

Отчет по лабораторной работе №9

Простейший вариант

Чигладзе Майя Владиславовна

Содержание

1	Цель работы	5
2	Порядок выполнения лабораторной работы	6
2.1	Реализация подпрограмм в NASM	6
2.1.1	Задание 1	6
2.1.2	Задание 2	6
2.2	Отладка программы с помощью GDB	9
2.2.1	Задание 0	9
2.2.2	Задание 1	12
2.2.3	Задание 2	12
2.2.4	Задание 3	15
3	Задание для самостоятельной работы	17
3.1	Задание 1	17
3.2	Задание 2	18
4	Выводы	21
	Список литературы	22

Список иллюстраций

2.1	Создание каталога	6
2.2	Работа программы	7
2.3	Код программы	8
2.4	Работа программы	9
2.5	Работа программы	10
2.6	Работа программы	11
2.7	Режимы псевдографики	11
2.8	Брейкпоинтс	12
2.9	Команда si	13
2.10	Команда x/1sb	13
2.11	Команда set	14
2.12	Команда print	15
2.13	Запуск с аргументами	16
2.14	Позиции стека	16
3.1	Преобразование программы	17
3.2	Ошибка	18
3.3	До	19
3.4	После	19
3.5	Исправила	20
3.6	Запустила	20

Список таблиц

1 Цель работы

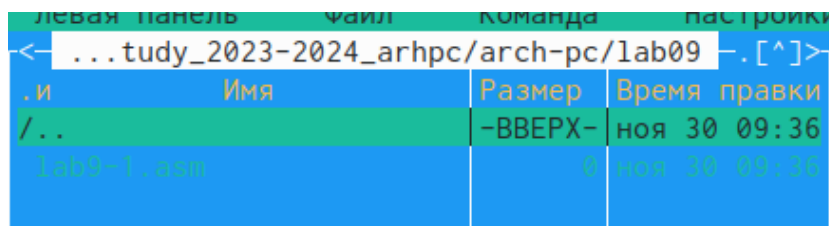
Приобретение навыков написания программ с использованием подпрограмм. Знакомство с методами отладки при помощи GDB и его основными возможностями.

2 Порядок выполнения лабораторной работы

2.1 Реализация подпрограмм в NASM

2.1.1 Задание 1

Создала каталог для программ лабораторной работы № 9, перешла в него и создала файл lab9-1.asm(рис. [2.1])



The screenshot shows a terminal window with a blue background. At the top, there are four tabs: 'левая панель', 'файл', 'команда', and 'настройки'. The 'команда' tab is active, showing a command prompt where the user has entered the path `... tudy_2023-2024_arhpc/arch-pc/lab09`. Below the command prompt, a table lists the contents of the directory. The table has four columns: 'Имя' (Name), 'Размер' (Size), and 'Время правки' (Modification Time). The first row shows the current directory `./` with a size of `-ВВЕРХ-` and a modification time of `ноя 30 09:36`. The second row shows the subdirectory `lab9-1.asm` with a size of `0` and a modification time of `ноя 30 09:36`.

Имя	Размер	Время правки
./	-ВВЕРХ-	ноя 30 09:36
lab9-1.asm	0	ноя 30 09:36

Рис. 2.1: Создание каталога

2.1.2 Задание 2

Введу в файл lab09-1.asm текст программы из листинга 9.1. Создам исполняемый файл и проверю его работу (рис. [2.2])

```
mvchigladze@dk5n52 ~ $ mc
09 $ touch lab9-1.asm
09 $ nasm -f elf lab9-1.asm
mvchigladze@dk5n52 ~/work/study/2023-2024/
09 $ ld -m elf_i386 -o lab9-1 lab9-1.o
mvchigladze@dk5n52 ~/work/study/2023-2024/
09 $ ./lab9-1
Введите x: 19
2x+7=45
mvchigladze@dk5n52 ~/work/study/2023-2024/
09 $
```

Рис. 2.2: Работа программы

Изменяю текст программы, добавив подпрограмму `_subcalcul` в подпрограмму `_calcul`, для вычисления выражения $f(g(x))$, где x вводится с клавиатуры, $f(x) = 2x + 7$, $g(x) = 3x - 1$. Т.е. x передается в подпрограмму `_calcul` из нее в подпрограмму `_subcalcul`, где вычисляется выражение $g(x)$, результат возвращается в `_calcul` и вычисляется выражение $f(g(x))$. Результат возвращается в основную программу для вывода результата на экран. Код моей программы (рис. [2.3]) и результат выполнения(рис. [2.4]).

```

lab9-1.asm      [-M--]  3 L:[ 16+27  43/ 47] *
mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi

call _calcul ; Вызов подпрограммы _calcul
mov eax, result
call sprint
mov eax, [res]
call iprintLF
call quit
;-----
; Подпрограмма вычисления
; выражения "2x+7"
_calcul:
call _subcalcul
mov ebx, 2
mul ebx
add eax, 7
mov [res], eax
ret ; выход из подпрограммы
;-----
_subcalcul:
mov ebx, 3
mul ebx
sub eax, 1
ret

```

Рис. 2.3: Код программы


```
mvchigladze@dk5n52 ~/work/study/2023-202
09 $ nasm -f elf lab9-1.asm
mvchigladze@dk5n52 ~/work/study/2023-202
09 $ ld -m elf_i386 -o lab9-1 lab9-1.o
mvchigladze@dk5n52 ~/work/study/2023-202
09 $ ./lab9-1
Введите x: 1
 $2(3x-1)+7=11$ 
mvchigladze@dk5n52 ~/work/study/2023-202
09 $
```

Рис. 2.4: Работа программы

2.2 Отладка программы с помощью GDB

2.2.1 Задание 0

Введу в файл lab09-2.asm текст программы из листинга 9.2. Создам исполняемый файл и проверю его работу с использованием команды run (рис. [2.5])

```

09 $ nasm -f elf -g -l lab09-2.lst lab09-2.asm
nasm: fatal: unable to open input file 'lab09-2.asm' No such file or directory
mvchigladze@dk5n52 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arhp
09 $ nasm -f elf -g -l lab9-2.lst lab9-2.asm
mvchigladze@dk5n52 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arhp
09 $ ld -m elf_i386 -o lab9-2 lab9-2.o
mvchigladze@dk5n52 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arhp
09 $ gdb lab9-2
GNU gdb (Gentoo 12.1 vanilla) 12.1
Copyright (C) 2022 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-pc-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://bugs.gentoo.org/>.
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) r
Starting program: /afs/.dk.sci.pfu.edu.ru/home/m/v/mvchigladze/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arhpc/arch-pc/lab09/lab9-2
Hello, world!
[Inferior 1 (process 6127) exited normally]

```

Рис. 2.5: Работа программы

Посмотрю дисассимилированный код программы с помощью команды `disassemble` начиная с метки `_start` (`gdb`) `disassemble _start`. Переключу на отображение команд с Intel'овским синтаксисом, введя команду `set disassembly-flavor intel` (рис. [2.6]). Наверное, главное отличие, это то что в АТТ выводятся переменные, в которые все записывается, а в Intel размер переменных.

```

(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) 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

```

Рис. 2.6: Работа программы

Включите режим псевдографики для более удобного анализа программы (рис. [2.7])

```

[ Register Values Unavailable ]

0x8049264    add     BYTE PTR [eax],al
0x8049266    add     BYTE PTR [eax],al
0x8049268    add     BYTE PTR [eax],al
0x804926a    add     BYTE PTR [eax],al
0x804926c    add     BYTE PTR [eax],al
0x804926e    add     BYTE PTR [eax],al
0x8049270    add     BYTE PTR [eax],al
0x8049272    add     BYTE PTR [eax],al
0x8049274    add     BYTE PTR [eax],al

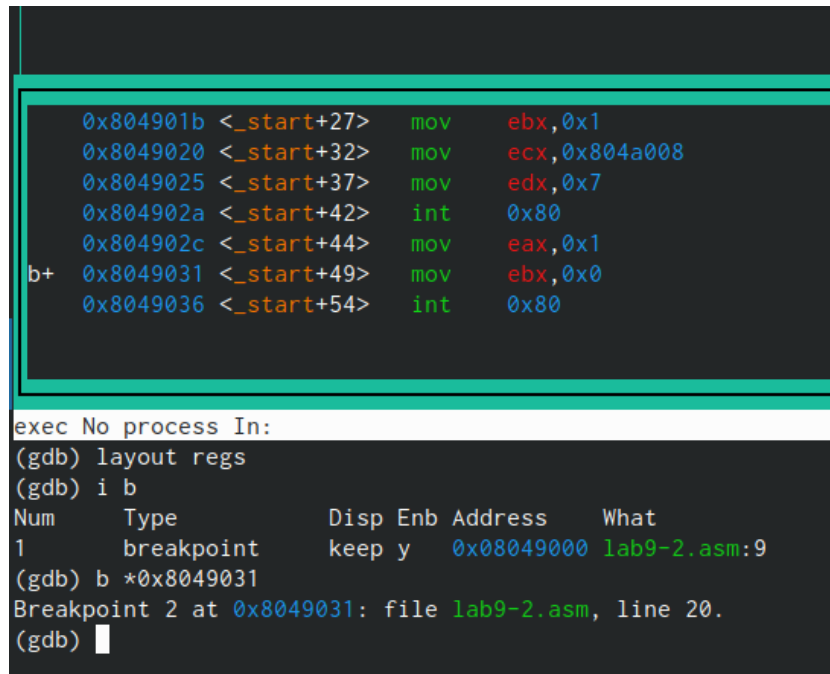
native process 6132 In: _start      L9    PC: 0x8049000
(gdb) layout regs
(gdb)

```

Рис. 2.7: Режим псевдографики

2.2.2 Задание 1

Проверила текущие брейкпоинты и установила новый на предпоследней инструкции (рис. [2.8])



```
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

exec No process In:
(gdb) layout regs
(gdb) i b
Num      Type          Disp Enb Address      What
1        breakpoint    keep y  0x08049000  lab9-2.asm:9
(gdb) b *0x8049031
Breakpoint 2 at 0x8049031: file lab9-2.asm, line 20.
(gdb) 
```

Рис. 2.8: Брейкпоинтс

2.2.3 Задание 2

Используя команда si 5 раз,я наблюдала изменение значений первых 4-ех регистров eax ecx edx ebx (рис. [2.9])

```
Register group: general
eax      0x8      8
ecx      0x804a000 134520832
edx      0x8      8
ebx      0x1      1
esp      0xffffc0e0 0xffffc0e0
ebp      0x0      0x0
esi      0x0      0
edi      0x0      0

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

native process 4245 In: _start
(gdb) run
Starting program: /afs/.dk.sci.pfu.edu.ru/home/m/v/mvchigladze/ура компьютера/study_2023-2024_arhpc/arch-pc/lab09/lab9-2

Breakpoint 1, _start () at lab9-2.asm:9
(gdb) si
(gdb) si
(gdb) si
(gdb) si
(gdb) si
(gdb) si
(gdb) 
```

Рис. 2.9: Команда si

Смотрим значение переменной по имени и адресу (рис. [2.10])

```
ура компьютера/study_2023-2024_arhpc/arch-pc/lab09/lab9-2
Hello, world!
(gdb) x/1sb &msg1
0x804a000 <msg1>:      "Hello, "
(gdb) x/1sb 0x804a008
0x804a008 <msg2>:      "world!\n"<err
```

Рис. 2.10: Команда x/1sb

Замена символов в переменных (рис. [2.11])

```
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
0x804901b <_start+27>    mov     $0x1,%ebx
0x8049020 <_start+32>    mov     $0x804a008,%ecx
0x8049025 <_start+37>    mov     $0x7,%edx

native No process in:
(gdb) set {char}msg1='h'
'msg1' has unknown type; cast it to its declared type
(gdb) x/1sb &msg1
0x804a000 <msg1>:      "Hello, "
(gdb) set {char}0x804a008='L'
Cannot access memory at address 0x804a008
(gdb) set {char}0x804a00b=' '
Cannot access memory at address 0x804a00b
(gdb) set {char}0x804900a='L'
Cannot access memory at address 0x804900a
(gdb) 
```

Рис. 2.11: Команда set

Использование команды print (рис. [2.12]). Используя команду p/s \$ebx мы получим значение регистра 1.

```

ebp      0x0      0
esi      0x0      0
edi      0x0      0
B+ 0x8049000 <_start>    mov    $0x4,%eax
    0x8049005 <_start+5>  mov    $0x1,%ebx0,%ecx
    0x804900a <_start+10> mov    $0x804a000,%ecx
    0x804900f <_start+15> mov    $0x8,%edx
    0x8049014 <_start+20> int     $0x80%eax
> 0x8049016 <_start+22>  mov    $0x4,%eax
    0x8049020 <_start+32> mov    $0x804a008,%ecx
    0x8049025 <_start+37> mov    $0x7,%edx

native No process In:
Cannot process 4639 In: _start$ 0x804a00b
(gdb) p/s $eax
$1 = 8
(gdb) p/t $eax
$2 = 1000
(gdb) p/s $ecx
$3 = 134520832
(gdb) p/x &ecx
No symbol "ecx" in current context.
(gdb) p/x $ecx
$4 = 0x804a000
(gdb) 
```

Рис. 2.12: Команда print

2.2.4 Задание 3

Скопировала второй файл из 8 лабораторной работы в папку, создала исполняемый файл и запустили с аргументами (рис. [2.13]).

```

mvchigladze@dk2n26 ~/work/study/2023-2024/Архитектура компьютера/
pc/lab09 $ cp ~/work/study/2023-2024/Архитектура\ компьютера/stu
ab08/lab8-2.asm ~/work/study/2023-2024/Архитектура\ компьютера/s
/lab09/lab9-3.asm
mvchigladze@dk2n26 ~/work/study/2023-2024/Архитектура компьютера/
pc/lab09 $ nasm -f elf -g -l lab9-3.lst lab9-3.asm
mvchigladze@dk2n26 ~/work/study/2023-2024/Архитектура компьютера/
pc/lab09 $ ld -m elf_i386 -o lab9-3 lab9-3.o
mvchigladze@dk2n26 ~/work/study/2023-2024/Архитектура компьютера/
pc/lab09 $ gdb --args lab9-3 2 3 '4'
GNU gdb (Gentoo 13.2 vanilla) 13.2
Copyright (C) 2023 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-pc-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://bugs.gentoo.org/>.
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...

```

Рис. 2.13: Запуск с аргументами

Смотрю что находится в переменных с разным адресом (рис. [2.14]). Шаг изменения адреса равен 4 из-за того сколько памяти выделяют на переменную.

```

уря компьютера/study_2023-2024_arhpc/arch-pc/lab09/lab9-3 2 3 4
Breakpoint 1, _start () at lab9-3.asm:5
5      pop ecx ; Извлекаем из стека в `ecx` количество
(gdb) x/x $esp
0xfffffc0e0: 0x00000004
(gdb) x/s *(void**)( $esp+4)
0xfffffc373: "/afs/.dk.sci.pfu.edu.ru/home/m/v/mvchigladze/work/s
ра компьютера/study_2023-2024_arhpc/arch-pc/lab09/lab9-3"
(gdb) x/s *(void**)( $esp + 8)
0xfffffc40c: "2"
(gdb) x/s *(void**)( $esp + 20)
0x0: <error: Cannot access memory at address 0x0>
(gdb) x/s *(void**)( $esp + 12)
0xfffffc40e: "3"
(gdb) x/s *(void**)( $esp + 16)
0xfffffc410: "4"
(gdb) x/s *(void**)( $esp + 24)
0xfffffc412: "SHELL=/bin/bash"
(gdb)

```

Рис. 2.14: Позиции стека

3 Задание для самостоятельной работы

3.1 Задание 1

Преобразовала программу из лабораторной работы №8 (Задание №1 для самостоятельной работы), реализовав вычисление значения функции $\boxtimes(\boxtimes)$ как подпрограмму(рис. [3.1]).

```
mvchigladze@dk3n35 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024
_arhpc/arch-pc/lab09 $ nasm -f elf -g -l lab9-4.lst lab9-4.asm
mvchigladze@dk3n35 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024
_arhpc/arch-pc/lab09 $ ld -m elf_i386 -o lab9-4 lab9-4.o
mvchigladze@dk3n35 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024
_arhpc/arch-pc/lab09 $ gdb --args lab9-4 1 2 3
GNU gdb (Gentoo 13.2 vanilla) 13.2
Copyright (C) 2023 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-pc-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://bugs.gentoo.org/>.
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 lab9-4...
(gdb) b _start
Breakpoint 1 at 0x80490e8: file lab9-4.asm, line 8.
(gdb) run
Starting program: /afs/.dk.sci.pfu.edu.ru/home/m/v/mvchigladze/work/study/2023-2
024/Архитектура компьютера/study_2023-2024_arhpc/arch-pc/lab09/lab9-4 1 2 3
Breakpoint 1, _start () at lab9-4.asm:8
```

Рис. 3.1: Преобразование программы

3.2 Задание 2

В листинге 9.3 приведена программа вычисления выражения $(3 + 2) \times 4 + 5$. При запуске данная программа дает неверный результат. Проверю это, выводит 10, должно 25 (рис. [3.2]).

```
mvchigladze@dk3n35 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024
_arhpc/arch-pc/lab09 $ touch lab9-5.asm
mvchigladze@dk3n35 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024
_arhpc/arch-pc/lab09 $ nasm -f elf lab9-5.asm
mvchigladze@dk3n35 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024
_arhpc/arch-pc/lab09 $ ld -m elf_i386 -o lab9-5 lab9-5.o
mvchigladze@dk3n35 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024
_arhpc/arch-pc/lab09 $ ./lab9-5
Результат: 10
mvchigladze@dk3n35 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024
```

Рис. 3.2: Ошибка

С помощью отладчика GDB, анализируя изменения значений регистров, определяю ошибку и исправлю ее. Ошибка была в том, что используя команду `mul`, идет умножение на `eax`, а не то что мы хотим (`ebx`) (рис. [3.3]) (рис. [3.4]). Я исправила программу (рис. [3.5]) и запустила (рис. [3.6]).

```

Register group: general
eax      0x2      2
ecx      0x4      4
edx      0x0      0
ebx      0x5      5
esp      0xffffc060 0xffffc060
ebp      0x0      0x0
esi      0x0      0
edi      0x0      0

B+ 0x80490e8 <_start>    mov     ebx,0x3
0x80490ed <_start+5>    mov     eax,0x2
0x80490f2 <_start+10>   add     ebx,eax
0x80490f4 <_start+12>   mov     ecx,0x4
> 0x80490f9 <_start+17> mul     ecx
0x80490fb <_start+19>   add     ebx,0x5
0x80490fe <_start+22>   mov     edi,ebx
0x8049100 <_start+24>   mov     eax,0x804a000
0x8049105 <_start+29>   call    0x804900f <sprint>

native process 8769 In: _start L12 PC: 0x80490f9
(gdb) r
Starting program: /afs/.dk.sci.pfu.edu.ru/home/m/v/mvchigladze/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arhpc/arch-pc/lab09/lab9-5

Breakpoint 1, _start () at lab9-5.asm:8
(gdb) si
(gdb) si
(gdb) si
(gdb) si

```

Рис. 3.3: До

```

Register group: general
eax      0x8      8
ecx      0x4      4
edx      0x0      0
ebx      0x5      5
esp      0xffffc060 0xffffc060
ebp      0x0      0x0
esi      0x0      0
edi      0x0      0

B+ 0x80490e8 <_start>    mov     ebx,0x3
0x80490ed <_start+5>    mov     eax,0x2
0x80490f2 <_start+10>   add     ebx,eax
0x80490f4 <_start+12>   mov     ecx,0x4
0x80490f9 <_start+17>   mul     ecx
> 0x80490fb <_start+19>   add     ebx,0x5
0x80490fe <_start+22>   mov     edi,ebx
0x8049100 <_start+24>   mov     eax,0x804a000
0x8049105 <_start+29>   call    0x804900f <sprint>

native process 8769 In: _start L13 PC: 0x80490fb
Starting program: /afs/.dk.sci.pfu.edu.ru/home/m/v/mvchigladze/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arhpc/arch-pc/lab09/lab9-5

Breakpoint 1, _start () at lab9-5.asm:8
(gdb) si
(gdb) si
(gdb) si
(gdb) si
(gdb) si

```

Рис. 3.4: После

```

lab9-5.asm      [----] 10 L: [ 1+13 14/ 20] *(232 / 348b) 0120 0x078 [*][X]
#include "nasm.asm"
SECTION .data
div: DB "Результат: ",0
SECTION .text
GLOBAL _start
_start:
; ----- Вычисление выражения (3+2)*4+5
mov eax,3
mov ebx,2
add eax,ebx
mov ecx,4
mul ecx
add eax,5
mov edi,edx
; ----- Вывод результата на экран
mov eax,div
call sprint
mov eax,edi
call iprintLF
call quit

```

Рис. 3.5: Исправила

```

Undefined command: "y". Try "help".
(gdb)
[2]+ Остановлен   gdb --args lab9-5
mvchigladze@dk3n35 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024
_arhpc/arch-pc/lab09 $ nasm -f elf -g -l lab9-5.lst lab9-5.asm
mvchigladze@dk3n35 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024
_arhpc/arch-pc/lab09 $ ld -m elf_i386 -o lab9-5 lab9-5.o
mvchigladze@dk3n35 ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024
_arhpc/arch-pc/lab09 $ gdb --args lab9-5
GNU gdb (Gentoo 13.2 vanilla) 13.2
Copyright (C) 2023 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-pc-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://bugs.gentoo.org/>.
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 lab9-5...
(gdb) r
Starting program: /afs/.dk.sci.pfu.edu.ru/home/m/v/mvchigladze/work/study/2023-2
024/Архитектура компьютера/study_2023-2024_arhpc/arch-pc/lab09/lab9-5
Результат: 25
[Inferior 1 (process 8926) exited normally]

```

Рис. 3.6: Запустила

4 Выводы

В ходе лабораторной работы, я приобрела навыки написания программ с использованием подпрограмм. Познакомилась с методами отладки при помощи GDB и его основными возможностями.

Список литературы