**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

# ОТЧЕТ
# ПО ЛАБОРАТОРНОЙ РАБОТЕ №9

*дисциплина: Архитектура компьютеров и операционные системы*

Студент: Галиев Самир Салаватович

Группа: НКАбд-02-25

**МОСКВА**

2025 г.

**Оглавление**

## 1. Цель работы

Приобретение навыков написания программ с использованием подпрограмм на ассемблере NASM, а также освоение методов отладки с помощью отладчика GDB: установка точек останова, пошаговое выполнение, просмотр и изменение содержимого регистров и памяти.

## 2. Выполнение задания

### 2.1 Реализация подпрограмм в NASM

**Описание задания:** Написать программу lab09-1.asm, вычисляющую значение функции f(x)=2x+7 с использованием подпрограммы _calcul. Затем модифицировать программу, вычисляющую g(x)=3x−1, и реализовать композицию f(g(x)).

```
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab09 % nasm -f elf32 lab09-1.asm -o lab09-1.o
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab09 % ld -m elf_i386 lab09-1.o -o lab09-1
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab09 % ./lab09-1
Введите x:3
2x+7=13
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab09 %
```

*Рисунок 1: вывод программы lab09-1.asm при x = 3*

**Комментарий:** Программа запрашивает x и вычисляет 2x+72x+7. При вводе x=3 выводится 13 (результат верный).

```
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab09 % nano lab09-1.asm
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab09 % nasm -f elf32 lab09-1.asm -o lab09-1.o
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab09 % ld -m elf_i386 lab09-1.o -o lab09-1
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab09 % ./lab09-1
Введите x:2
2x+7=17
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab09 %
```

*Рисунок 2: вывод программы вычисляющая f(g(x))*

**Комментарий:** Программа вычисляет f(g(x))=2(3x−1)+7. При x = 2 вывод 17(ответ верный).

## 2.2 Откладка программы с помощью GDB

**Описание задания:** Отладить программу lab09-2.asm (Hello, world!) с использованием GDB: запуск с точкой останова, пошаговое выполнение, просмотр и изменение памяти и регистров.

```
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab09 % nano lab09-2.asm
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab09 % nasm -f elf32 -g -l lab09-2.lst lab09-2.asm -o lab09-2.o
ld -m elf_i386 -o lab09-2 lab09-2.o
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab09 % ./lab09-2
Hello,world!
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab09 %
```

*Рисунок 3: Вывод программы lab09-2.asm*

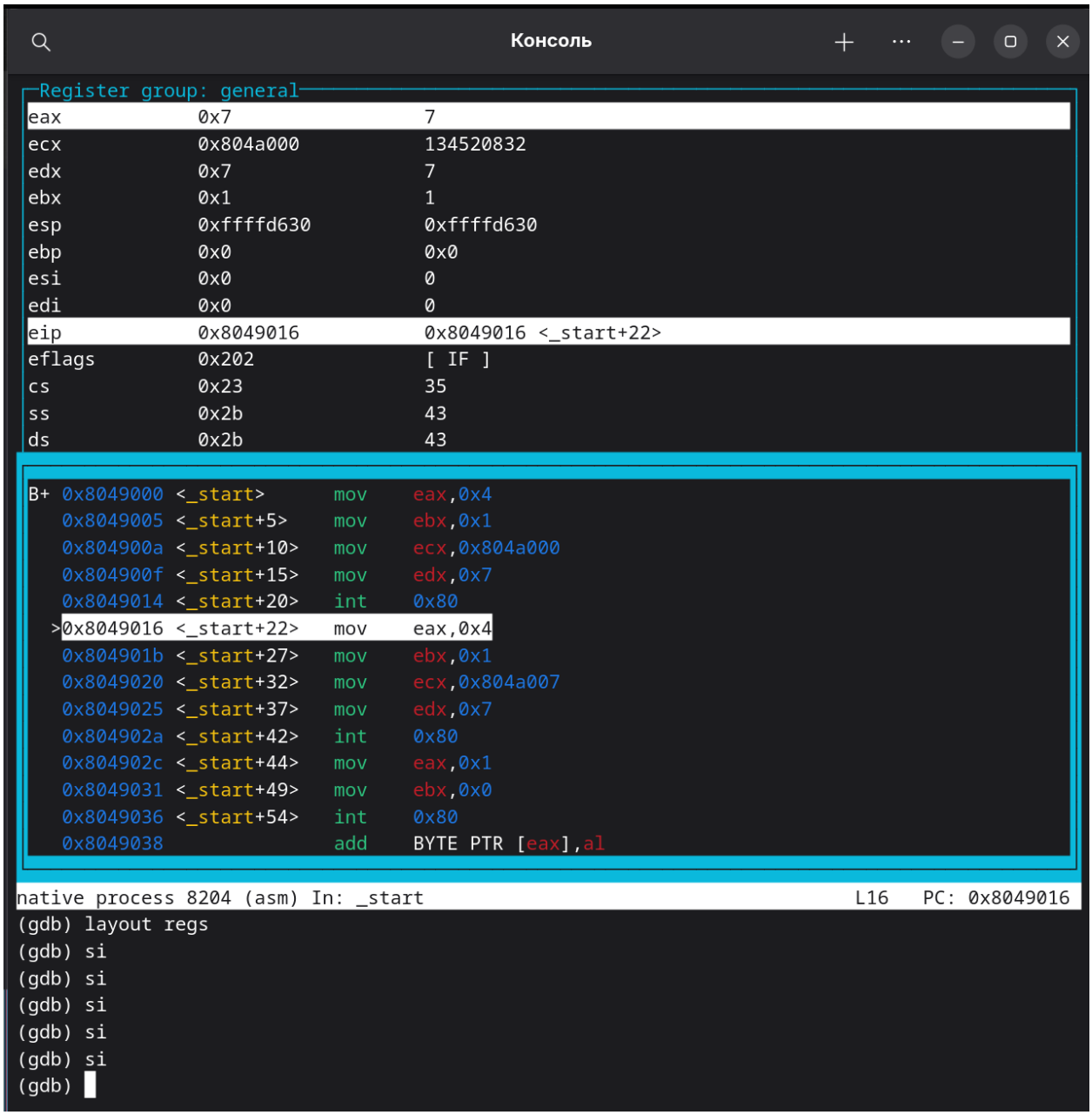


*Рисунок 4: После пяти si регистры содержат eax = 4, ebx = 1, ecx = msg1, edx = 6.*

**Комментарий:** Включен удобный режим интерфейса (Верхняя панель — регистры, Средняя — дизассемблированный код, а Нижняя — командная строка)

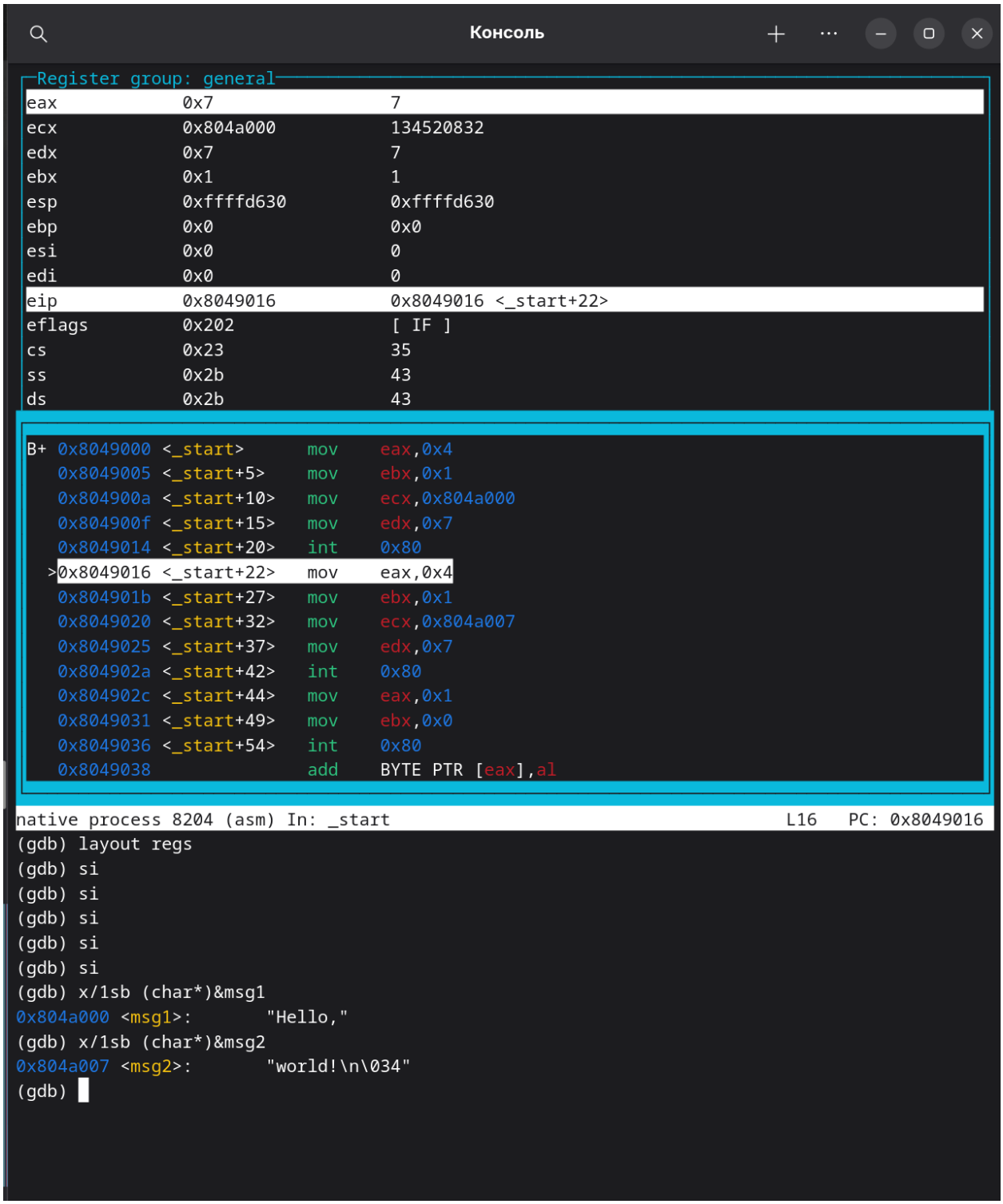


*Рисунок 5: После двух команд вызова msg1 и msg2 нам показывают строки «Hello,» и «world!\n\034»*

**Комментарий:** использование команд x/1sb (char*)&msg1 и x/1sb (char*)&msg2 показывают исходные строки.

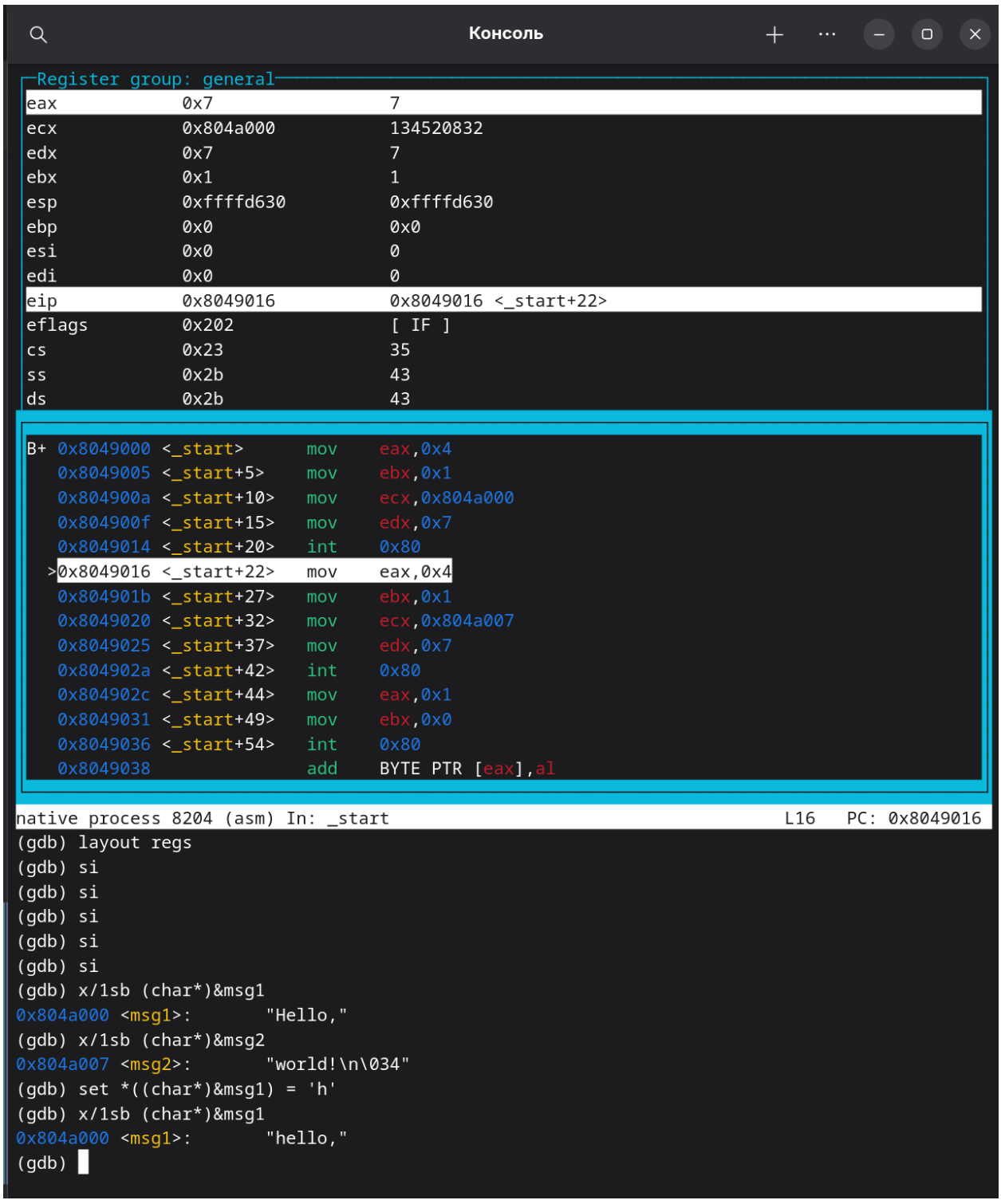


*Рисунок 6: замена первой буквы в слове «Hello»*

**Комментарий:** С помощью set *((char*)&msg1) = 'h' первая буква изменена. Повторный вывод показывает: "hello,".

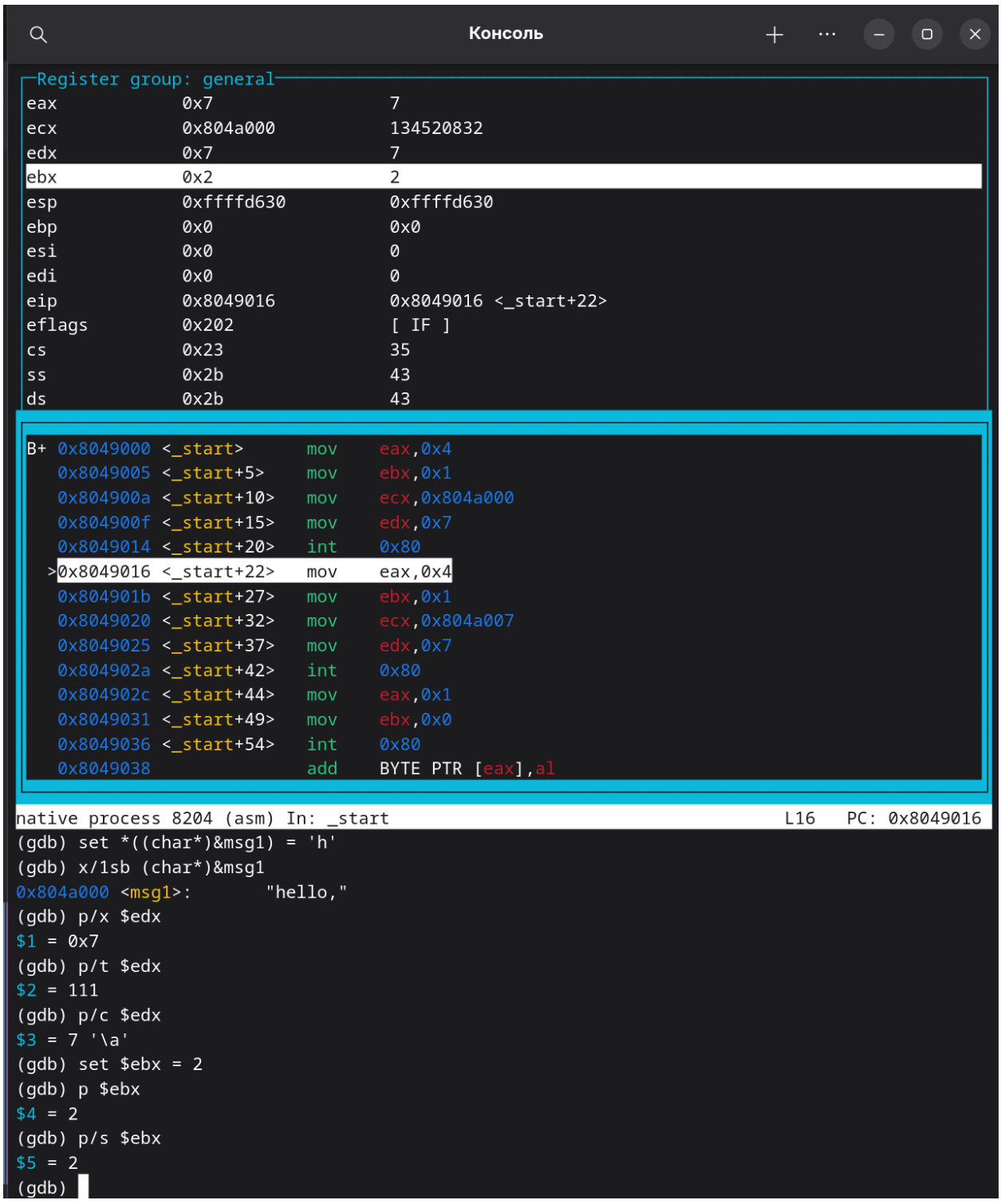


*Рисунок 7: вывод переменных в шестизначном значении и д.р. и т. д.*

**Комментарий:** Команда p/s $ebx пытается интерпретировать число 2 как адрес строки, что в корректной отладочной среде приводит к ошибке или мусору. Это демонстрирует разницу между числовым значением и указателем на строку.

(Вывод $5 = 2 это упрощённое поведение GDB для удобства, но сути не отменяет)

### 2.3 Анализ аргументов командной строки в GDB

**Описание задания:** Использовать GDB для изучения расположения аргументов командной строки в стеке при запуске программы.

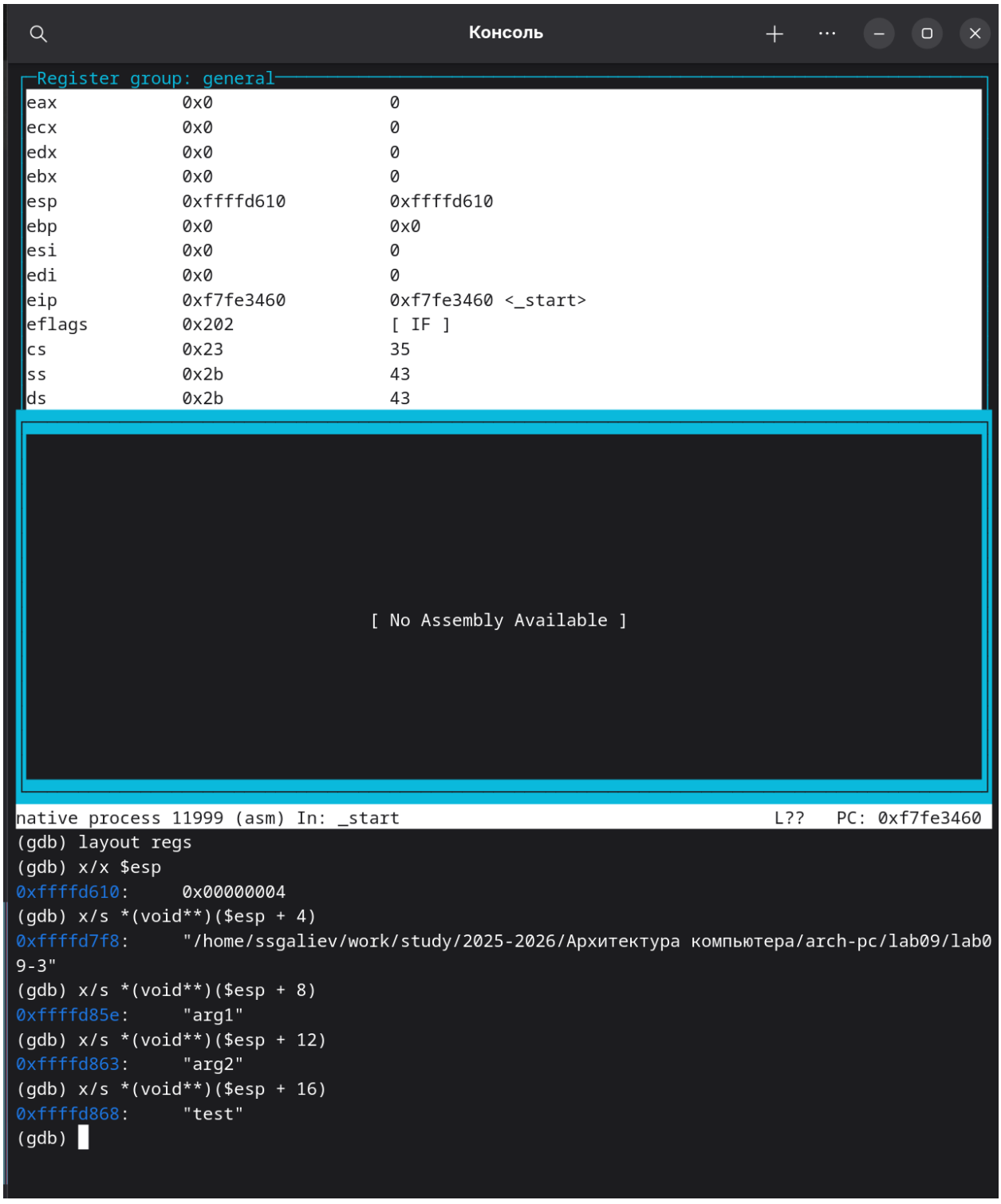


*Рисунок 8: запуск программы с аргументами arg1, arg2, test*

**Комментарий:** +8, +12, +16 при arg1, arg2, test.

Шаг = 4 байта, так как в 32-битной системе указатель занимает 4 байта.

## 3. Самостоятельная работа

### 3.1 Реализация функции f(x) = 15*x+2 в виде подпрограммы

**Описание задания:** Преобразовать программу из лабораторной работы №8 в версию с подпрограммой _f, вычисляющей f(x)=15x+2 для каждого аргумента и суммирующей результаты.

```
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab09 % nano lab09-4.asm
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab09 % nasm -f elf32 lab
09-4.asm -o lab09-4.o
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab09 % gcc -m32 -nostdli
b lab09-4.o -o lab09-4
/usr/bin/ld: lab09-4.o: предупреждение: перемещение в разделе только для чтения «.text»
/usr/bin/ld: предупреждение: создаётся DT_TEXTREL в PIE
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab09 % ./lab09-4 1 2 3
Функция: f(x)=15x+2
Результат:96
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab09 %
```

*Рисунок 9: Вывод программы.*

**Комментарий:** Программа запущена с аргументами 1 2 3. Вывод 96.

### 3.2 Исправление ошибки в листинге 9.3

**Описание задания:** В листинге 9.3 вычисляется выражение (3+2)·4+5=25, но из-за ошибки в использовании mul программа выводит неверный результат. Требуется найти и исправить ошибку с помощью GDB.

```
Результат:96
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab09 % nano lab09-5.asm
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab09 % nasm -f elf32 lab
09-5.asm -o lab09-5.o
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab09 % gcc -m32 -nostdli
b lab09-5.o -o lab09-5
/usr/bin/ld: lab09-5.o: предупреждение: перемещение в разделе только для чтения «.text»
/usr/bin/ld: предупреждение: создаётся DT_TEXTREL в PIE
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab09 % ./lab09-5
Результат: 10
ssgaliev@linux ~/work/study/2025-2026/Архитектура компьютера/arch-pc/lab09 %
```

*Рисунок 10: Неверный вывод программы.*

**Комментарий:** До исправления программа выводит 10, что неверно. Ошибка: mul ecx умножает eax, а не ebx.

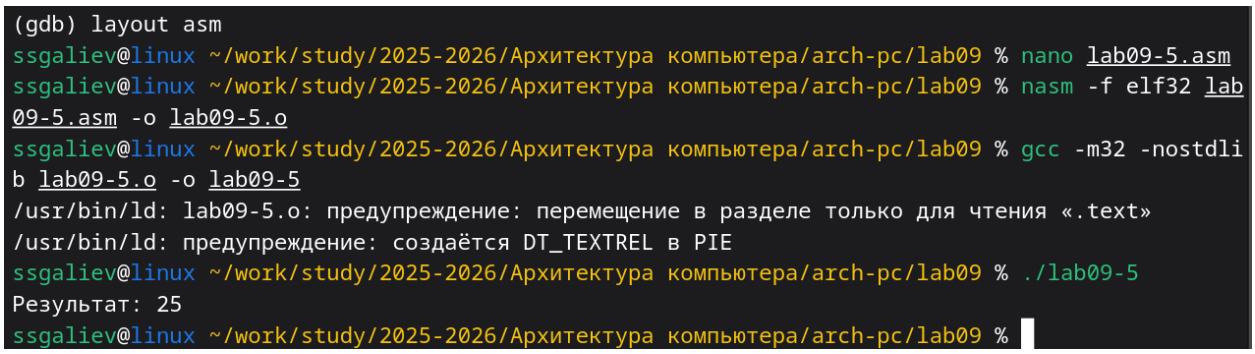


*Рисунок 11: Корректный вывод программы после исправления.*

**Комментарий:** После исправления сумма 3+2 помещена в eax, mul ebx(где ebx = 4), затем add eax, 5. В итоге мы получаем верный результат(25).

**4. Выводы**

Подпрограммы (call/ret) позволяют структурировать код, изолировать логику и повторно использовать функциональность.

1)Стек используется для передачи адреса возврата, а также для временного хранения данных.

2) GDB — мощный инструмент для отладки:

- break, si, continue — управление выполнением,
- x, p, set — анализ и изменение памяти/регистров,
- layout asm, layout regs — удобный режим интерфейса.

- При работе с mul важно помнить: первый операнд — всегда eax.
- Аргументы командной строки размещаются в стеке: argc, имя программы, затем аргументы (по 4 байта на указатель в 32-битной системе).

Цель работы полностью достигнута.

## 5. Список литературы

1)Демидова А. В.  Архитектура ЭВМ. Лабораторная работа №9.

2)Столяров А.  Программирование на языке ассемблера NASM для ОС Unix.  — М.: МАКС Пресс, 2011.

3)GDB: The GNU Project Debugger. —  https://www.gnu.org/software/gdb/

4)The NASM documentation. —  https://www.nasm.us/docs.php

5)Debuginfod — Arch Linux Wiki. —  https://wiki.archlinux.org/title/Debuginfod