Лабораторная работа №10

Понятие подпрограммы. Отладчик GDB.

Киньябаева Аиша Иделевна

Содержание

# 1 Цель работы

Целью работы является приобретение навыков написания программ с использованием подпрограмм. Знакомство с методами отладки при помощи GDB и его основными возможно- стями.

# 2 Задание

Научиться отладке программы с помощью отладчика GDB и изучение подпрограмм.

# 3 Выполнение лабораторной работы

Пишу программу с использованием с подпрограммы для вычисления выражения: 2x + 7 (рис. 1), (рис. 2)

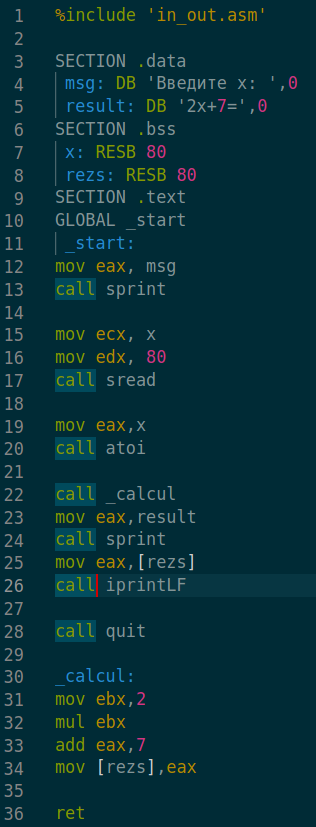


Рис. 1: код

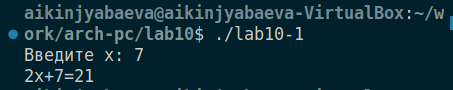


Рис. 2: lab10-1.asm\_вывод

Меняем программу, чтобы она вычисляла: f(g(x)) = 2(3x - 1) + 7 (рис. [#fig:fig3]), (рис. 3), (рис. 5)

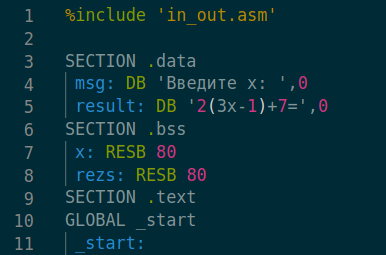


Рис. 3: код

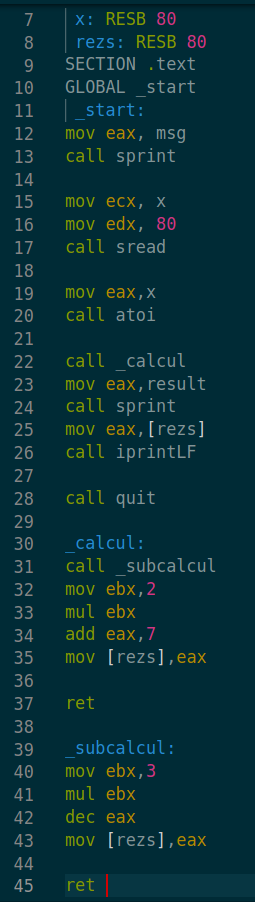


Рис. 4: код

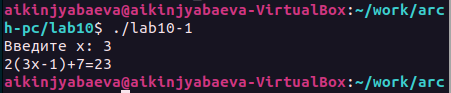


Рис. 5: lab10-1.asm\_вывод

Создаем программу для вывода надписи: Hello world, компилируем листинг файл с отладочной информацией, проверяем с помощью дебаггера GDB.(рис. 6), (рис. 7)

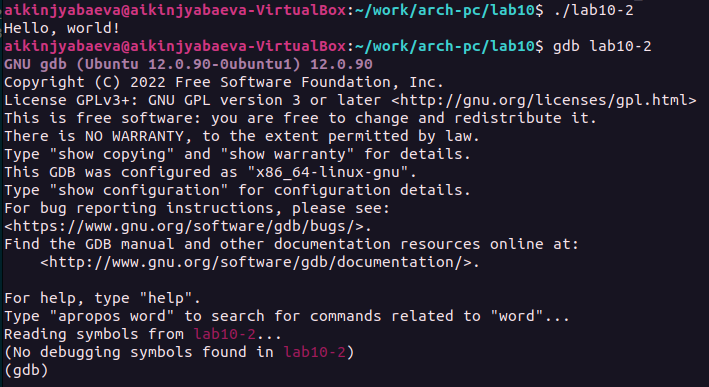


Рис. 6: GDB

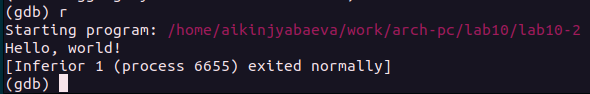


Рис. 7: проверка через run

Устанавливаем break на метку \_start, смотрим работу программы и далее запускаем диссамилированный код. Переключаемся на Intel’овский синтаксис. Различия синтаксисов заключается в разном отображении аргументов и значений(в Intel без доп. символов), а так же изменен порядок их отображения(рис. 8), (рис. 9)

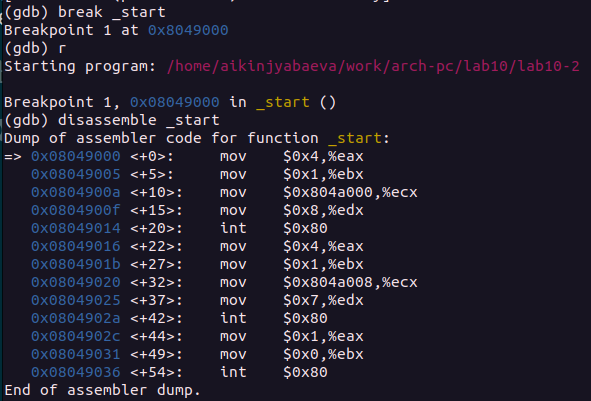


Рис. 8: break point



Рис. 9: Синтаксис Intel

Два режима псевдографики(рис. 10), (рис. 11)

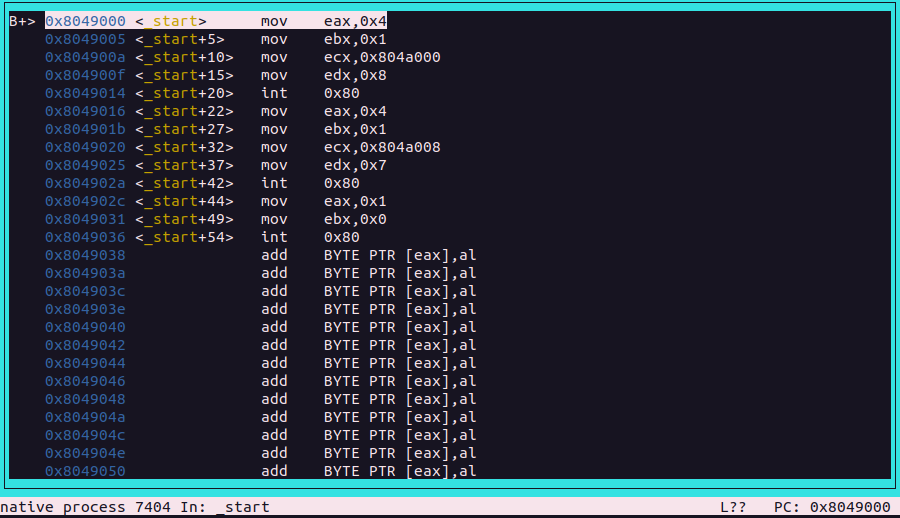


Рис. 10: layout asm

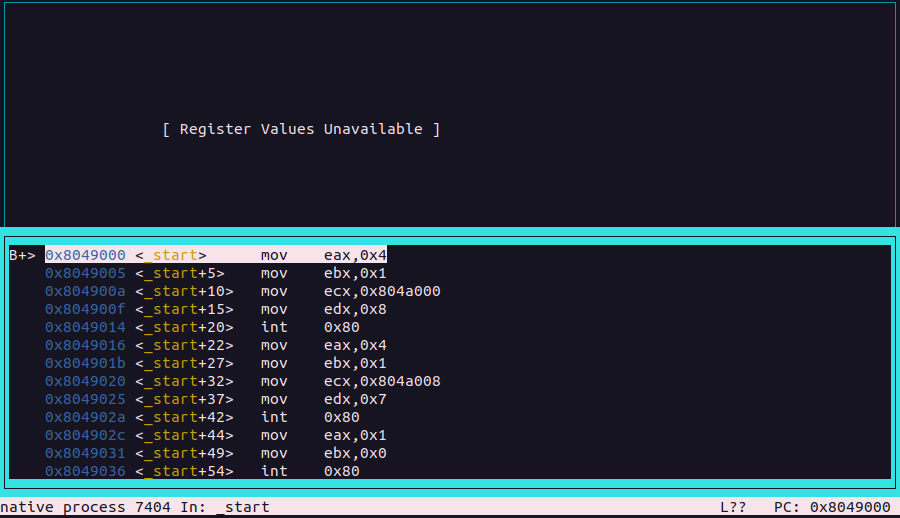


Рис. 11: layout rex

В режиме дебаггинга проверяем точки остановы, и видим, что можем установить еще одну, используя адрес инструкции (рис. 12)

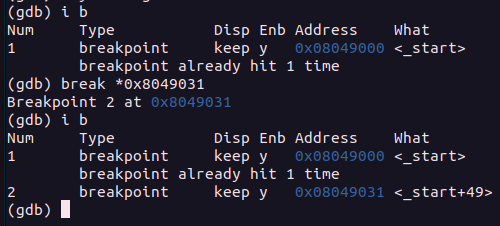


Рис. 12: info breakpoints

Проверяем значения регистров(рис. 13)

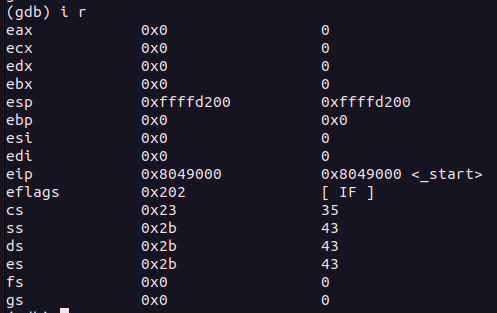


Рис. 13: info registers

Провела 5 инструкций с помощью команды si, меняются значения регистров: eax, ebx, ecx, edx. (рис. 14)

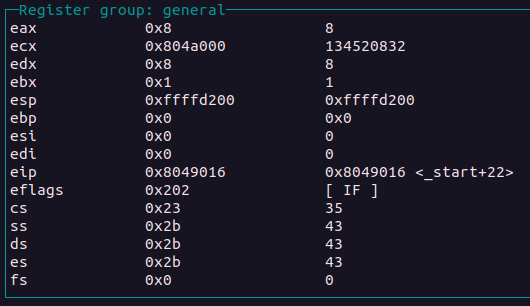


Рис. 14: si 5

Видим, что с помощью различных команд можно смотреть содержимое переменных(в конце скриншота просмотрено значение переменной msg2) (рис. 15)

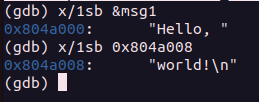


Рис. 15: x/1sb

А с помощью команды set можно менять значение регистра(в конце скриншота изменен символ второй переменной msg2)(рис. 16)

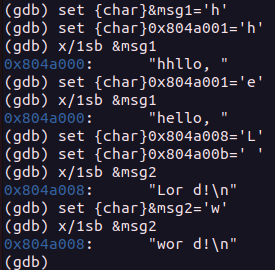


Рис. 16: set

Изучаем использование команды print /F, которое смотрит значение регистров(можно регулировать формат отображения). Вывела в различных форматах значение регистра edx(16ричных, 2чный и символьный вид соотвественно)(рис. 17)

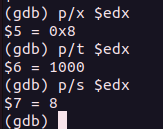


Рис. 17: значение edx

С помощью команды set, меняем значение регистра ebx (рис. 18)

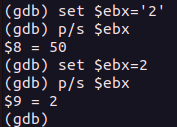


Рис. 18: значение ebx

Указываем аргументы файла в отладчик и далее смотрим позиции аргумента в стеке. Заметим, что [esp + 24] не выводит значения, так как у нас нет 6го аргумента. Шаг изменения адреса равен 4, потому что память имеет 16ричный вид, и 2^4=16.(рис. 19)

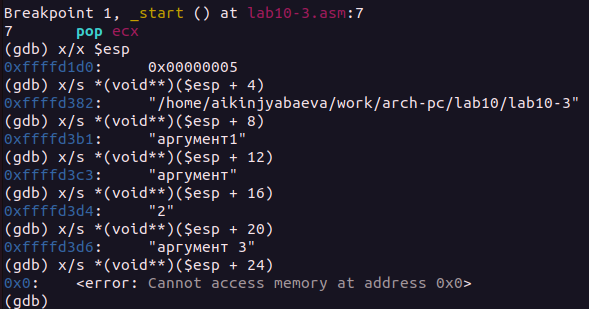


Рис. 19: аргументы

САМОСТОЯТЕЛЬНАЯ РАБОТА

1.Меняю программу из 9 лаб. работы, которая вычисляла 2(x-1) с использованием подпрограмм (рис. 20), (рис. 21)

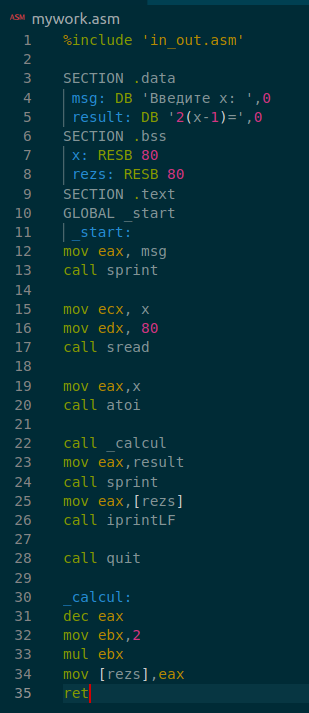


Рис. 20: код

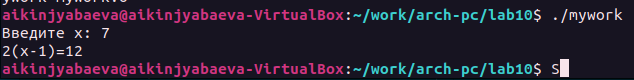


Рис. 21: 2(х-1) вывод

2.Находим ошибка в листинге, для этого используем GDB, смотрим значение регистров. Просматриваю каждый шаг программы с помощью si (рис. 22), (рис. 23), (рис. 24), (рис. 25), (рис. 26), (рис. 27), (рис. 28)

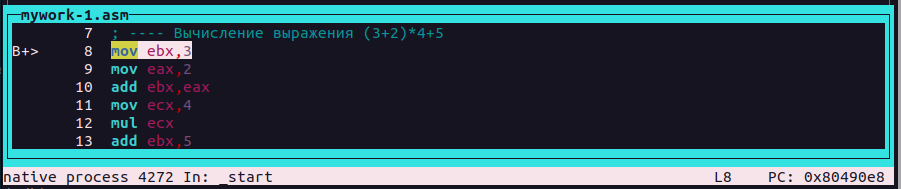


Рис. 22: проверка программы

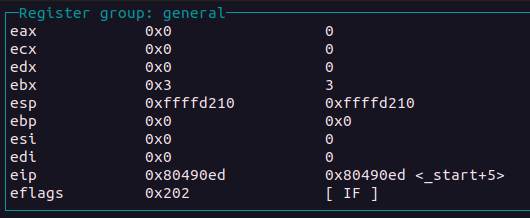


Рис. 23: проверка программы

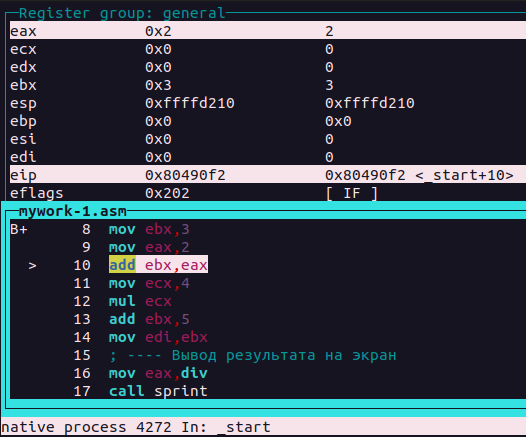


Рис. 24: проверка программы

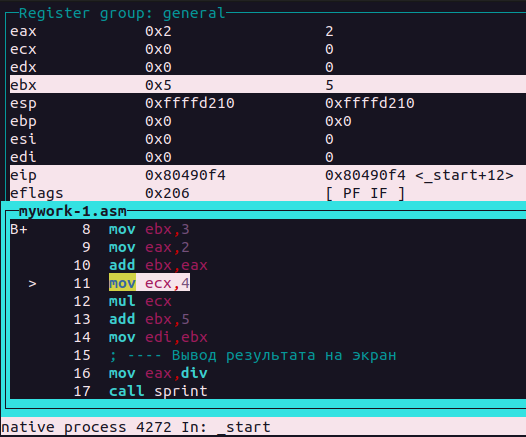


Рис. 25: проверка программы

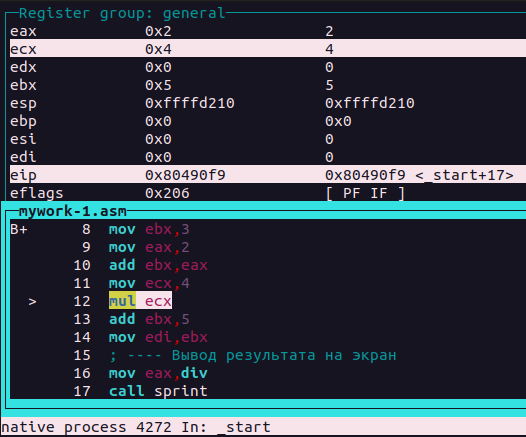


Рис. 26: проверка программы

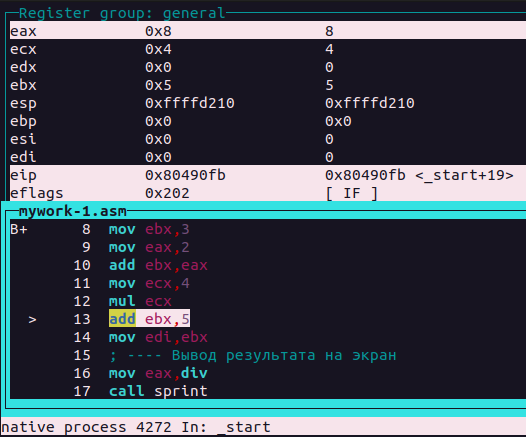


Рис. 27: проверка программы

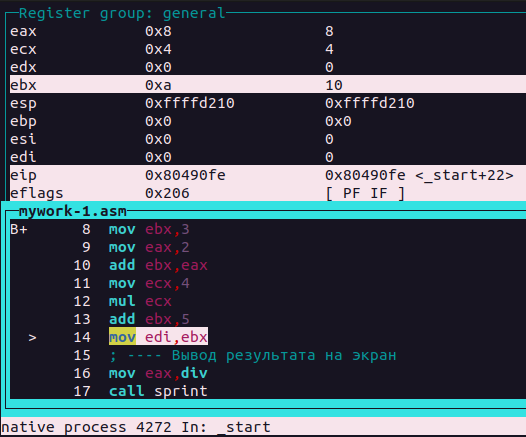


Рис. 28: проверка программы

Видим, что ошибка в том, что программа умножает на 4 двойку из eax. Исправляем программу в редакторе. (рис. 29), (рис. 30)

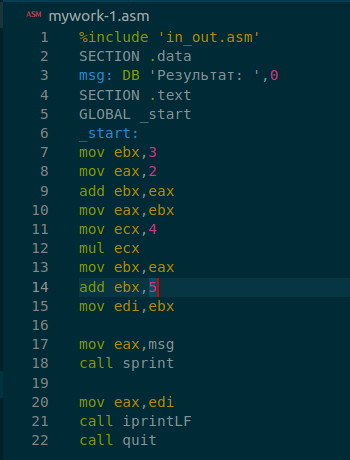


Рис. 29: код

Рис. 30: (3 + 2) * 4 + 5 вывод

Рис. 30: (3 + 2) \* 4 + 5 вывод

Загрузка всех файлов на Git.

Далее создается отчет по 10й лабораторной работе с помощью Markdown.

# 4 Выводы

В ходе данной лабораторной работы были изучены подпрограммы, освнен дебагинг с помощью GDB.