**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**ОТЧЕТ**

**по лабораторной работе № 9**

*дисциплина: Архитектура компьютер*

Студенты: Выонг Нам Кхань

Группа: НКАбд-02-23

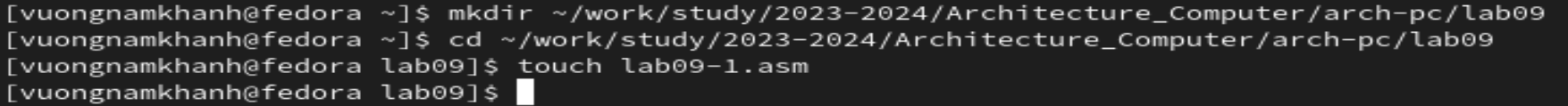
**МОСКВА**

2023 г.

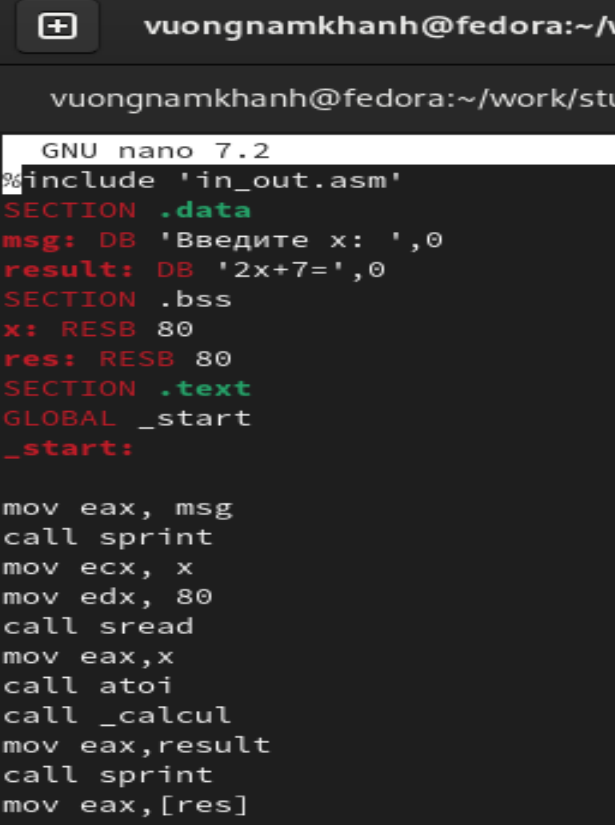
**Выполнение лабораторной работы**

1. Реализация подпрограмм в NASM.

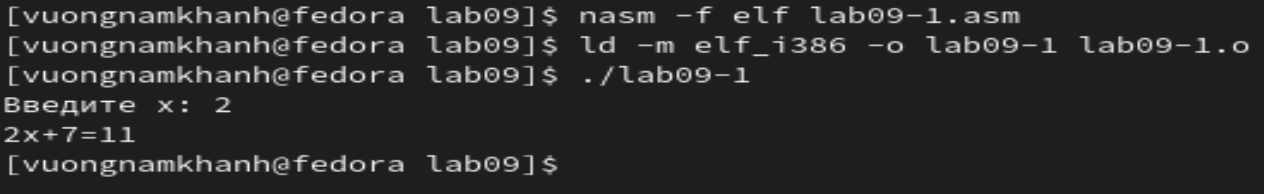
Создаю каталог для выполнения лабораторной работы № 9, перехожу в него и создаю файл lab09-1.asm.



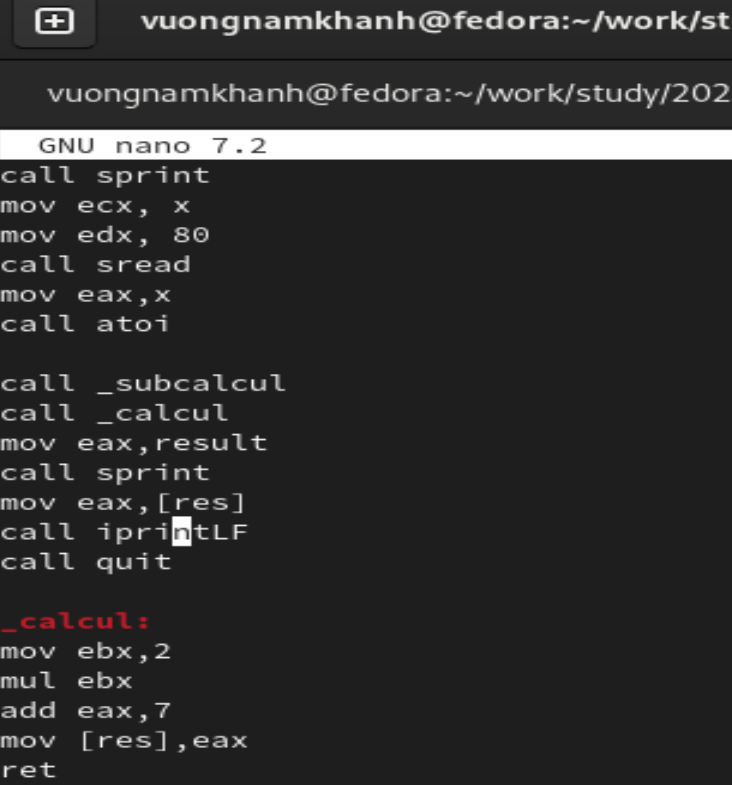
Ввожу в файл lab09-1.asm текст программы с использованием подпрограммы из листинга 9.1.



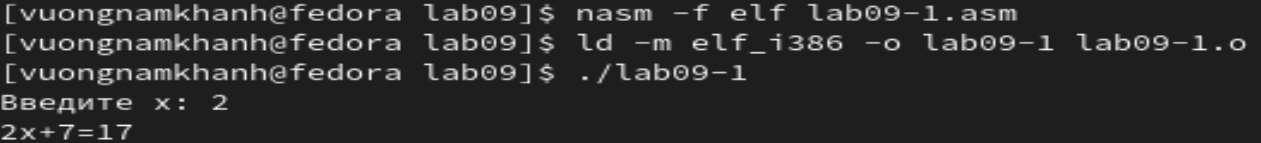
Создаю исполняемый файл и проверяю его работу



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.

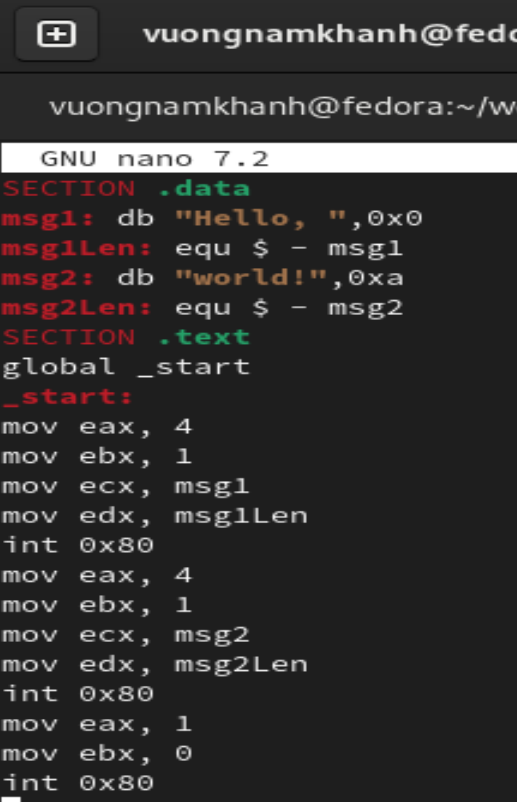


Создаю исполняемый файл и проверяю его работу.

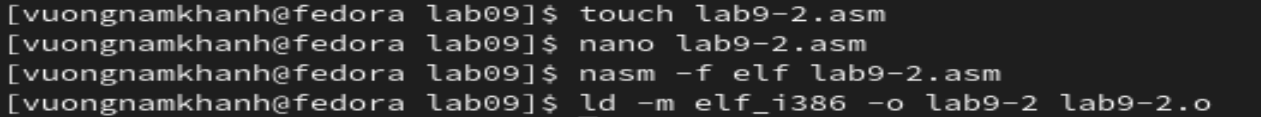


**Отладка программам с помощью GDB.**

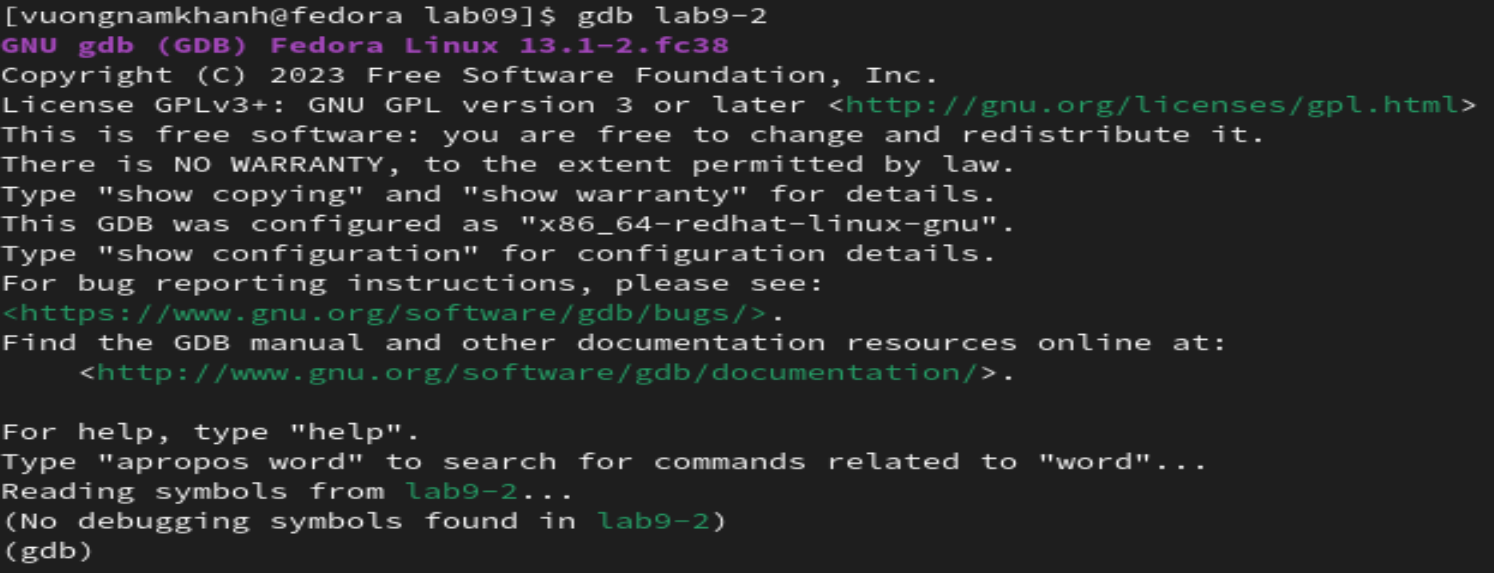
На этом шаге мы создали файл lab09-2.asm с текстом программы из ли- стинга 9.2



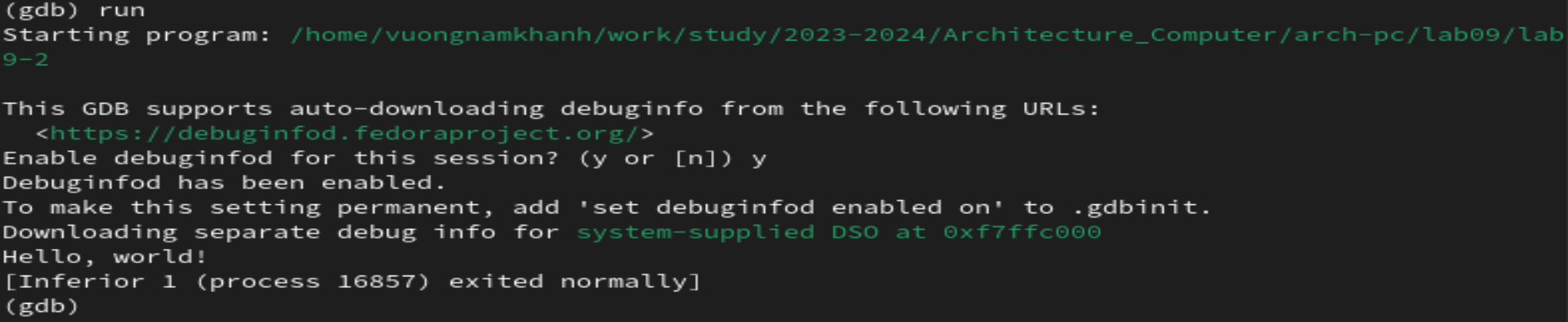
Получаю исполняемый файл для работы с GDB с ключом ‘-g’.



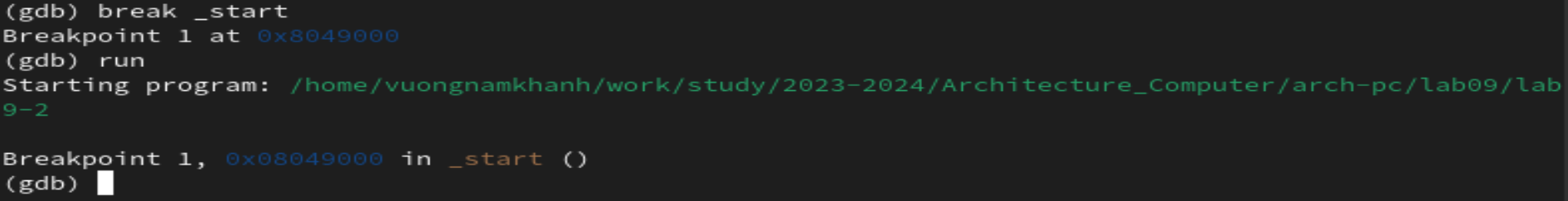
Загружаю исполняемый файл в отладчик gdb.



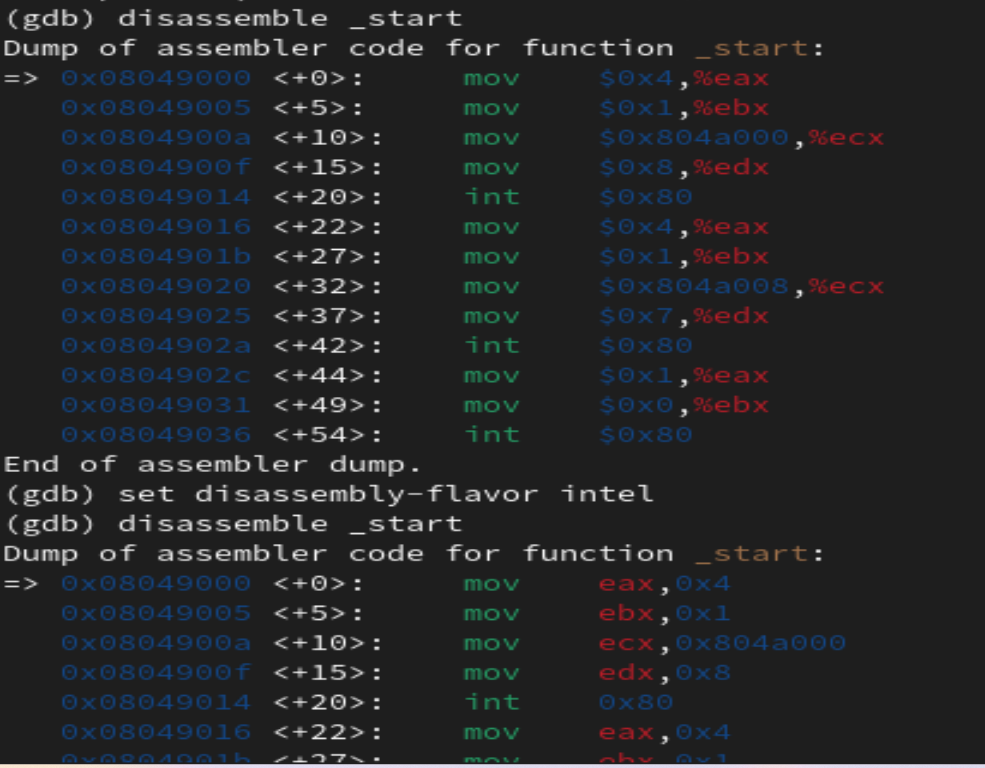
Проверяю работу программы, запустив ее в оболочке GDB с помощью команды run.



Для более подробного анализа программы устанавливаю брейкпоинт на метку \_start и запускаю её.

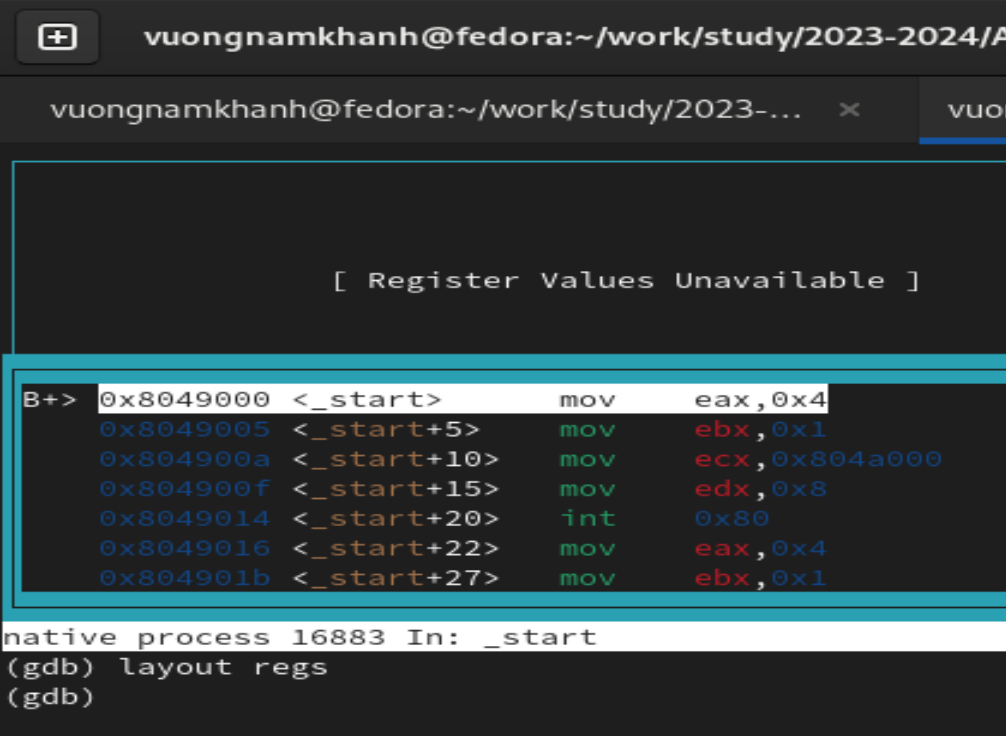


Просматриваю дисассимилированный код программы с помощью команды disassemble, начиная с метки \_start, и переключаюсь на отображение команд с синтаксисом Intel, введя команду set disassembly-flavor intel



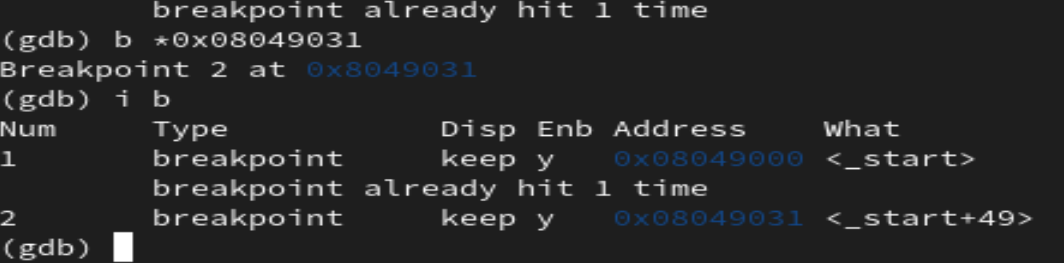
Разница в синтаксисе между AT&T и INTEL заключается в том, что AT&T использует синтаксис mov $0x4,%eax, который популярен среди пользова- телей Linux, с другой стороны, INTEL использует синтаксис mov eax,0x4 , который является популярен среди пользователей Windows.

Включаю режим псевдографики для более удобного анализа программы с помощью команд layout asm и layout regs



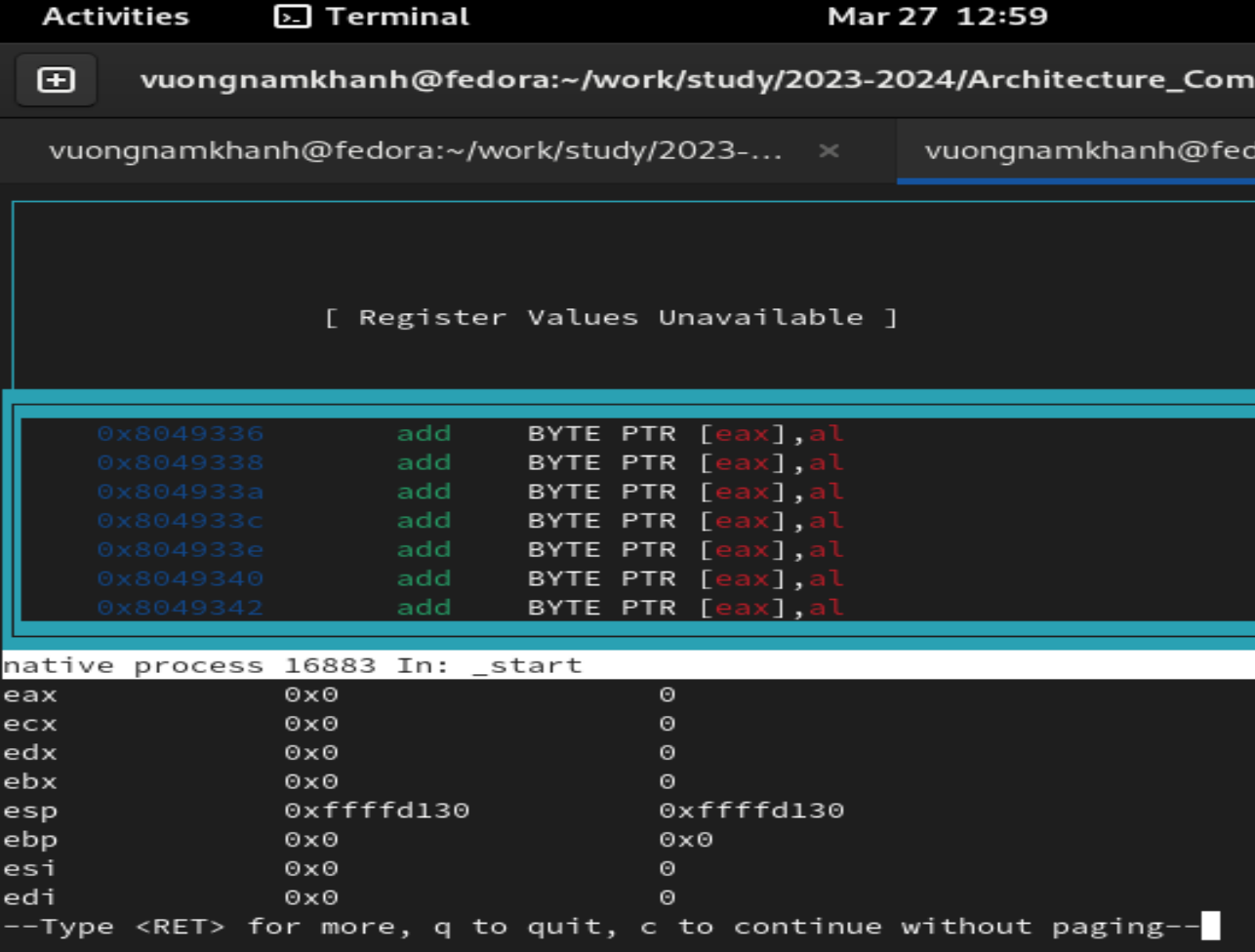
**Добавление точек останова.**

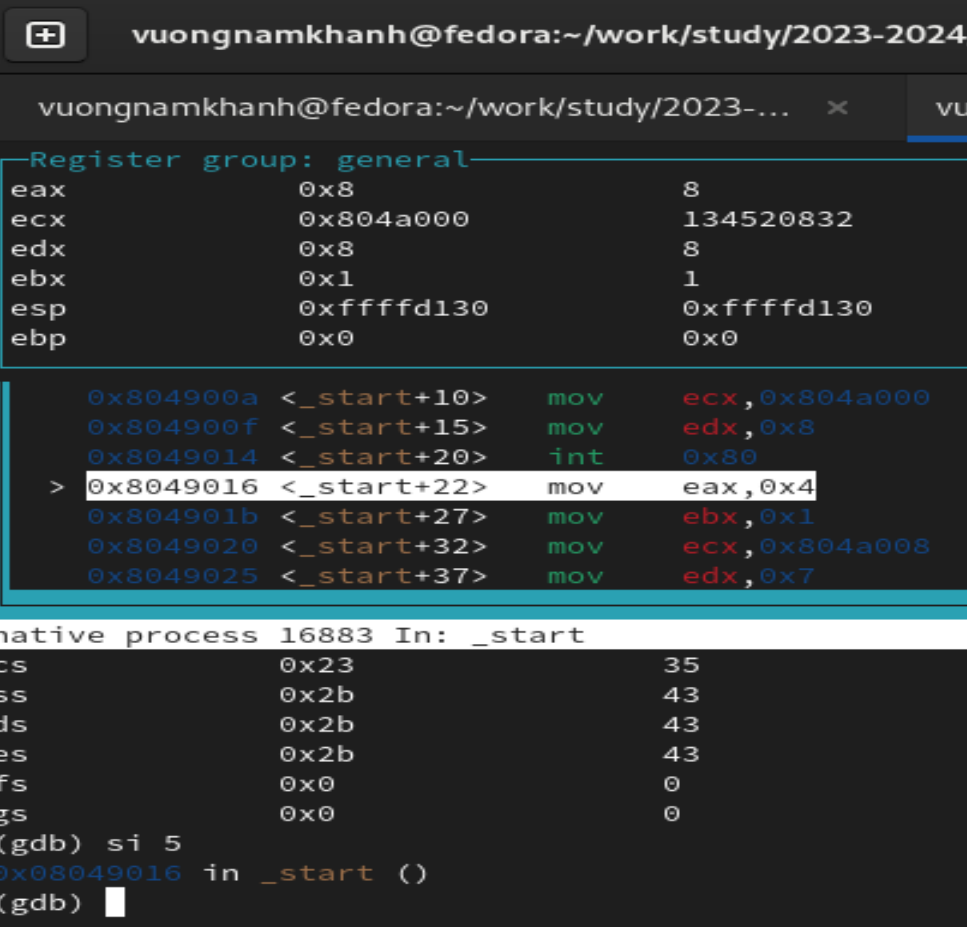
Проверяю, что точка останова по имени метки \_start установлена с помощью команды info breakpoints и устанавливаю еще одну точку останова по адресу инструкции mov ebx,0x0. Просматриваю информацию о всех установленных точках останова.



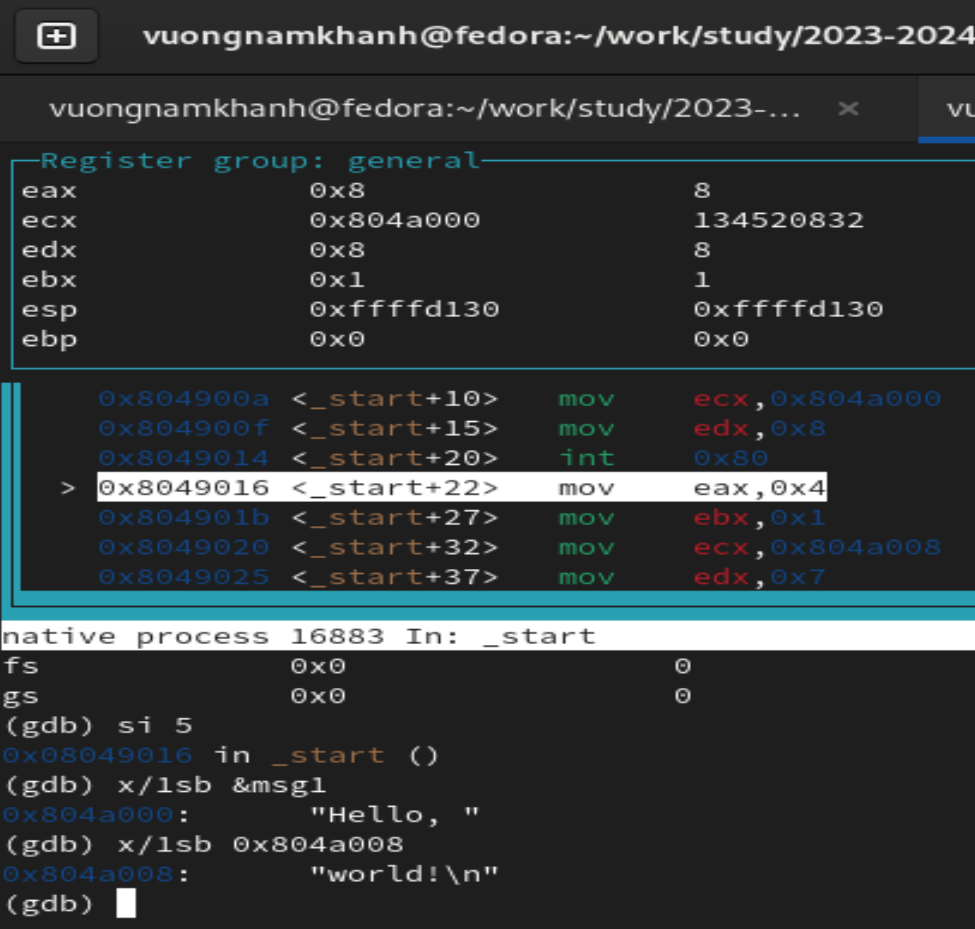
**Работа с данными программы в GDB.**

Выполняю 5 инструкций с помощью команды stepi и слежу за изменением значений регистров.

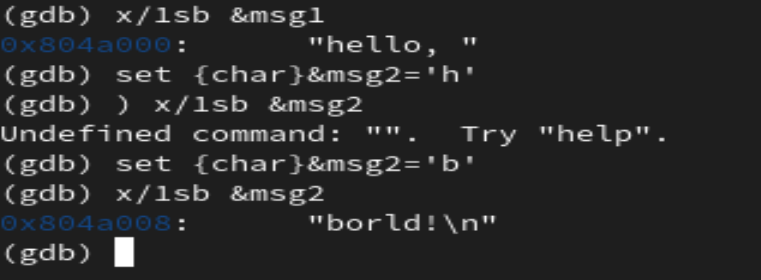




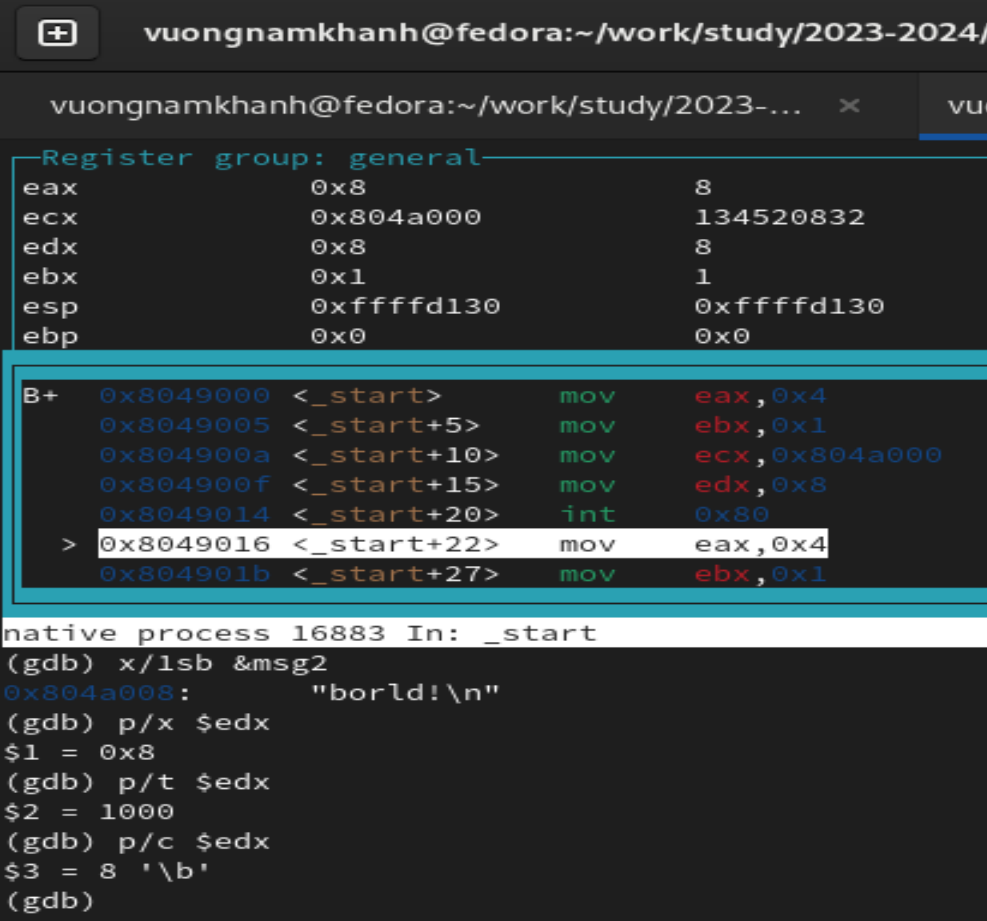
Просматриваю значение переменной msg1 по имени с помощью команды x/1sb &msg1 и значение переменной msg2 по ее адресу



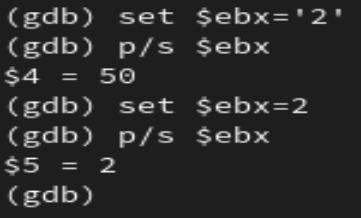
С помощью команды set изменяю первый символ переменной msg1 и заменяю первый символ в переменной msg2.



Вывожу в шестнадцатеричном формате, в двоичном формате и в символьном виде соответственно значение регистра edx с помощью команды print p/F $val.



С помощью команды set изменяю значение регистра ebx в соответствии с заданием.



Разница вывода команд p/s $ebx отличается тем, что в первом случае мы переводим символ в его строковый вид, а во втором случае число в строковом виде не изменяется.

Завершаю выполнение программы с помощью команды continue и выхожу из GDB с помощью команды quit.

