2.asm, созданный при выполнении лабораторной работы No8, с программой выводящей на экран аргументы командной строки (Листинг 8.2) в файл с именем lab09-3.asm(рис. ??).



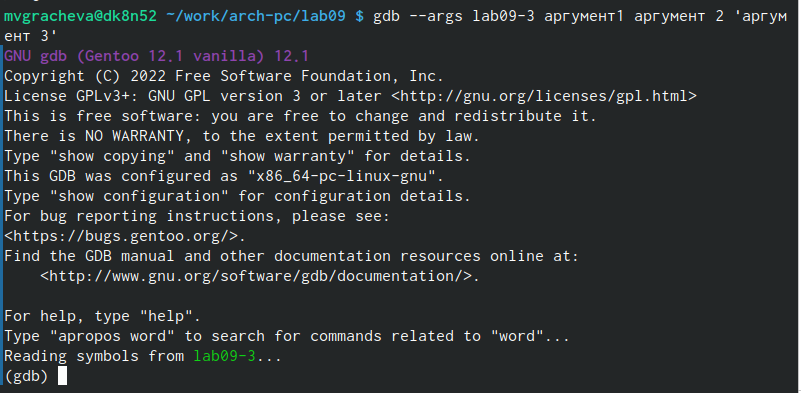
Копирование файла

Создаю исполняемый файл (рис. ??).



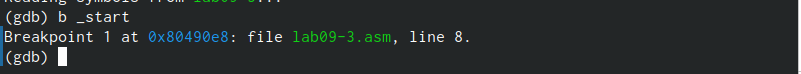
Создание исполняемого файла

Загружаю исполняемый файл в отладчик, указав аргументы(рис. ??).

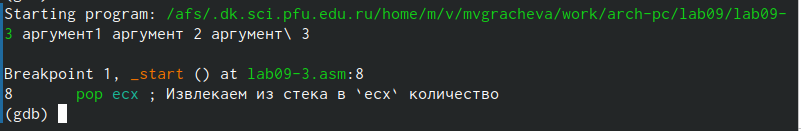


Загрузка в отладчик

Устанавливаю точку останова (рис. ??) и запускаю файл (рис. ??).

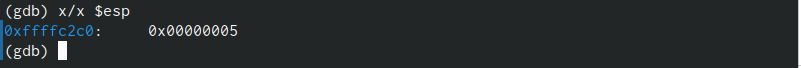


Точка останова



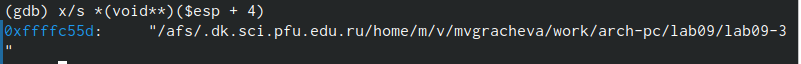
Запуск файла

Адрес вершины стека храниться в регистре esp и по этому адресу располагается число равное количеству аргументов командной строки (включая имя программы)(рис. ??).

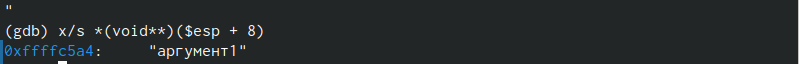


Команда x/x

Смотрю остальные позиции стека (рис. ??), (рис. ??), (рис. ??), (рис. ??), (рис. ??), (рис. ??).



Команда x/s



Команда x/s 2

Команда x/s 3

Команда x/s 3

Команда x/s 4

Команда x/s 4

Команда x/s 5

Команда x/s 5

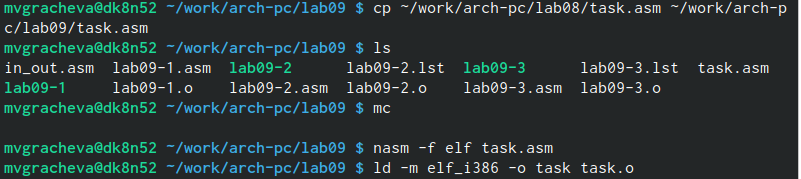
Команда x/s 6

Команда x/s 6

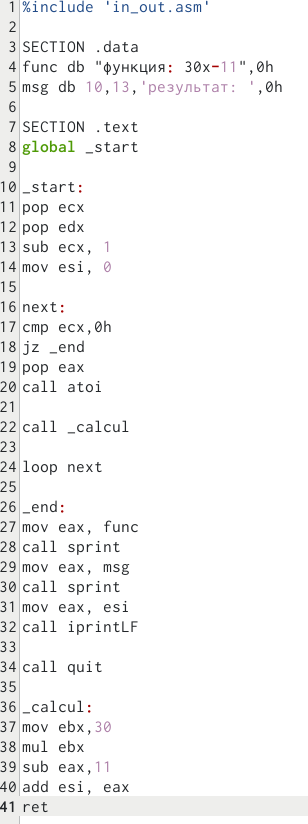
# 4 Самостоятельная работа

**Задание 1**

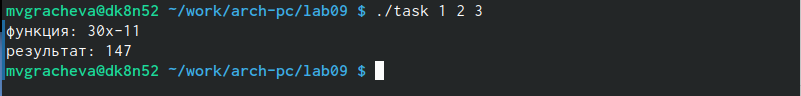
Копирую файл, созданный в ходе выполнения предыдущей лабораторной (рис. ??) , вношу изменения в текст (рис. ??) и проверяю работу (рис. ??).



Копирование файла, созданного в ходе выполнения предыдущей лабораторной



Изменения в текст



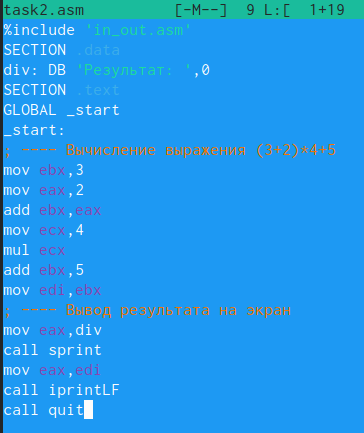
Проверка работы 4

**Задание 2**

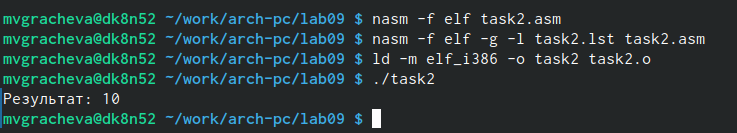
Создаю файл (рис. ??), вношу текст (рис. ??), проверяю работу файла (рис. ??)

Создание файла task2

Создание файла task2

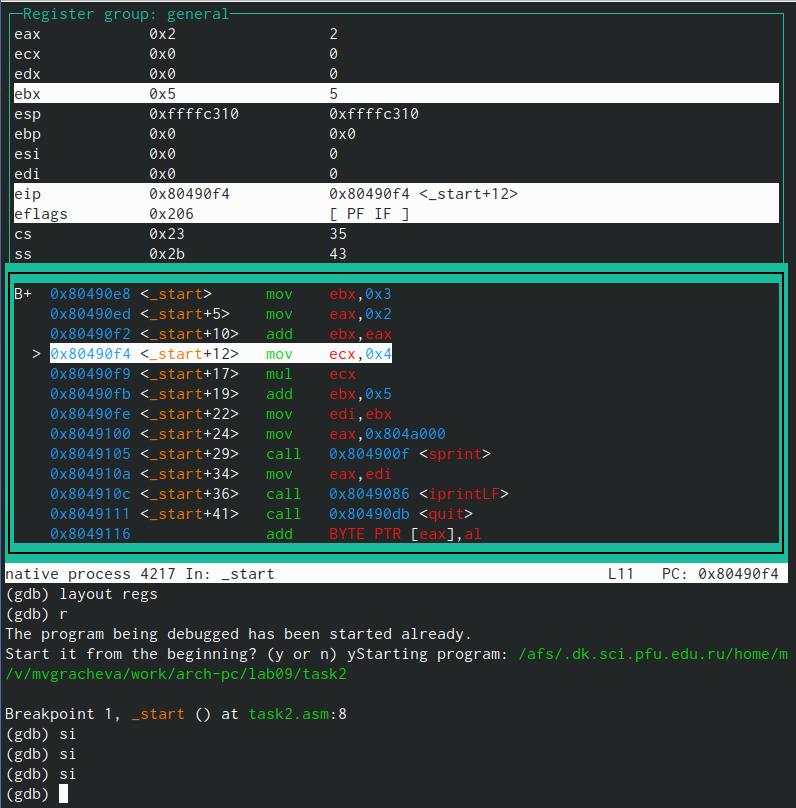


Введение текста



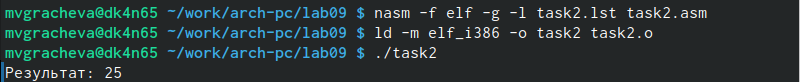
Проверка работы 5

Включите режим псевдографики (зарание изменив синтаксис для удобства). С помощью команды si поэтапно проверяю код и вижу, что значение после сложение сохраняется не в тот регистр(рис. ??).



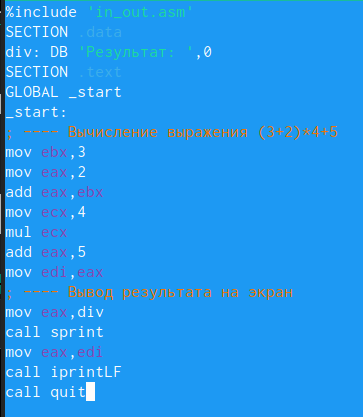
Поиск ошибки

Меняю местами регистр + в дальнейших шагах меняем eax на ebx(рис. ??).



Правки в исходной программе

Проверяю работу файла (рис. ??).



Проверка работы 6

# 5 Выводы

Приобрела навыки написания программ с использованием подпрограмм.Ознакомилась с методами отладки при помощи GDB и его основными возможностями.

# Список литературы