

Отчёт по лабораторной работе”

Простейший вариант

Тииур Ринатович Каримов

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	22
	Список литературы	23

Список иллюстраций

4.1	Создание каталога и файла	8
4.2	Текст программы	8
4.3	Работа программы	9
4.4	Создание каталога и файла	9
4.5	Создание каталога и файла	9
4.6	Текст второй программы	10
4.7	Отладка второго файла	10
4.8	Брежпоинт на метку _start	11
4.9	Дисассимплриованный код	11
4.10	Отображение Intel	12
4.11	Псевдографика	12
4.12	Наличие меток	13
4.13	Прсмотр регистров	13
4.14	Измененные регистры	13
4.15	Просмотр значения переменной msg1	13
4.16	Просмотр значения переменной msg2	14
4.17	Изменение значения переменной msg1	14
4.18	Изменение значения переменной msg2	14
4.19	значения регистров esx и eax	15
4.20	Значения регистров ebx	15
4.21	Завершение работы с файлом	15
4.22	Запуск файла в отладчике	16
4.23	Запуск файла lab9-3.asm через метку	16
4.24	Адрес вершины стека	16
4.25	Все позиции стека	17
4.26	Текст программы	17
4.27	Запуск программы	18
4.28	Текст программы	19
4.29	Запуск программы	20
4.30	Запуск программы в отладчике	20
4.31	Анализ регистров	21
4.32	Работа программы	21

Список таблиц

3.1	Описание некоторых каталогов файловой системы GNU Linux . . .	7
-----	---	---

1 Цель работы

Приобретение навыков написания программ с использованием подпрограмм.
Знакомство с методами отладки при помощи GDB.

2 Задание

- 1) Выполнение лабораторной работы.
- 2) Выполнение самостоятельной работы.

3 Теоретическое введение

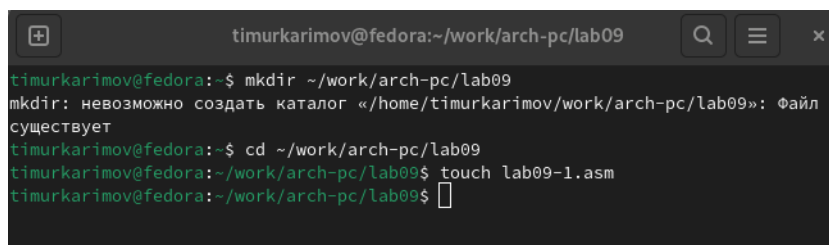
Здесь описываются теоретические аспекты, связанные с выполнением работы. Например, в табл. 3.1 приведено краткое описание стандартных каталогов Unix.

Таблица 3.1: Описание некоторых каталогов файловой системы GNU Linux	
Имя каталога	Описание каталога
/	Корневая директория, содержащая всю файловую
/bin	Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям
/etc	Общесистемные конфигурационные файлы и файлы конфигурации установленных программ
/home	Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя
/media	Точки монтирования для сменных носителей
/root	Домашняя директория пользователя root
/tmp	Временные файлы
/usr	Вторичная иерархия для данных пользователя

Более подробно про Unix см. в [1–4].

4 Выполнение лабораторной работы

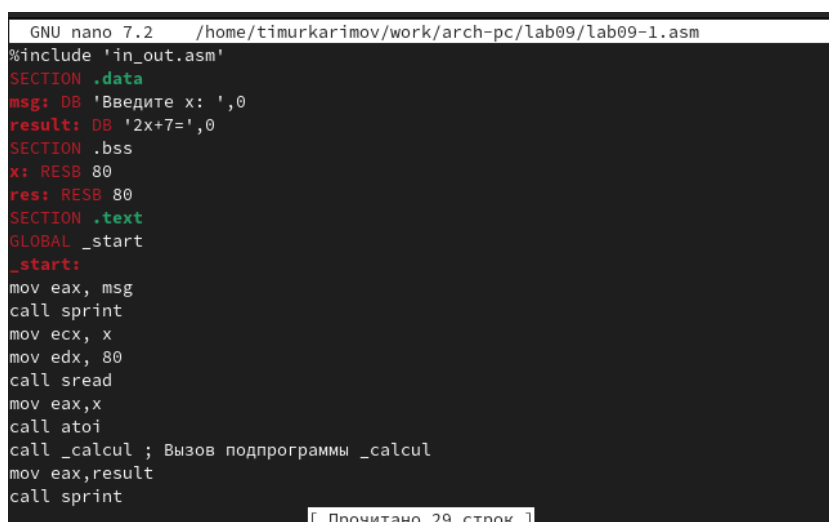
1. Создание каталога и файла Был создан каталог *lab9* и файл *lab9-1.asm* (рис. 4.1).



```
timurkarimov@fedora:~/work/arch-pc/lab09
timurkarimov@fedora:~$ mkdir ~/work/arch-pc/lab09
mkdir: невозможно создать каталог «/home/timurkarimov/work/arch-pc/lab09»: Файл
существует
timurkarimov@fedora:~$ cd ~/work/arch-pc/lab09
timurkarimov@fedora:~/work/arch-pc/lab09$ touch lab09-1.asm
timurkarimov@fedora:~/work/arch-pc/lab09$
```

Рис. 4.1: Создание каталога и файла

2. Ввод текста программы и запуск В файл был введен код программы *Листинг 9.1* (рис. 4.2), после чего он была запущен (рис. 4.3).



```
GNU nano 7.2 /home/timurkarimov/work/arch-pc/lab09/lab09-1.asm
#include 'in_out.asm'
SECTION .data
msg: DB 'Введите x: ',0
result: DB '2x+7=',0
SECTION .bss
x: RESB 80
res: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprint
mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi
call _calcul ; Вызов подпрограммы _calcul
mov eax, result
call sprint
```

Рис. 4.2: Текст программы


```

timurkarimov@fedora:~/work/arch-pc/lab09$ nasm -f elf lab09-1.asm
timurkarimov@fedora:~/work/arch-pc/lab09$ ld -m elf_i386 -o lab09-1 lab09-1.o
timurkarimov@fedora:~/work/arch-pc/lab09$ ./lab09-1
Hello, world!
timurkarimov@fedora:~/work/arch-pc/lab09$ ./lab09-1
Введите x: 2
2x+7=11
timurkarimov@fedora:~/work/arch-pc/lab09$

```

Рис. 4.3: Работа программы

3. Изменение программы Текст программы был изменен для решения выражения $f(g(x))$ (рис. 4.4). После внесения изменений программа была снова запущена (рис. 4.5).

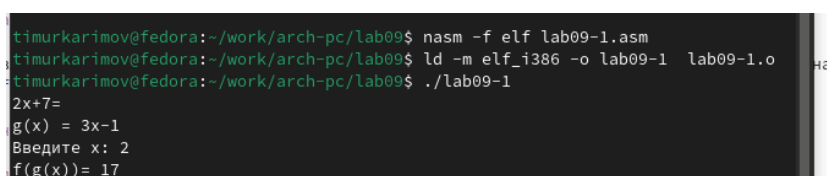


```

GNU nano 7.2 /home/timurkarimov/work/arch-pc/lab09/lab09-1.asm
#include 'in_out.asm'
SECTION .data
msg: DB 'Введите x: ',0
prim1: DB '2x+7=',0
prim2: DB 'g(x) = 3x-1',0
result: DB 'f(g(x))= ',0
SECTION .bss
x: RESB 80
res: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, prim1
call sprintf
mov eax, prim2
call sprintf
mov eax, msg
call sprintf
mov ecx, x
mov edx, 80

```

Рис. 4.4: Создание каталога и файла



```

timurkarimov@fedora:~/work/arch-pc/lab09$ nasm -f elf lab09-1.asm
timurkarimov@fedora:~/work/arch-pc/lab09$ ld -m elf_i386 -o lab09-1 lab09-1.o
timurkarimov@fedora:~/work/arch-pc/lab09$ ./lab09-1
2x+7=
g(x) = 3x-1
Введите x: 2
f(g(x))= 17

```

Рис. 4.5: Создание каталога и файла

4. Создание второго файла Создан файл lab10-2.asm, в который была вписана новая программа (рис. 4.6).

```

GNU nano 7.2 /home/timurkarimov/work/arch-pc/lab09/lab09-2.asm  Изменён
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

```

Рис. 4.6: Текст второй программы

5. Загрузка и отладка второй программы. Файл второй программы был загружен и запущен в отладчике GDB (рис. 4.7).

```

timurkarimov@fedora:~/work/arch-pc/lab09$ nasm -f elf -g -l lab09-2.lst lab09-2.asm
timurkarimov@fedora:~/work/arch-pc/lab09$ ld -m elf_i386 -o lab09-2 lab09-2.o
timurkarimov@fedora:~/work/arch-pc/lab09$ gdb lab09-2
GNU gdb (Fedora Linux) 15.2-1.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 lab09-2...
(gdb) r
Starting program: /home/timurkarimov/work/arch-pc/lab09/lab09-2

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.
Downloading 50.30 K separate debug info for system-supplied DSO at 0xf7ffc000
Hello, world!
[Inferior 1 (process 6369) exited normally]
(gdb)

```

Рис. 4.7: Отладка второго файла

Установлен брекпоинт на метку `_start`, после чего программа была запущена (рис. 4.8).

```
(gdb) break _start
Breakpoint 1 at 0x00400000: file lab09-2.asm, line 11.
(gdb) r
Starting program: /home/timurkarimov/work/arch-pc/lab09/lab09-2
Breakpoint 1, _start () at lab09-2.asm:11
11     mov eax, 4
(gdb)
```

Рис. 4.8: Брекпоинт на метку `_start`

6. Дисассемблирование кода Просмотрен дисассемблированный код программы, начиная с метки `_start` (рис. 4.9).

```
(gdb) disassemble _start
Dump of assembler code for function _start:
=> 0x00400000 <+0>:    mov     $0x4,%eax
0x00400005 <+5>:    mov     $0x1,%ebx
0x0040000a <+10>:   mov     $0x00400008,%ecx
0x0040000f <+15>:   mov     $0x8,%edx
0x00400014 <+20>:   int     $0x80
0x00400016 <+22>:   mov     $0x4,%eax
0x0040001b <+27>:   mov     $0x1,%ebx
0x00400020 <+32>:   mov     $0x00400008,%ecx
0x00400025 <+37>:   mov     $0x7,%edx
0x0040002a <+42>:   int     $0x80
0x0040002c <+44>:   mov     $0x1,%eax
0x00400031 <+49>:   mov     $0x0,%ebx
0x00400036 <+54>:   int     $0x80
End of assembler dump.
(gdb)
```

Рис. 4.9: Дисассемблированный код

7. Смена синтаксиса на *Intel* С помощью команды был переключен синтаксис на Intel. Отличие заключается в отсутствии символов `%` и `$` в командах, что делает отображение более удобным (рис. 4.10).

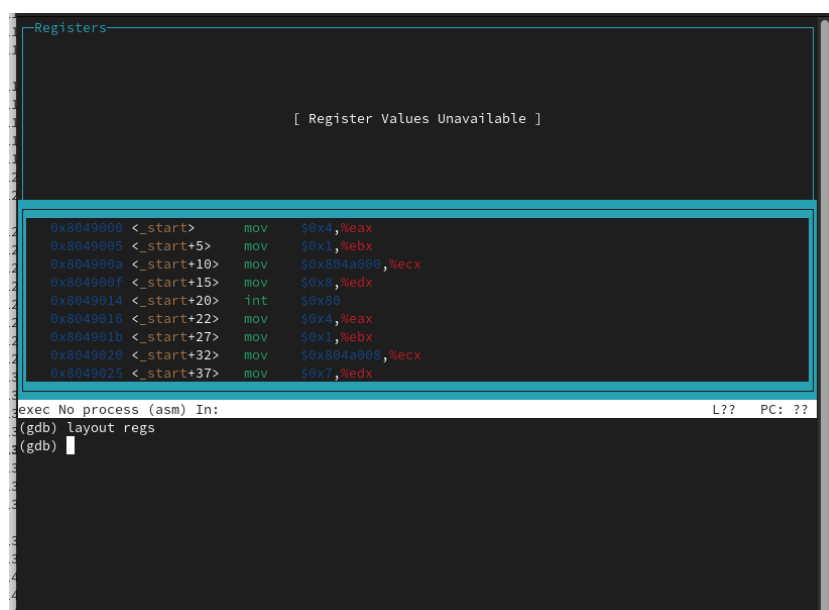
```

(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. 4.10: Отображение Intel

8. Включение режима псевдографики Для удобства работы был включен режим псевдографики (рис. 4.11).



```

Registers
[ Register Values Unavailable ]

0x08049000 <_start>      mov     $0x4,%eax
0x08049005 <_start+5>    mov     $0x1,%ebx
0x0804900a <_start+10>   mov     $0x804a000,%ecx
0x0804900f <_start+15>   mov     $0x8,%edx
0x08049014 <_start+20>   int     $0x80
0x08049016 <_start+22>   mov     $0x4,%eax
0x0804901b <_start+27>   mov     $0x1,%ebx
0x08049020 <_start+32>   mov     $0x804a008,%ecx
0x08049025 <_start+37>   mov     $0x7,%edx

exec No process (asm) In:
(gdb) layout regs
(gdb)

```

Рис. 4.11: Псевдографика

9. Проверка наличия меток Проверено наличие меток в коде, добавлена еще одна метка на предпоследнюю инструкцию (рис. 4.12).

```

36 native process 7803 (asm) In: _start L11 PC: 0x8049000
37 (gdb) layout regs
38 (gdb) info breakpoints
39 Num      Type      Disp Enb Address      What
40 1 breakpoint keep y 0x08049000 lab09-2.asm:11
41 breakpoint already hit 1 time
42 (gdb) break *0x8049031
43 Breakpoint 2 at 0x8049031: file lab09-2.asm, line 26.
44 (gdb) i b
45 Num      Type      Disp Enb Address      What
46 1 breakpoint keep y 0x08049000 lab09-2.asm:11
47 breakpoint already hit 1 time
48 2 breakpoint keep y 0x08049031 lab09-2.asm:26
49 (gdb)

```

Рис. 4.12: Наличие меток

10. Просмотр и изменение регистров С помощью команды *si* были просмотрены значения регистров (рис. 4.13) и выполнены их изменения (рис. 4.14).

```

native process 7803 (asm) In: _start L14 PC: 0x804900f
--Type <RET> for more, q to quit, c to continue without paging--
es      0x2b      43
fs      0x0       0
gs      0x0       0
(gdb) x/1sb &msg1
0x804a000 <msg1>: "Hello, "
(gdb) si

```

Рис. 4.13: Прсмотр регистров

```

eax      0x4       4
ecx      0x0       0
edx      0x0       0
ebx      0x0       0
esp      0xffffd1c0 0xffffd1c0
ebp      0x0       0x0
esi      0x0       0
edi      0x0       0
eip      0x8049005 0x8049005 <_start+5>
eflags   0x202     [ IF ]
cs       0x23      35
ss       0x2b      43
ds       0x2b      43
es       0x2b      43

```

Рис. 4.14: Измененные регистры

11. Проверка значений переменных Проверены значения переменных *msg1* (рис. 4.15) и *msg2* (рис. 4.16).

```

native process 7803 (asm) In: _start L14 PC: 0x804900f
--Type <RET> for more, q to quit, c to continue without paging--
es      0x2b      43
fs      0x0       0
gs      0x0       0
(gdb) x/1sb &msg1
0x804a000 <msg1>: "Hello, "
(gdb) si

```

Рис. 4.15: Просмотр значения переменной msg1

```
(gdb) x/lsb 0x804a008
0x804a008 <msg2>: "world!\n\034"
(gdb)
```

Рис. 4.16: Просмотр значения переменной msg2

12. Изменение значений переменных C помощью команды set были изменены значения переменных *msg1* (рис. 4.17) и *msg2* (рис. 4.18).

```
(gdb) x/lsb 0x804a008
0x804a008 <msg2>: "world!\n\034"
(gdb) set {char}&msg1='h'
(gdb) set {char}0x804a001='h'
(gdb) x/lsb &msg1
No symbol "msg1" in current context.
(gdb) x/lsb &msg1
0x804a000 <msg1>: "hhllo, "
(gdb)
```

Рис. 4.17: Изменение значения переменной msg1

```
0x804a000 <msg1>: "hhllo, "
(gdb) set {char}0x804a008='L'
(gdb) set {char}0x804a00b=' '
(gdb) x/lsb &msg2
0x804a008 <msg2>: "Lor d!\n\034"
(gdb)
```

Рис. 4.18: Изменение значения переменной msg2

13. Вывод значений регистров Выведены значения регистров *ecx* и *eax* (рис. 4.19).

```

native process 7803 (asm) In: _start
$1 = void
(gdb) p/s $eax
$2 = 4
(gdb) p/t $eax
$3 = 100
(gdb) p/c $ecx
$4 = 0 '\000'
(gdb) p/x $ecx
$5 = 0x804a000

```

Рис. 4.19: значения регистров ecx и eax

14. Изменение значения регистра. Значение регистра ebx было изменено, при этом выводились два разных значения, так как в первый раз было внесено значение 2, а во второй раз регистр уже имел это значение (рис. 4.20).

```

$5 = 0x804a000
(gdb) set $ebx='2'
(gdb) p/s $ebx
$6 = 50
(gdb) set $ebx=2
(gdb) p/s $ebx
$7 = 2
(gdb) 

```

Рис. 4.20: Значения регистров ebx

15. Завершение работы с файлами. Работа с файлами была завершена, после чего выполнен выход (рис. 4.21).

```

[Inferior 1 (process 3985) exited normally]

```

Рис. 4.21: Завершение работы с файлом

16. Копирование и переименование файла. Файл lab8-2.asm был скопирован и переименован. Запущен в отладчике с указанием аргументов (рис. 4.22).

```

timurkarimov@fedora:~/work/arch-pc/lab09$ gdb --args lab09-3 аргумент1 аргумент 2 'аргумент 3'
GNU gdb (Fedora Linux) 15.2-1.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 lab09-3...
(No debugging symbols found in lab09-3)
(gdb)

```

Рис. 4.22: Запуск файла в отладчике

17. Запуск нового файла через метку Запущен файл lab10-3 через метку `_start` (рис. 4.23).

```

<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from lab09-3...
(No debugging symbols found in lab09-3)
(gdb) b _start
Breakpoint 1 at 0x004090e8
(gdb) r
Starting program: /home/timurkarimov/work/arch-pc/lab09/lab09-3 аргумент1 аргумент 2 аргумент\ 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, 0x004090e8 in _start ()
(gdb)

```

Рис. 4.23: Запуск файла lab9-3.asm через метку

18. Проверка адреса вершины стека Проверен адрес вершины стека, подтверждено хранение 5 элементов (рис. 4.24).

```

(gdb) x/x $esp
0xffffd180: 0x00000005
(gdb)

```

Рис. 4.24: Адрес вершины стека

19. Просмотр всех позиций стека Просмотрены все позиции стека, где по первому адресу хранится адрес возврата, а остальные адреса содержат элементы, расположенные с интервалом в 4 байта для корректного хранения данных (рис. 4.25).


```

(gdb) x/x $esp
0xffffd000: 0x00000005
(gdb) x/s *(void**)(esp + 4)
0xffffd1c2: "/home/timurkarimov/work/arch-pc/lab09/lab09-3"
(gdb) x/s *(void*)(esp + 8)
Attempt to take contents of a non-pointer value.
(gdb) x/s *(void**)(esp + 8)
0xffffd1f0: "аргумент1"
(gdb) x/s *(void**)(esp + 12)
0xffffd202: "аргумент"
(gdb) x/s *(void**)(esp + 16)
0xffffd213: "2"
(gdb) x/s *(void**)(esp + 20)
0xffffd215: "аргумент 3"
(gdb) x/s *(void**)(esp + 24)
0x0: <error: Cannot access memory at address 0x0>
(gdb)

```

Рис. 4.25: Все позиции стека

#Самостоятельная работа

1. Преобразование программы из лабораторной работы №8 в виде подпрограммы (рис. 4.26).

```

GNU nano 7.2 /home/timurkarimov/work/arch-pc/lab09/lab09-3.asm
#include 'in_out.asm'

SECTION .data
prim DB 'f(x)=2x+15',0
otv DB 'Результат: ',0
SECTION .text
GLOBAL _start
_start:

pop ecx
pop edx
sub ecx,1
mov esi,0

mov eax,prim
call sprintf
next:
cmp ecx,0
jz _end

pop eax
call atoi
call fir
add esi,eax

```

Рис. 4.26: Текст программы

2. Попытка запуска программы для проверки на наличие ошибок (рис. 4.26).

```

timurkarimov@fedora:~/work/arch-pc/lab09$ nasm -f elf -g -l task_1.lst task_1.asm
timurkarimov@fedora:~/work/arch-pc/lab09$ ld -m elf_i386 -o task_1 task_1.o
timurkarimov@fedora:~/work/arch-pc/lab09$ ./task_1.asm 1 2 3
bash: ./task_1.asm: Отказано в доступе
timurkarimov@fedora:~/work/arch-pc/lab09$ ./task_1 1 2 3
f(x)=2x+15
Результат: 57
timurkarimov@fedora:~/work/arch-pc/lab09$ ./task_1 1 2 3 4
f(x)=2x+15

```

Рис. 4.27: Запуск программы

3. Возникновение арифметической ошибки: ожидаемый результат 25, фактический — 10 (рис. 4.28).

```
GNU nano 7.2 /ho
#include 'in_out.asm'

SECTION .data
div: DB 'Результат: ',0

SECTION .text
GLOBAL _start
_start:

mov ebx,3
mov eax,2
add ebx,eax
mov ecx,4
mul ecx
add ebx,5
mov edi,eax

mov eax,div
call sprint
mov eax,edi
call iprintLF

call quit
```

Рис. 4.28: Текст программы

4. Использование отладчика GDB для анализа проблемы (рис. 4.29).

```
timurkarimov@fedora:~/work/arch-pc/lab09$ nasm -f elf -g -l task_2.lst task_2.asm
timurkarimov@fedora:~/work/arch-pc/lab09$ ld -m elf_i386 -o task_2 task_2.o
timurkarimov@fedora:~/work/arch-pc/lab09$ ./task_2
Результат: 8
```

Рис. 4.29: Запуск программы

5. Запуск программы в отладчике (рис. 4.30).

```
timurkarimov@fedora:~/work/arch-pc/lab09$ gdb task_2
GNU gdb (Fedora Linux) 15.2-1.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 task_2...
(gdb) b _start
Breakpoint 1 at 0x80490e8: file task_2.asm, line 10.
(gdb) r
Starting program: /home/timurkarimov/work/arch-pc/lab09/task_2

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 task_2.asm:10
10      mov ebx, 3
```

Рис. 4.30: Запуск программы в отладчике

6. Открытие и анализ состояния регистров (рис. 4.31). Были обнаружены некорректные значения в регистрах, влияющих на вычисления. Были внесены исправлений в их значениях.

```

Register group: general
eax      0x0      0
ecx      0x0      0
edx      0x0      0
ebx      0x0      0
esp      0xffffd1d0 0xffffd1d0
ebp      0x0      0
esi      0x0      0
edi      0x0      0
eip      0x80490e8 0x80490e8 <_start>
eflags   0x202    [ IF ]
cs       0x23     35
ss       0x2b     43

0x80490db <quit>      mov     ebx,0x0
0x80490e0 <quit+5>    mov     eax,0x1
0x80490e5 <quit+10>   int     0x80
0x80490e7 <quit+12>   ret
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>
0x804910a <_start+34> mov     eax,edi

```

Рис. 4.31: Анализ регистров

9. Повторный запуск программы с получением правильного ответа — 25 (рис. 4.32).

```

(gdb)
[14]+  Остановлен   gdb task_2
timurkarimov@fedora:~/work/arch-pc/lab09$ gdb task_2
GNU gdb (Fedora Linux) 15.2-1.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 task_2...
(No debugging symbols found in task_2)
(gdb) r
Starting program: /home/timurkarimov/work/arch-pc/lab09/task_2

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.
Результат: 25
[Inferior 1 (process 15862) exited normally]
(gdb)

```

Рис. 4.32: Работа программы

5 Выводы

В ходе выполнения лабораторной работы были успешно освоены основные принципы работы с ассемблерными программами и отладчиком GDB.

Список литературы

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.
2. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.