

0. Лабораторная работа №9. Понятие подпрограммы. Отладчик GDB

Архитектура компьютеров

Акрур Имад НКАбд-06-24

Содержание

1	Цель работы	5
2	Описание результатов выполнения лабораторной работы:	6
2.1	Выполнение лабораторной работы	6
3	Описание результатов выполнения заданий для самостоятельной работы:	22
3.1	Описание выполняемого задания	22
4	Выводы	28

Список иллюстраций

2.1	Программа в файле lab9-1.asm	7
2.2	Запуск программы lab9-1.asm	8
2.3	Программа в файле lab9-1.asm	8
2.4	Запуск программы lab9-1.asm	9
2.5	Программа в файле lab9-2.asm	10
2.6	Запуск программы lab9-2.asm в отладчике	11
2.7	Дизассемблированный код	12
2.8	Дизассемблированный код в режиме интел	13
2.9	Точка остановки	14
2.10	Изменение регистров	15
2.11	Изменение регистров	16
2.12	Изменение значения переменной	17
2.13	Вывод значения регистра	18
2.14	Вывод значения регистра	19
2.15	Вывод значения регистра	21
3.1	Программа в файле prog-1.asm	22
3.2	Запуск программы prog-1.asm	23
3.3	Код с ошибкой	24
3.4	Отладка	25
3.5	Код исправлен	26
3.6	Проверка работы	27

Список таблиц

1 Цель работы

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

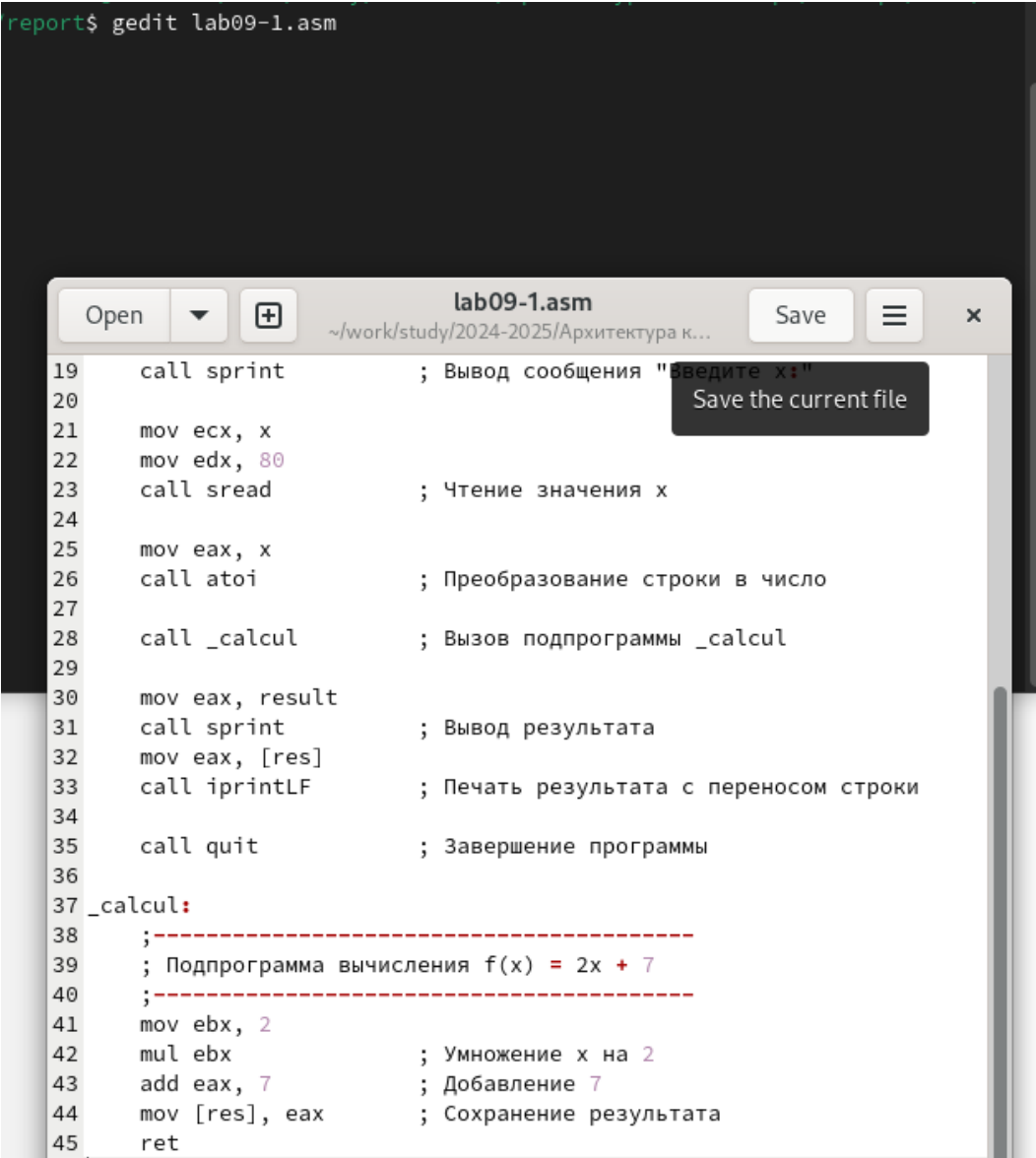
2 Описание результатов выполнения лабораторной работы:

2.1 Выполнение лабораторной работы

Я создал каталог для выполнения лабораторной работы № 9 и перешел в него. Затем я создал файл lab9-1.asm.

В качестве примера рассмотрим программу вычисления арифметического выражения $f(x) = 2x + 7$ с помощью подпрограммы calcul. В данном примере x вводится с клавиатуры, а само выражение вычисляется в подпрограмме.(рис. 2.1) (рис. 2.2)

```
report$ gedit lab09-1.asm
```



```
19  call sprint          ; Вывод сообщения "Введите x:"
20
21  mov ecx, x
22  mov edx, 80
23  call sread           ; Чтение значения x
24
25  mov eax, x
26  call atoi            ; Преобразование строки в число
27
28  call _calcul          ; Вызов подпрограммы _calcul
29
30  mov eax, result
31  call sprint           ; Вывод результата
32  mov eax, [res]
33  call iprintLF         ; Печать результата с переносом строки
34
35  call quit            ; Завершение программы
36
37 _calcul:
38  ; -----
39  ; Подпрограмма вычисления f(x) = 2x + 7
40  ; -----
41  mov ebx, 2
42  mul ebx               ; Умножение x на 2
43  add eax, 7            ; Добавление 7
44  mov [res], eax        ; Сохранение результата
45  ret
```

Рис. 2.1: Программа в файле lab9-1.asm

```
imadakrour@fedora:~/work/study/2024-2025/Архитектура ко...
imadakrour@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab09/report$ nasm -f elf32 lab9-1.asm
imadakrour@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab09/report$ ld -m elf_i386 -o lab9-1 lab9-1.o
imadakrour@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab09/report$ ./lab9-1
Введите x: 2
2x+7=11
```

Рис. 2.2: Запуск программы lab9-1.asm

Изменил 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. 2.3) (рис. 2.4)

```
36
37 _calcul:
38 ;-----
39 ; Подпрограмма вычисления f(g(x))
40 ;-----
41 push eax          ; Сохранение значения x
42 call _subcalcul   ; Вызов подпрограммы g(x)
43 mov eax, [res]    ; Получение результата g(x)
44 mov ebx, 2
45 mul ebx           ; Умножение результата g(x) на 2
46 add eax, 7        ; Добавление 7
47 mov [res], eax    ; Сохранение результата f(g(x))
48 ret
49
50 _subcalcul:
51 ;-----
52 ; Подпрограмма вычисления g(x) = 3x - 1
53 ;-----
54 mov ebx, 3
55 mul ebx           ; Умножение x на 3
56 sub eax, 1        ; Вычитание 1
57 mov [res], eax    ; Сохранение результата g(x)
58 ret
59
```

Рис. 2.3: Программа в файле lab9-1.asm

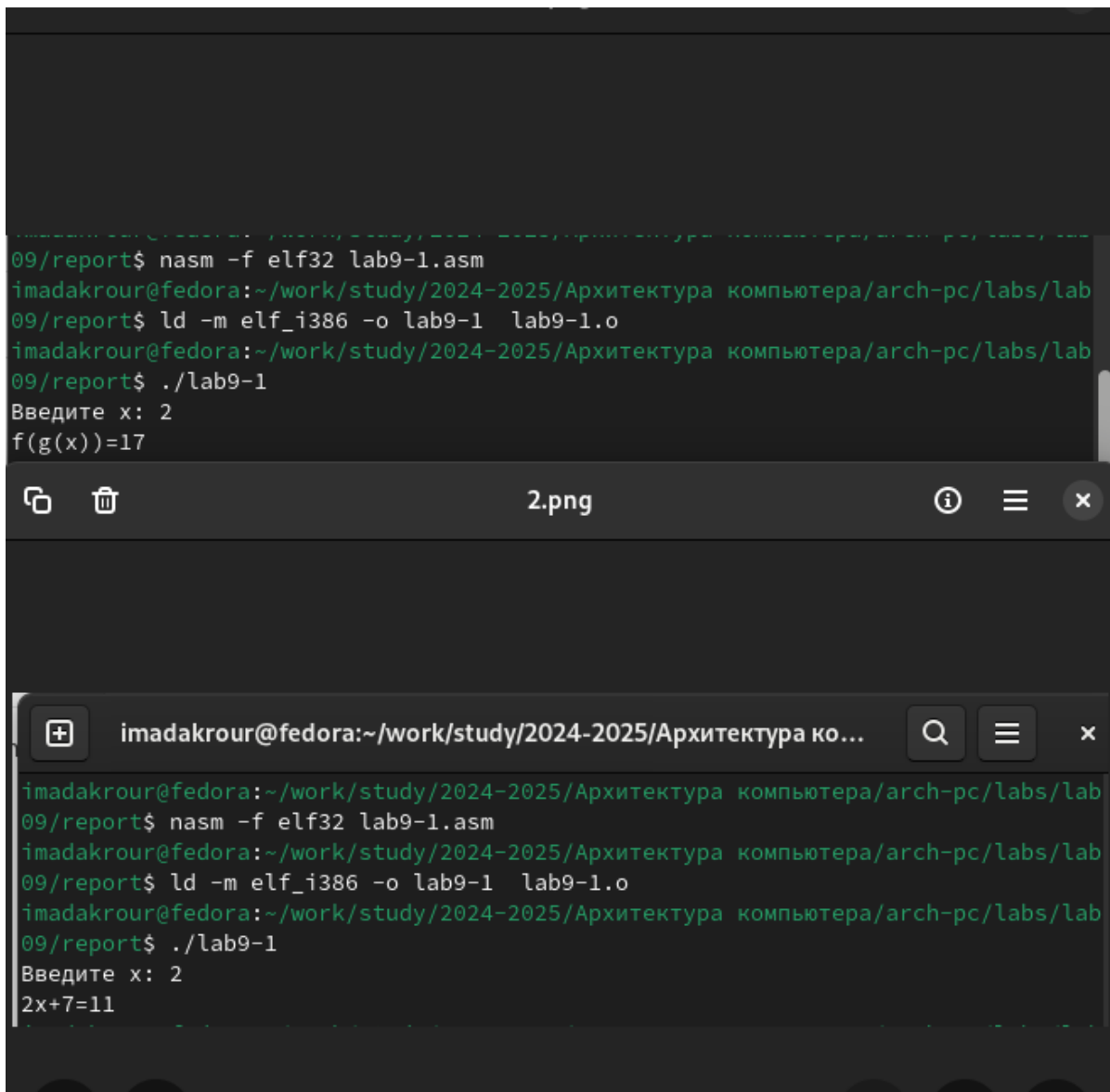



Рис. 2.4: Запуск программы lab9-1.asm

Создал файл lab9-2.asm с текстом программы из Листинга 9.2. (Программа печати сообщения Hello world!). (рис. 2.5)



```
SECTION .data
msg1: db "Hello, ",0x0
msg1len: equ $ - msg1
msg2: db "world!",0xa
msg2len: equ $ - msg2

SECTION .text
global _start

_start:|
mov eax, 4
mov ebx, 1
mov ecx, msg1
mov edx, msg1len
int 0x80
mov eax, 4
mov ebx, 1
mov ecx, msg2
mov edx, msg2len
int 0x80
```

Рис. 2.5: Программа в файле lab9-2.asm

Получил исполняемый файл и добавил отладочную информацию с помощью ключа ‘-g’ для работы с GDB.

Загрузил исполняемый файл в отладчик GDB и проверил работу программы, запустив ее с помощью команды ‘run’ (сокращенно ‘r’). (рис. 2.6)

```
imadakrour@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab
09/report$ nasm -f elf -g -l lab9-2.lst lab9-2.asm
imadakrour@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab
09/report$ ld -m elf_i386 -o lab9-2 lab9-2.o
imadakrour@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab
09/report$ gdb lab9-2
GNU gdb (Fedora Linux) 15.1-1.fc40
Copyright (C) 2024 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-redhat-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from lab9-2...
(gdb) █
```

Рис. 2.6: Запуск программы lab9-2.asm в отладчике

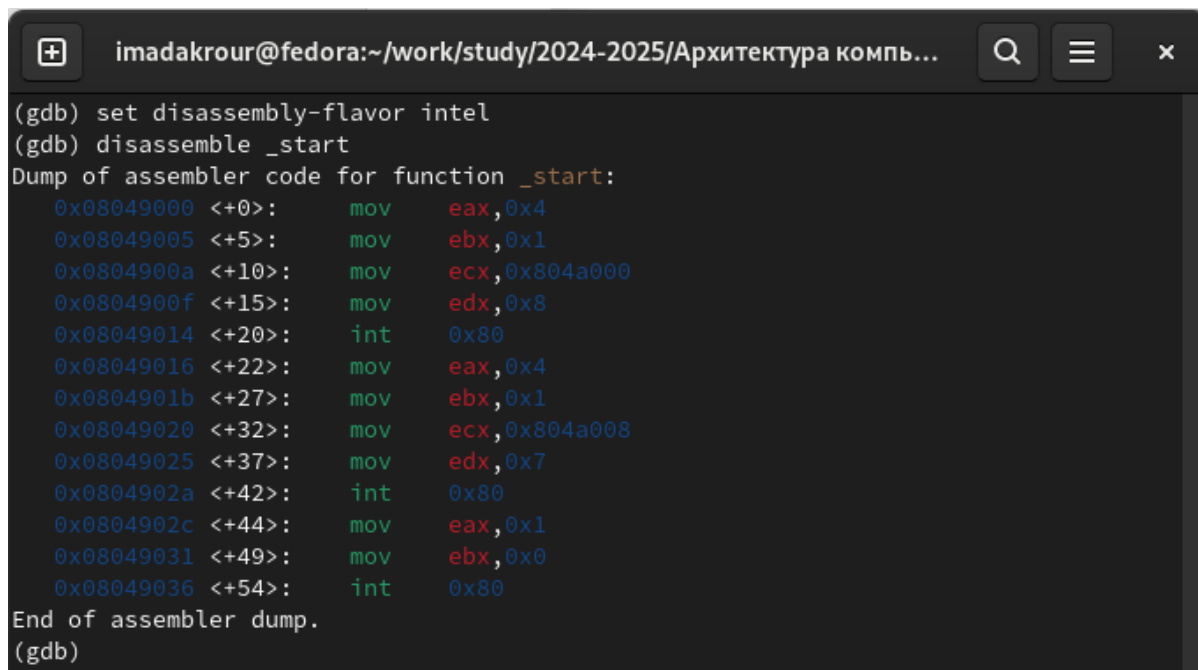
Для более подробного анализа программы, установил точку остановки на метке 'start', с которой начинается выполнение любой ассемблерной программы, и запустил ее. Затем просмотрел дизассемблированный код программы.(рис. 2.7) (рис. 2.8)

```
imadakrour@fedora:~/work/study/2024-2025/Архитектура компь...
report$ gdb lab9-2
GNU gdb (Fedora Linux) 15.1-1.fc40
Copyright (C) 2024 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-redhat-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from lab9-2...
(gdb) run
Starting program: /home/imadakrour/work/study/2024-2025/Архитектура компьютера/arch-
pc/labs/lab09/report/lab9-2

This GDB supports auto-downloading debuginfo from the following URLs:
<https://debuginfod.fedoraproject.org/>
Enable debuginfod for this session? (y or [n]) n
Debuginfod has been disabled.
To make this setting permanent, add 'set debuginfod enabled off' to .gdbinit.
Hello, world!
[Inferior 1 (process 11649) exited normally]
(gdb) break _start
Breakpoint 1 at 0x8049000: file lab9-2.asm, line 11.
(gdb) disassemble _start
Dump of assembler code for function _start:
0x08049000 <+0>:    mov     $0x4,%eax
0x08049005 <+5>:    mov     $0x1,%ebx
0x0804900a <+10>:   mov     $0x804a000,%ecx
0x0804900f <+15>:   mov     $0x8,%edx
0x08049014 <+20>:   int     $0x80
0x08049016 <+22>:   mov     $0x4,%eax
0x0804901b <+27>:   mov     $0x1,%ebx
0x08049020 <+32>:   mov     $0x804a008,%ecx
0x08049025 <+37>:   mov     $0x7,%edx
0x0804902a <+42>:   int     $0x80
0x0804902c <+44>:   mov     $0x1,%eax
0x08049031 <+49>:   mov     $0x0,%ebx
0x08049036 <+54>:   int     $0x80
End of assembler dump.
(gdb) 
```

Рис. 2.7: Дизассемблированный код



```
imadakrour@fedora:~/work/study/2024-2025/Архитектура компь...
(gdb) set disassembly-flavor intel
(gdb) disassemble _start
Dump of assembler code for function _start:
0x08049000 <+0>:    mov     eax,0x4
0x08049005 <+5>:    mov     ebx,0x1
0x0804900a <+10>:   mov     ecx,0x804a000
0x0804900f <+15>:   mov     edx,0x8
0x08049014 <+20>:   int     0x80
0x08049016 <+22>:   mov     eax,0x4
0x0804901b <+27>:   mov     ebx,0x1
0x08049020 <+32>:   mov     ecx,0x804a008
0x08049025 <+37>:   mov     edx,0x7
0x0804902a <+42>:   int     0x80
0x0804902c <+44>:   mov     eax,0x1
0x08049031 <+49>:   mov     ebx,0x0
0x08049036 <+54>:   int     0x80
End of assembler dump.
(gdb)
```

Рис. 2.8: Дизассемблированный код в режиме интел

Для проверки точки останова по имени метки '_start', использовал команду 'info breakpoints' (сокращенно 'i b'). Затем установил еще одну точку останова по адресу инструкции, определив адрес предпоследней инструкции 'mov ebx, 0x0'. (рис. 2.9)

The screenshot shows a GDB terminal window with the title bar "imadakrour@fedora:~/work/study/2024-2025/Архитектура ко...". The window is divided into two main sections. The top section, titled "Register group: general", displays the values of several registers: `eax` (0x0), `ecx` (0x0), `edx` (0x0), `ebx` (0x0), `esp` (0xffffcf80), and `ebp` (0x0). The bottom section shows a list of instructions at a breakpoint, with the first instruction highlighted: `B+>0x8049000 <_start> mov eax,0x4`. Other instructions include `0x8049005 <_start+5> mov ebx,0x1`, `0x804900a <_start+10> mov ecx,0x804a000`, `0x804900f <_start+15> mov edx,0x8`, `0x8049014 <_start+20> int 0x80`, and `0x8049016 <_start+22> mov eax,0x4`. At the bottom of the window, the status bar indicates "native process 11861 (asm) In: _start", "L11", and "PC: 0x8049000". The command prompt shows `(gdb) layout regs` and `(gdb)`.

```
imadakrour@fedora:~/work/study/2024-2025/Архитектура ко...
Register group: general
eax      0x0      0
ecx      0x0      0
edx      0x0      0
ebx      0x0      0
esp      0xffffcf80 0xffffcf80
ebp      0x0      0x0

B+>0x8049000 <_start>  mov  eax,0x4
0x8049005 <_start+5>   mov  ebx,0x1
0x804900a <_start+10>  mov  ecx,0x804a000
0x804900f <_start+15>  mov  edx,0x8
0x8049014 <_start+20>  int  0x80
0x8049016 <_start+22>  mov  eax,0x4

native process 11861 (asm) In: _start          L11  PC: 0x8049000
(gdb) layout regs
(gdb)
```

Рис. 2.9: Точка остановки

В отладчике GDB можно просматривать содержимое ячеек памяти и регистров, а также изменять значения регистров и переменных. Выполнил 5 инструкций с помощью команды 'stepi' (сокращенно 'si') и отследил изменение значений регистров. (рис. 2.10) (рис. 2.11)

```
imadakrour@fedora:~/work/study/2024-2025/Архитектура ко...
Register group: general
eax      0x0      0
ecx      0x0      0
edx      0x0      0
ebx      0x0      0
esp      0xffffcf80 0xffffcf80
ebp      0x0      0x0

B+>0x8049000 <_start>    mov    $0x4,%eax
0x8049005 <_start+5>    mov    $0x1,%ebx
0x804900a <_start+10>   mov    $0x804a000,%ecx
0x804900f <_start+15>   mov    $0x8,%edx
0x8049014 <_start+20>   int    $0x80
0x8049016 <_start+22>   mov    $0x4,%eax

native process 18247 (asm) In: _start          L11  PC: 0x8049000
(gdb) layout asm
(gdb) layout regs
(gdb) 
```

Рис. 2.10: Изменение регистров

```
imadakrour@fedora:~/work/study/2024-2025/Архитектура ко...
Register group: general
eax      0x8      8
ecx      0x804a000 134520832
edx      0x8      8
ebx      0x1      1
esp      0xffffcf80 0xffffcf80
ebp      0x0      0x0

0x804900a <_start+10> mov $0x804a000,%ecx
0x804900f <_start+15> mov $0x8,%edx
0x8049014 <_start+20> int $0x80
>0x8049016 <_start+22> mov $0x4,%eax
0x804901b <_start+27> mov $0x1,%ebx
0x8049020 <_start+32> mov $0x804a008,%ecx

native process 18247 (asm) In: _start L16 PC: 0x8049016
(gdb) layout asm
(gdb) layout regs
(gdb) si
(gdb) si
(gdb) si
(gdb) si
(gdb) si
(gdb) si
(gdb)
```

Рис. 2.11: Изменение регистров

Просмотрел значение переменной msg1 по имени и получил нужные данные.
Просмотрел значение переменной msg1 по имени и получил нужные данные.
Для изменения значения регистра или ячейки памяти использовал команду set, указав имя регистра или адрес в качестве аргумента. Изменил первый символ переменной msg1. (рис. 2.12)


```
imadakrour@fedora:~/work/study/2024-2025/Архитектура ко...
Register group: general
eax      0x8      8
ecx      0x804a000 134520832
edx      0x8      8
ebx      0x1      1
esp      0xffffcf80 0xffffcf80
ebp      0x0      0x0
esi      0x0      0
edi      0x0      0
eip      0x8049016 0x8049016 <_start+22>

0x804900a <_start+10> mov ecx,0x804a000
0x804900f <_start+15> mov edx,0x8
0x8049014 <_start+20> int 0x80
>0x8049016 <_start+22> mov eax,0x4
0x804901b <_start+27> mov ebx,0x1
0x8049020 <_start+32> mov ecx,0x804a008
0x8049025 <_start+37> mov edx,0x7
0x804902a <_start+42> int 0x80
0x804902c <_start+44> mov eax,0x1

native process 11861 (asm) In: _start L16 PC: 0x8049016
(gdb) set {char}&msg1='H'
(gdb) x/1sb &msg1
0x804a000 <msg1>: "Hello, "
(gdb) set {char}&msg1='h'
(gdb) x/1sb &msg1
0x804a000 <msg1>: "hello, "
(gdb) set {char}0x804a008
warning: Expression is not an assignment (and might have no effect)
(gdb) set {char}0x804a008='L'
(gdb) x/1sb 0x804a008
0x804a008 <msg2>: "Lor!d!\n\034"
(gdb)
```

Рис. 2.12: Изменение значения переменной

Для изменения значения регистра или ячейки памяти использовал команду set, указав имя регистра или адрес в качестве аргумента. Изменил первый символ переменной msg1.(рис. 2.13)

```
imadakrour@fedora:~/work/study/2024-2025/Архитектура ко...
Register group: general
eax      0x8      8
ecx      0x804a000 134520832
edx      0x8      8
ebx      0x1      1
esp      0xffffcf80 0xffffcf80
ebp      0x0      0x0
esi      0x0      0
edi      0x0      0
eip      0x8049016 0x8049016 <_start+22>
eflags   0x202    [ IF ]
cs       0x23     35
ss       0x2b     43

B+ 0x8049000 <_start>    mov     eax,0x4
    0x8049005 <_start+5>  mov     ebx,0x1
    0x804900a <_start+10> mov     ecx,0x804a000
    0x804900f <_start+15> mov     edx,0x8
    0x8049014 <_start+20> int      0x80
>0x8049016 <_start+22>  mov     eax,0x4
    0x804901b <_start+27> mov     ebx,0x1
    0x8049020 <_start+32> mov     ecx,0x804a008
    0x8049025 <_start+37> mov     edx,0x7
    0x804902a <_start+42> int      0x80
    0x804902c <_start+44> mov     eax,0x1
b+ 0x8049031 <_start+49> mov     ebx,0x0
    0x8049036 <_start+54> int      0x80

native process 12040 (asm) In: _start L16 PC: 0x8049016
(gdb) p/s $eax
$1 = 8
(gdb) p/t &eax
No symbol "eax" in current context.
(gdb) p/t $eax
$2 = 1000
(gdb) p/s $ecx
$3 = 134520832
(gdb) p/x $ecx
$4 = 0x804a000
(gdb) p/s $edx
$5 = 8
(gdb) p/t $edx
$6 = 1000
(gdb) 
```

Рис. 2.13: Вывод значения регистра

С помощью команды set изменил значение регистра ebx на нужное значение.
(рис. 2.14)

```
imadakrou@fedora:~/work/study/2024-2025/Архитектура ко...
-Register group: general-
eax      0x8      8
ecx      0x804a000 134520832
edx      0x8      8
ebx      0x2      2
esp      0xffffcf80 0xffffcf80
ebp      0x0      0x0
esi      0x0      0
edi      0x0      0
eip      0x8049016 0x8049016 <_start+22>
eflags   0x202    [ IF ]
cs       0x23     35
ss       0x2b     43

B+ 0x8049000 <_start>    mov     eax,0x4
0x8049005 <_start+5>    mov     ebx,0x1
0x804900a <_start+10>   mov     ecx,0x804a000
0x804900f <_start+15>   mov     edx,0x8
0x8049014 <_start+20>   int     0x80
>0x8049016 <_start+22>   mov     eax,0x4
0x804901b <_start+27>   mov     ebx,0x1
0x8049020 <_start+32>   mov     ecx,0x804a008
0x8049025 <_start+37>   mov     edx,0x7
0x804902a <_start+42>   int     0x80
0x804902c <_start+44>   mov     eax,0x1
b+ 0x8049031 <_start+49> mov     ebx,0x0
0x8049036 <_start+54>   int     0x80

native process 12040 (asm) In: _start L16 PC: 0x8049016
(gdb) p/t $edx
$6 = 1000
(gdb) p/x $edx
$7 = 0x8
(gdb) set $ebx='2'
(gdb) p/s $eax
$8 = 8
(gdb) p/t $eax
$9 = 1000
(gdb) p/s $ecx
$10 = 134520832
(gdb) set $ebx=2
(gdb) p/s $ebx
$11 = 2
(gdb)
```

Рис. 2.14: Вывод значения регистра

Скопировал файл lab8-2.asm, созданный во время выполнения лабораторной работы №8, который содержит программу для вывода аргументов командной строки. Создал исполняемый файл из скопированного файла.

Для загрузки программы с аргументами в gdb использовал ключ -args и загрузил исполняемый файл в отладчик с указанными аргументами.

Установил точку останова перед первой инструкцией программы и запустил ее.

Адрес вершины стека, содержащий количество аргументов командной строки (включая имя программы), хранится в регистре `esp`. По этому адресу находится число, указывающее количество аргументов. В данном случае видно, что количество аргументов равно 5, включая имя программы `lab9-3` и сами аргументы: `аргумент1`, `аргумент2` и `'аргумент 3'`.

Просмотрел остальные позиции стека. По адресу `[esp+4]` находится адрес в памяти, где располагается имя программы. По адресу `[esp+8]` хранится адрес первого аргумента, по адресу `[esp+12]` - второго и так далее. (рис. 2.15)

```
imadakrour@fedora:~/work/study/2024-2025/Архитектура ко...
09/report$ nasm -f elf -g -l lab9-3.lst lab9-3.asm
imadakrour@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab
09/report$ ld -m elf_i386 -o lab9-3 lab9-3.o
imadakrour@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab
09/report$ gdb --args lab9-3 arg1 arg 2 'argument 3'
GNU gdb (Fedora Linux) 15.1-1.fc40
Copyright (C) 2024 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-redhat-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
  <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from lab9-3...
(gdb) b_start
Undefined command: "b_start". Try "help".
(gdb) b _start
Breakpoint 1 at 0x8049000: file lab9-3.asm, line 11.
(gdb) run
Starting program: /home/imadakrour/work/study/2024-2025/Архитектура компьютера/a
rch-pc/labs/lab09/report/lab9-3 arg1 arg 2 argument\ 3

This GDB supports auto-downloading debuginfo from the following URLs:
  <https://debuginfod.fedoraproject.org/>
Enable debuginfod for this session? (y or [n]) y
Debuginfod has been enabled.
To make this setting permanent, add 'set debuginfod enabled on' to .gdbinit.

Breakpoint 1, _start () at lab9-3.asm:11
11      mov eax, 4
(gdb) |
```

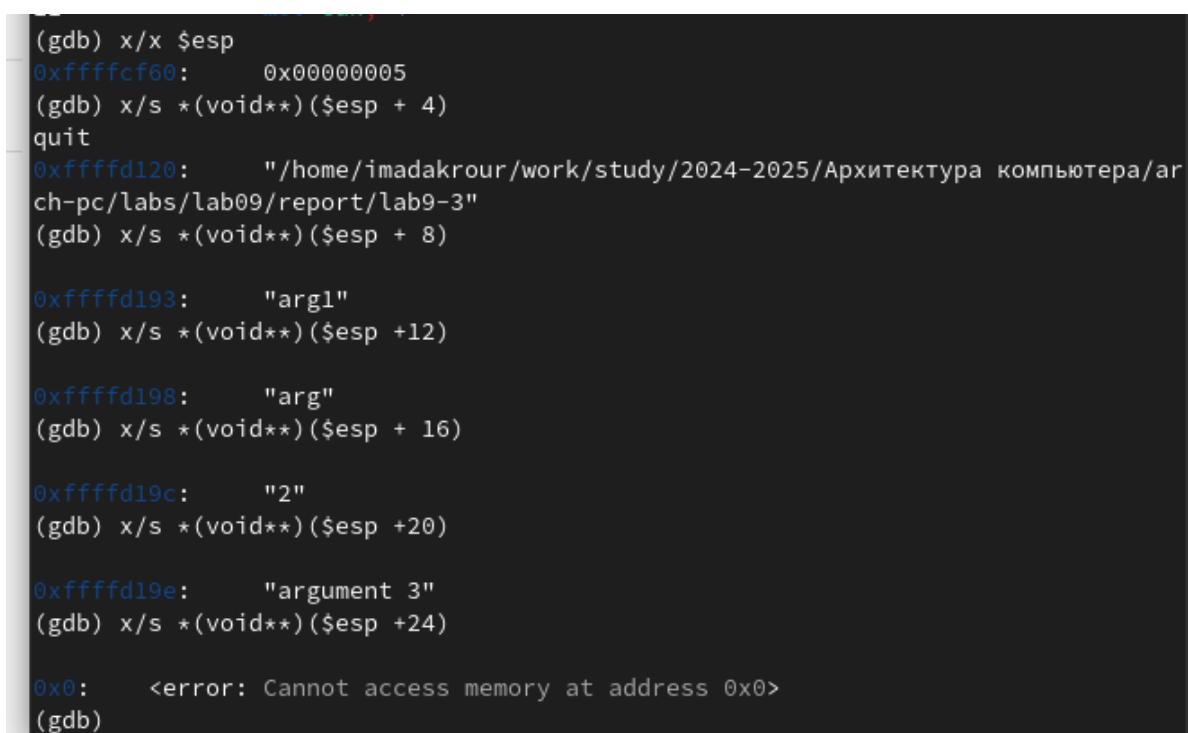
Рис. 2.15: Вывод значения регистра

Шаг изменения адреса равен 4, так как каждый следующий адрес на стеке находится на расстоянии 4 байт от предыдущего ([esp+4], [esp+8], [esp+12]).

3 Описание результатов выполнения заданий для самостоятельной работы:

3.1 Описание выполняемого задания

Преобразовал программу из лабораторной работы №8 (Задание №1 для самостоятельной работы), реализовав вычисление значения функции $f(x)$ как подпрограмму. (рис. 3.1) (рис. 3.2)



```
(gdb) x/x $esp
0xffffcf60: 0x00000005
(gdb) x/s *(void**)(esp + 4)
quit
0xffffd120: "/home/imadakrour/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab09/report/lab9-3"
(gdb) x/s *(void**)(esp + 8)

0xffffd193: "arg1"
(gdb) x/s *(void**)(esp + 12)

0xffffd198: "arg"
(gdb) x/s *(void**)(esp + 16)

0xffffd19c: "2"
(gdb) x/s *(void**)(esp + 20)

0xffffd19e: "argument 3"
(gdb) x/s *(void**)(esp + 24)

0x0: <error: Cannot access memory at address 0x0>
(gdb)
```

Рис. 3.1: Программа в файле prog-1.asm

The image shows a terminal window and a text editor window. The terminal window has a title bar with the text "imadakrour@fedora:~/work/study/2024-2025/Архитектура ко...". The terminal content shows the following commands and output:

```
imadakrour@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab09/report$ touch cam.asm
imadakrour@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab09/report$ gedit cam.asm
```

The text editor window has a title bar with the text "*cam.asm" and a file path "~/work/study/2024-2025/Архитектура к...". The editor contains the following assembly code:

```
1 %include 'in_out.asm'
2 SECTION .data
3 msg db "Результат: ",0
4 fx: db 'f(x)= 15x + 2',0
5
6 SECTION .text
7 global _start
8 _start:
9 mov eax, fx
10 call sprintLF
11 pop ecx
12 pop edx
13 sub ecx,1
14 mov esi, 0
15
16 next:
17 cmp ecx,0h
18 jz _end
19 pop eax
20 call atoi
21 call _funk
22 add esi,eax
23
24 loop next
25
26 _end:
27 mov eax, msg
28 call sprint
```

The status bar at the bottom of the text editor shows "Plain Text", "Tab Width: 8", "Ln 38, Col 1", and "INS".

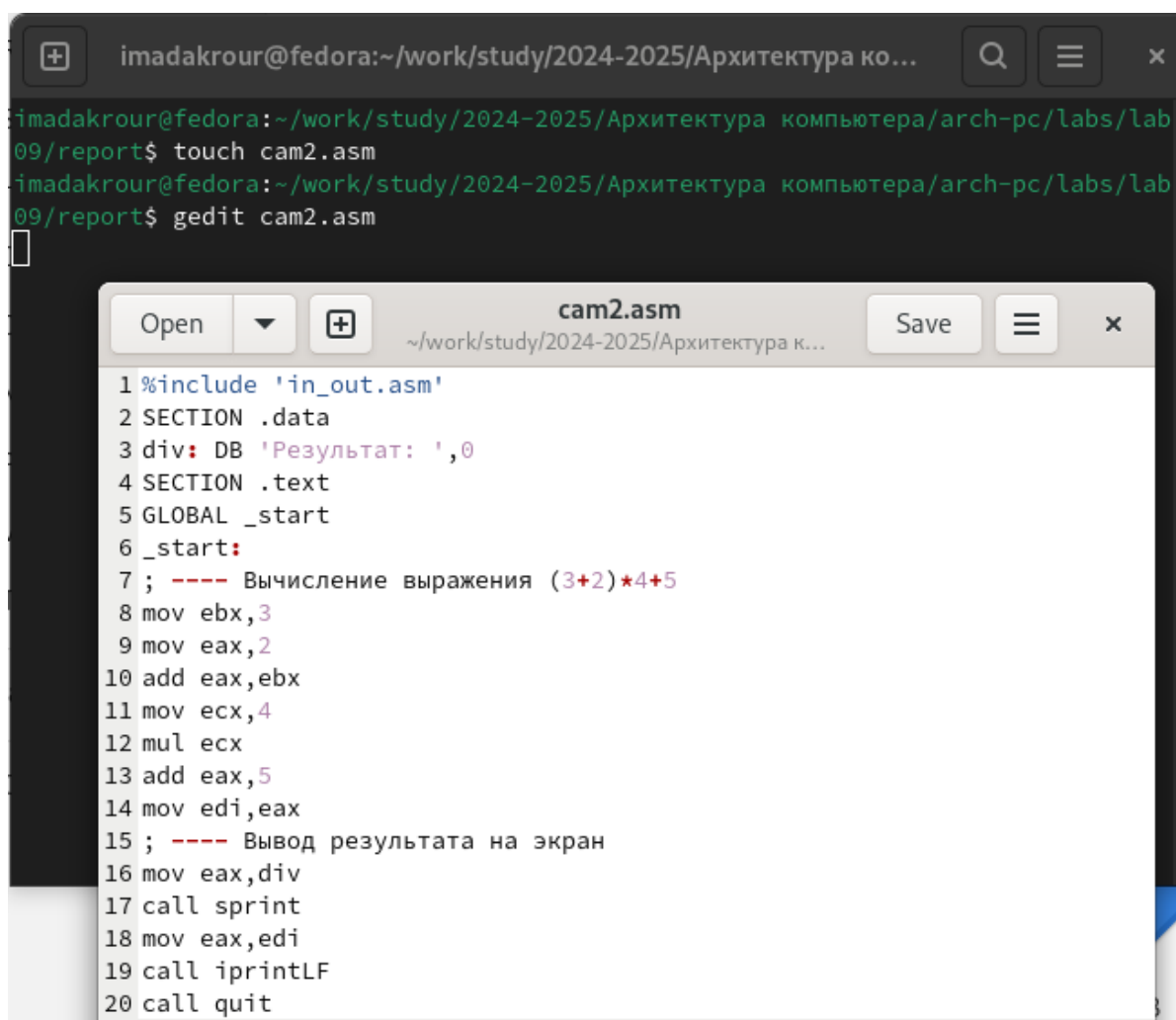
Рис. 3.2: Запуск программы prog-1.asm

В листинге приведена программа вычисления выражения $(3 + 2) * 4 + 5$. При запуске данная программа дает неверный результат. Проверил это, анализируя изменения значений регистров с помощью отладчика GDB.

Определил ошибку - перепутан порядок аргументов у инструкции add. Также обнаружил, что по окончании работы в edi отправляется ebx вместо eax.(рис. 3.3)

```
imadakrour@fedora:~/work/study/2024-2025/Архитектура ко...
imadakrour@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab09/report$ touch cam.asm
imadakrour@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab09/report$ gedit cam.asm
imadakrour@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab09/report$ nasm -f elf cam.asm
imadakrour@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab09/report$ ld -m elf_i386 cam.o -o cam
imadakrour@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab09/report$ ./cam
f(x)= 15x + 2
Результат: 0
imadakrour@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab09/report$ ./cam 6 6 4 1 2 3
f(x)= 15x + 2
Результат: 342
```

Рис. 3.3: Код с ошибкой



The image shows a terminal window and a code editor window. The terminal window has a title bar with the text "imadakrour@fedora:~/work/study/2024-2025/Архитектура ко...". The terminal content shows the following commands and output:

```
imadakrour@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab09/report$ touch cam2.asm
imadakrour@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab09/report$ gedit cam2.asm
```

The code editor window has a title bar with the text "cam2.asm" and the path "~/work/study/2024-2025/Архитектура к...". The code editor content shows the following assembly code:

```
1 %include 'in_out.asm'
2 SECTION .data
3 div: DB 'Результат: ',0
4 SECTION .text
5 GLOBAL _start
6 _start:
7 ; ---- Вычисление выражения (3+2)*4+5
8 mov ebx,3
9 mov eax,2
10 add eax,ebx
11 mov ecx,4
12 mul ecx
13 add eax,5
14 mov edi,eax
15 ; ---- Вывод результата на экран
16 mov eax,div
17 call sprint
18 mov eax,edi
19 call iprintLF
20 call quit
```

Рис. 3.4: Отладка

Отмечу, что перепутан порядок аргументов у инструкции `add` и что по окончании работы в `edi` отправляется `ebx` вместо `eax` (рис. 3.4)

Исправленный код программы (рис. 3.5) (рис. 3.6)

```
imadakrour@fedora:~/work/study/2024-2025/Архитектура ко...
eax 134520832

ffffcf[ Register Values Unavailable ]

0x8049105 <_start+29> call 0x804900f <sprint>
0x804910a <_start+34> mov $0di,%eax,%eax
>0x804910c <_start+36> call 0x8049086 <iprintLF>
0x8049111 <_start+41> call 0x80490db <quit>
                                <iprintLF>

native process 14629 (asm) In: _start L17 PC: 0x8049105
(gdb) sNo process (asm) In: L?? PC: ??
(gdb) si
(gdb) si
(gdb) si
(gdb) c
Continuing.
Результат: 13
[Inferior 1 (process 14629) exited normally]
(gdb)
```

Рис. 3.5: Код исправлен



```
div: DB 'Результат: ',0
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения (3+2)*4+5
mov ebx,3
mov eax,2
add eax,ebx
mov ecx,4
mul ecx
add eax,5
mov edi,eax
; ---- Вывод результата на экран
mov eax,div
call sprint
mov eax,edi
call iprintLF
call quit
```

Рис. 3.6: Проверка работы

4 Выводы

Освоили работу с подпрограммами и отладчиком.