

Лабораторная работа номер 10

Сафин Андрей Алексеевич

Содержание

1	Цель работы	5
2	Задание	6
3	Ход лабораторной работы	7
4	Самостоятельная работа	22
5	Выводы	28
	Список литературы	29

Список иллюстраций

3.1	Текст lab10-1.asm	8
3.2	Работа lab10-1.asm	8
3.3	Новый текст lab10-1.asm	9
3.4	Текст lab10-2.asm	10
3.5	Использование gdb к lab10-2	10
3.6	Использование run	11
3.7	Создание брейкпоинта	11
3.8	Дизассемблированный код АТТ	12
3.9	Дизассемблированный код intel	12
3.10	Рассмотрение и создание брейкпоинтов в режиме всевдографики	13
3.11	Рассмотрение инструкций с помощью stepi (1)	14
3.12	Рассмотрение инструкций с помощью stepi (2)	15
3.13	Рассмотрение инструкций с помощью stepi (3)	16
3.14	Рассмотрение инструкций с помощью stepi (4)	17
3.15	Рассмотрение инструкций с помощью stepi (5)	18
3.16	Значение msg1	19
3.17	Новое значение msg1	19
3.18	Изменение значение msg2	19
3.19	Значение edx в разных форматах	20
3.20	Установка и выведение значений ebx	20
3.21	Использование gdb к lab10-3.asm	20
3.22	Установление точки останова на _start	20
3.23	Выведение значения esp	21
3.24	Выведение значений стека	21
4.1	Текст новой программы sr.asm	23
4.2	Выполнение sr	23
4.3	Текст новой программы sr2.asm	24
4.4	Выполнение sr2	24
4.5	Рассмотрение работы регистров в GDB (1)	25
4.6	Рассмотрение работы регистров в GDB (2)	25
4.7	Рассмотрение работы регистров в GDB (3)	26
4.8	Рассмотрение работы регистров в GDB (4)	26
4.9	Текст измененной программы sr2.asm	27
4.10	Работа sr2	27

Список таблиц

1 Цель работы

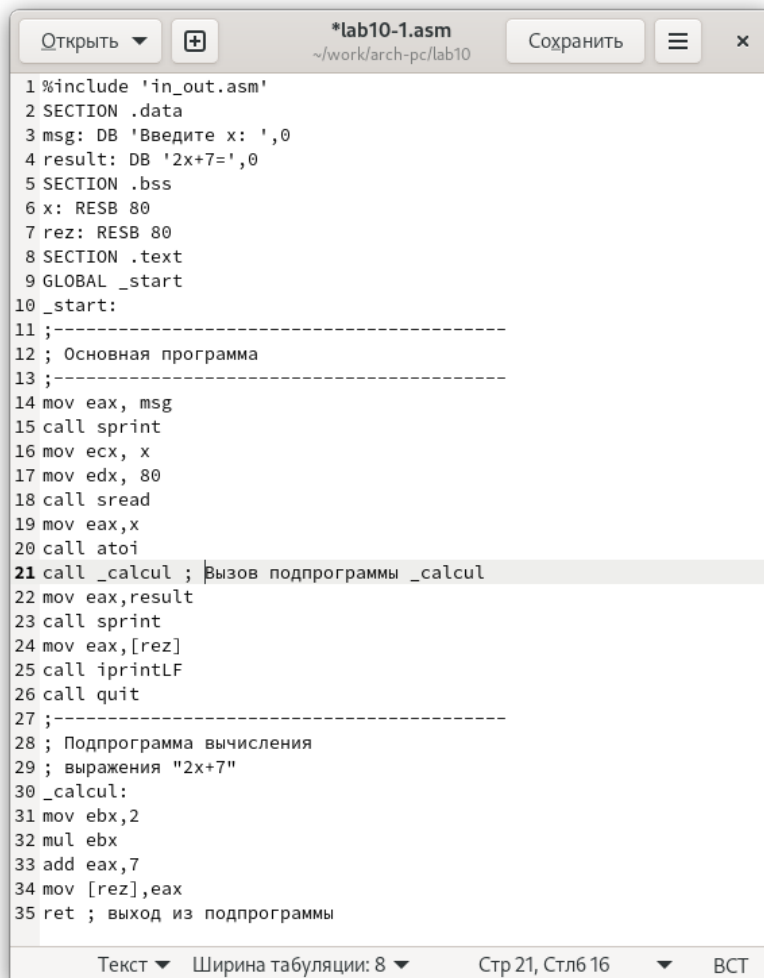
Приобретение навыков написания программ с использованием подпрограмм. Знакомство с методами отладки при помощи GDB и его основными возможностями.

2 Задание

Написать и отладить ряд программ с использованием подпрограмм.

3 Ход лабораторной работы

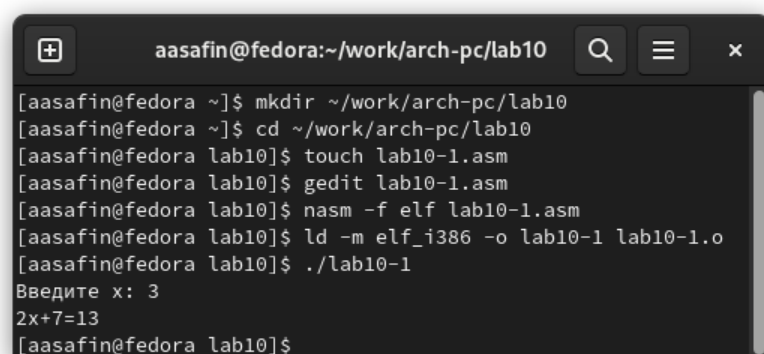
1. Создан файл `~/work/arch-pc/lab10/lab10-1.asm` с программой, вычисляющей $f(x)=2x+7$ с использованием подпрограммы (рис. 3.1). Ее работа проверена (рис. 3.2). В текст добавлена подпрограмма (рис. 3.3), вычисляющая $g(x)=3x-1$, и ссылка на неё вставлена в `lab10-1.asm` так, что вычисляется $f(g(x))$ (рис. ??).



```
1 %include 'in_out.asm'
2 SECTION .data
3 msg: DB 'Введите x: ',0
4 result: DB '2x+7=',0
5 SECTION .bss
6 x: RESB 80
7 rez: RESB 80
8 SECTION .text
9 GLOBAL _start
10 _start:
11 ;-----
12 ; Основная программа
13 ;-----
14 mov eax, msg
15 call sprint
16 mov ecx, x
17 mov edx, 80
18 call sread
19 mov eax,x
20 call atoi
21 call _calcul ; Вызов подпрограммы _calcul
22 mov eax,result
23 call sprint
24 mov eax,[rez]
25 call iprintLF
26 call quit
27 ;-----
28 ; Подпрограмма вычисления
29 ; выражения "2x+7"
30 _calcul:
31 mov ebx,2
32 mul ebx
33 add eax,7
34 mov [rez],eax
35 ret ; выход из подпрограммы
```

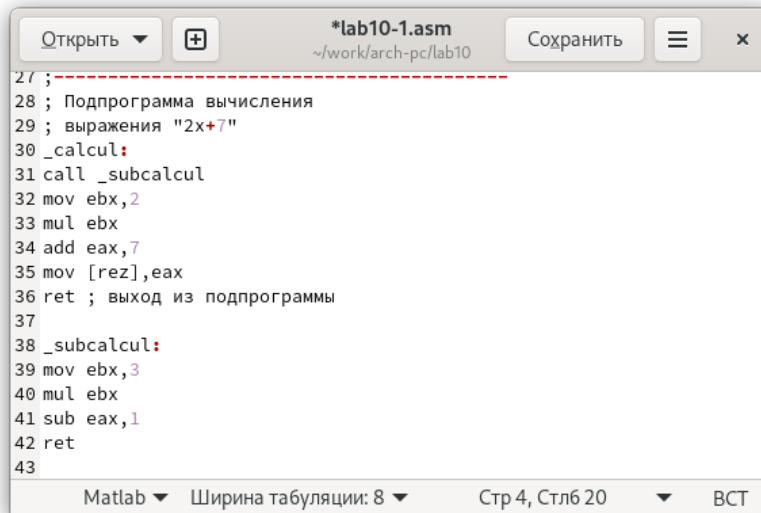
Текст ▾ Ширина табуляции: 8 ▾ Стр 21, Стлб 16 ▾ ВСТ

Рис. 3.1: Текст lab10-1.asm



```
aasafin@fedora:~/work/arch-pc/lab10
[aasafin@fedora ~]$ mkdir ~/work/arch-pc/lab10
[aasafin@fedora ~]$ cd ~/work/arch-pc/lab10
[aasafin@fedora lab10]$ touch lab10-1.asm
[aasafin@fedora lab10]$ gedit lab10-1.asm
[aasafin@fedora lab10]$ nasm -f elf lab10-1.asm
[aasafin@fedora lab10]$ ld -m elf_i386 -o lab10-1 lab10-1.o
[aasafin@fedora lab10]$ ./lab10-1
Введите x: 3
2x+7=13
[aasafin@fedora lab10]$
```

Рис. 3.2: Рабора lab10-1.asm



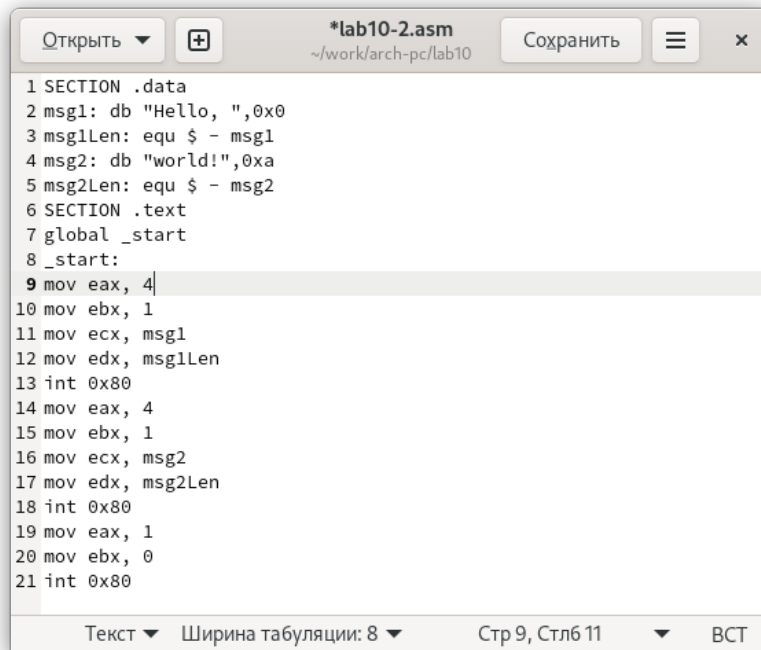
```
27 ;
28 ; Подпрограмма вычисления
29 ; выражения "2x+7"
30 _calcul:
31 call _subcalcul
32 mov ebx,2
33 mul ebx
34 add eax,7
35 mov [rez],eax
36 ret ; выход из подпрограммы
37
38 _subcalcul:
39 mov ebx,3
40 mul ebx
41 sub eax,1
42 ret
43
```

Matlab Ширина табуляции: 8 Стр 4, Стлб 20 ВСТ

Рис. 3.3: Новый текст lab10-1.asm

Повторное выполнение lab10-1.asm

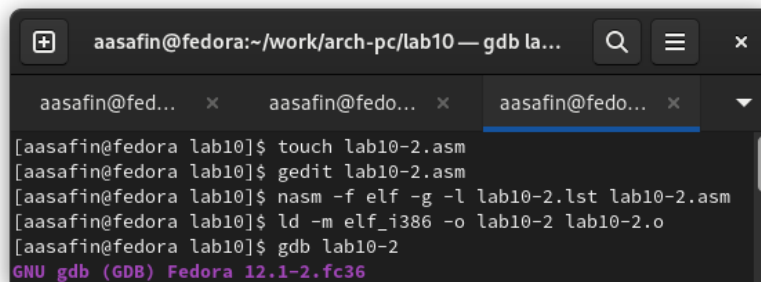
2. Создан файл lab10-2.asm с текстом программы из листинга 10.2 (рис. 3.4), печатающей Hello world. Файл оттранслирован, скомпилирован и загружен в отладчик gdb (рис. 3.5). Проверена её работа с помощью run (рис. 3.6). Установлен брейкпоинт на _start (рис. 3.7). С этой же метки программа дизассемблирована сначала в синтаксисе АТТ (рис. 3.8), а затем в intel (рис. 3.9). Из наблюдаемых отличий, в АТТ ставится \$ перед численными операндами и адресам, и % перед регистрами.



```
1 SECTION .data
2 msg1: db "Hello, ",0x0
3 msg1Len: equ $ - msg1
4 msg2: db "world!",0xa
5 msg2Len: equ $ - msg2
6 SECTION .text
7 global _start
8 _start:
9 mov eax, 4
10 mov ebx, 1
11 mov ecx, msg1
12 mov edx, msg1Len
13 int 0x80
14 mov eax, 4
15 mov ebx, 1
16 mov ecx, msg2
17 mov edx, msg2Len
18 int 0x80
19 mov eax, 1
20 mov ebx, 0
21 int 0x80
```

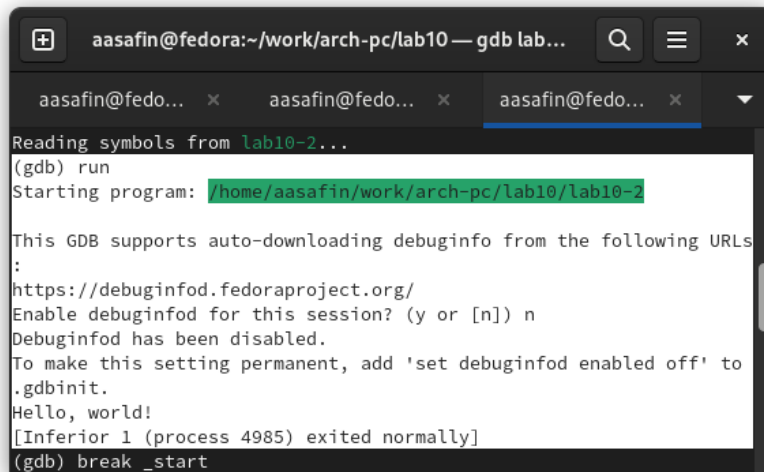
Текст Ширина табуляции: 8 Стр 9, Стлб 11 ВСТ

Рис. 3.4: Текст lab10-2.asm



```
aasafin@fedora:~/work/arch-pc/lab10 — gdb la...
aasafin@fed... x aasafin@fedo... x aasafin@fedo... x
[aasafin@fedora lab10]$ touch lab10-2.asm
[aasafin@fedora lab10]$ gedit lab10-2.asm
[aasafin@fedora lab10]$ nasm -f elf -g -l lab10-2.lst lab10-2.asm
[aasafin@fedora lab10]$ ld -m elf_i386 -o lab10-2 lab10-2.o
[aasafin@fedora lab10]$ gdb lab10-2
GNU gdb (GDB) Fedora 12.1-2.fc36
```

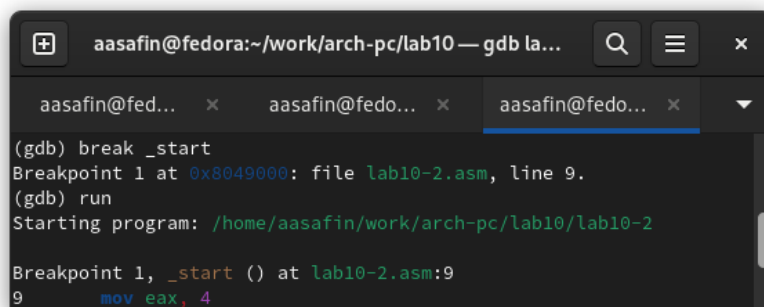
Рис. 3.5: Использование gdb к lab10-2



```
aasafin@fedora:~/work/arch-pc/lab10 — gdb lab...
aasafin@fedo... x aasafin@fedo... x aasafin@fedo... x
Reading symbols from lab10-2...
(gdb) run
Starting program: /home/aasafin/work/arch-pc/lab10/lab10-2

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 4985) exited normally]
(gdb) break _start
```

Рис. 3.6: Использование run



```
aasafin@fedora:~/work/arch-pc/lab10 — gdb la...
aasafin@fedo... x aasafin@fedo... x aasafin@fedo... x
(gdb) break _start
Breakpoint 1 at 0x8049000: file lab10-2.asm, line 9.
(gdb) run
Starting program: /home/aasafin/work/arch-pc/lab10/lab10-2

Breakpoint 1, _start () at lab10-2.asm:9
9      mov eax, 4
```

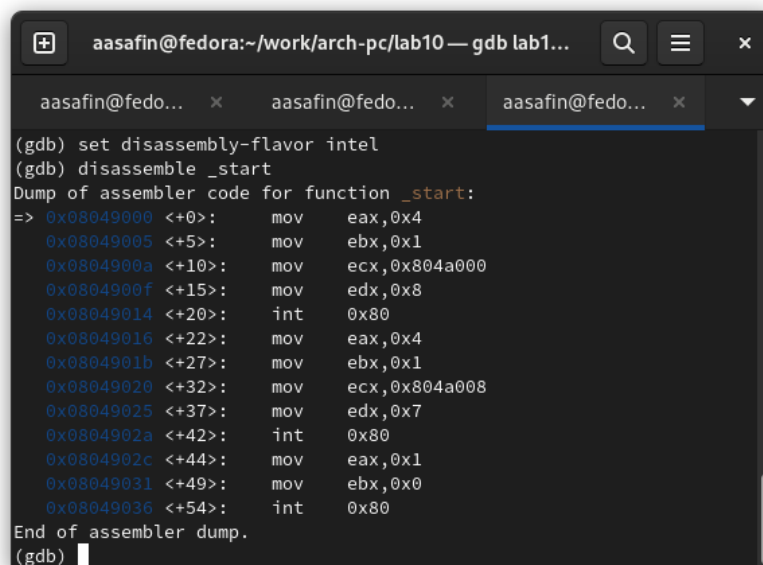
Рис. 3.7: Создание брейкпоинта

```

(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. 3.8: Дизассемблированный код АТТ



```

aasafin@fedora:~/work/arch-pc/lab10 — gdb lab1...
(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. 3.9: Дизассемблированный код intel

Включен режим псевдографики, просмотрена информация по точкам останова, создана ещё одна. (рис. 3.10). С помощью step выполнено пять инструкций (рис. 3.11-3.15). Изменялись значения регистров eax, ebx, ecx и edx.

The screenshot shows a GDB terminal window with the title bar "aasafin@fedora:~/work/arch-pc/lab10 — gdb lab10-2". The window contains several tabs, all named "aasafin@fe...". The main content area displays assembly code with addresses and instructions, highlighted by a red box:

```
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
```

Below the assembly code, the terminal shows the following commands and output:

```
exec No process in: L?? PC: ??
(gdb) layout regs
(gdb) info breakpoints
Num    Type             Disp Enb Address      What
1      breakpoint       keep y  0x08049000 lab10-2.asm:9
(gdb) break *0x8049031
Breakpoint 2 at 0x8049031: file lab10-2.asm, line 20.
(gdb) i b
Num    Type             Disp Enb Address      What
1      breakpoint       keep y  0x08049000 lab10-2.asm:9
2      breakpoint       keep y  0x08049031 lab10-2.asm:20
(gdb)
```

Рис. 3.10: Рассмотрение и создание брейкпоинтов в режиме всевдографики

```
aasafin@fedora:~/work/arch-pc/lab10 — gdb lab10-2
aasafin@fe... x aasafin@fe... x aasafin@fe... x aasafin@fe... x
Register group: general
eax      0x4      4
ecx      0x0      0
edx      0x0      0
ebx      0x0      0
esp      0xffffd230 0xffffd230
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 5947 In: _start L10 PC: 0x8049005
(gdb) run
Starting program: /home/aasafin/work/arch-pc/lab10/lab10-2

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

Breakpoint 1, _start () at lab10-2.asm:9
(gdb) si
(gdb) 
```

Рис. 3.11: Рассмотрение инструкций с помощью stepi (1)

```
aasafin@fedora:~/work/arch-pc/lab10 — gdb lab10-2
aasafin@fe... x aasafin@fe... x aasafin@fe... x aasafin@fe... x
Register group: general
eax      0x4      4
ecx      0x0      0
edx      0x0      0
ebx      0x1      1
esp      0xffffd230 0xffffd230
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 5947 In: _start L11 PC: 0x804900a
Starting program: /home/aasafin/work/arch-pc/lab10/lab10-2

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

Breakpoint 1, _start () at lab10-2.asm:9
(gdb) si
(gdb) ni
(gdb) 
```

Рис. 3.12: Рассмотрение инструкций с помощью stepi (2)

```
aasafin@fedora:~/work/arch-pc/lab10 — gdb lab10-2
aasafin@fe... x aasafin@fe... x aasafin@fe... x aasafin@fe... x
Register group: general
eax      0x4      4
ecx      0x804a000 134520832
edx      0x0      0
ebx      0x1      1
esp      0xffffd230 0xffffd230
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 5947 In: _start L12 PC: 0x804900f

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

Breakpoint 1, _start () at lab10-2.asm:9
(gdb) si
(gdb) ni
(gdb) ni
(gdb)
```

Рис. 3.13: Рассмотрение инструкций с помощью stepi (3)


```
aasafin@fedora:~/work/arch-pc/lab10 — gdb lab10-2
aasafin@fe... x aasafin@fe... x aasafin@fe... x aasafin@fe... x
Register group: general
eax      0x4      4
ecx      0x804a000 134520832
edx      0x8      8
ebx      0x1      1
esp      0xffffd230 0xffffd230
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 5947 In: _start L13 PC: 0x8049014
This GDB supports auto-downloading debuginfo from the following URLs:
https://debuginfod.fedoraproject.org/
Enable debuginfod for this session? (y or [n]) nDebuginfod has been disabled.
To make this setting permanent, add 'set debuginfod enabled off' to .gdbinit.

Breakpoint 1, _start () at lab10-2.asm:9
(gdb) si
(gdb) ni
(gdb) ni
(gdb) ni
(gdb) 
```

Рис. 3.14: Рассмотрение инструкций с помощью stepi (4)

```
aasafin@fedora:~/work/arch-pc/lab10 — gdb lab10-2
aasafin@fe... x aasafin@fe... x aasafin@fe... x aasafin@fe... x
Register group: general
eax      0x8      8
ecx      0x804a000 134520832
edx      0x8      8
ebx      0x1      1
esp      0xffffd230 0xffffd230
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 5947 In: _start L14 PC: 0x8049016
https://debuginfod.fedoraproject.org/
Enable debuginfod for this session? (y or [n]) nDebuginfod has been disabled.
To make this setting permanent, add 'set debuginfod enabled off' to .gdbinit.

Breakpoint 1, _start () at lab10-2.asm:9
(gdb) si
(gdb) ni
(gdb) ni
(gdb) ni
(gdb) ni
(gdb)
```

Рис. 3.15: Рассмотрение инструкций с помощью stepi (5)

Значению msg1 просмотрено (рис. 3.16), изменено (рис. 3.17), а затем просмотрено и изменено значение msg2 ((рис. 3.18). Выведены в различных форматах значение edx (рис. 3.19). С помощью команды set изменено значение ebx сначала на строчную двойку, а затем на численную. Поскольку в обоих случаях выводится численное значение двойки, вывод отличается (рис. 3.20). После выполнение программы было завершено с помощью quit.

```
aasafin@fedora:~/work/arch-pc/lab10 — gdb lab10-2
aasafin@fe... x aasafin@fe... x aasafin@fe... x aasafin@fe... x
Register group: general
eax      0x8      8
ecx      0x804a000 134520832
edx      0x8      8
ebx      0x1      1
esp      0xffffd230 0xffffd230
ebp      0x0      0x0
esi      0x0      0
edi      0x0      0

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

native process 5947 In: _start L14 PC: 0x8049016
Breakpoint 1, _start () at lab10-2.asm:9
(gdb) si
(gdb) ni
(gdb) ni
(gdb) ni
(gdb) ni
(gdb) x/1sb &msg1
0x804a000 <msg1>: "Hello, "
(gdb) x/1sb 0x804a008
0x804a008 <msg2>: "world!\n\034"
(gdb)
```

Рис. 3.16: Значение msg1

```
(gdb) set {char}&msg1='h'
(gdb) x/1sb &msg1
0x804a000 <msg1>: "hello, "
(gdb)
```

Рис. 3.17: Новое значение msg1

```
(gdb) x/1sb &msg2
0x804a008 <msg2>: "world!\n\034"
(gdb) set {char}0x804a00c=' '
(gdb) x/1sb &msg2
0x804a008 <msg2>: "worl !\n\034"
(gdb)
```

Рис. 3.18: Изменение значение msg2

```
(gdb) p/s $edx
$5 = 8
(gdb) p/t $edx
$6 = 1000
(gdb) p/x $edx
$7 = 0x8
(gdb) █
```

Рис. 3.19: Значение `edx` в разных форматах

```
(gdb) set $ebx='2'
(gdb) p/s $ebx
$8 = 50
(gdb) set $ebx=2
(gdb) p/s $ebx
$9 = 2
(gdb) █
```

Рис. 3.20: Установка и вывод значений `ebx`

3. Скопирован `lab9-2.asm` в `lab10-3.asm`, а затем к итоговой программе применен `gdb` при введении многих аргументов с помощью ключа `-args` (рис. 3.21). Установлена точка останова на `_start` (рис. 3.22). Выведено число аргументов, хранящееся в `esp` (рис. 3.23). Выведены значения в остальных позициях стека (рис. 3.24). Адресация сдвигается на четыре, так как на элемент стека выведено по 4 байта.

```
[aasafin@fedora lab10]$ cp ~/work/arch-pc/lab09/lab9-2.asm ~/work/arch-pc/lab10/lab10-3.asm
[aasafin@fedora lab10]$ nasm -f elf -g -l lab10-3.lst lab10-3.asm
[aasafin@fedora lab10]$ ld -m elf_i386 -o lab10-3 lab10-3.o
[aasafin@fedora lab10]$ gdb --args lab10-3 аргумент1 аргумент 2 'аргумент 3'
GNU gdb (GDB) Fedora 12.1-2.fc36
```

Рис. 3.21: Использование `gdb` к `lab10-3.asm`

```
Reading symbols from lab10-3...
(gdb) b _start
Breakpoint 1 at 0x80490e8: file lab10-3.asm, line 5.
(gdb) run
Starting program: /home/aasafin/work/arch-pc/lab10/lab10-3 аргумент1 аргумент 2 аргумент\ 3
```

Рис. 3.22: Установление точки останова на `_start`

```
Breakpoint 1, _start () at lab10-3.asm:5
5      pop ecx ; Извлекаем из стека в `ecx` количество
(gdb) x/x $esp
0xffffd1f0:  0x00000005
```

Рис. 3.23: Выведение значения esp

```
(gdb) x/s *(void**)(esp + 4)
0xffffd3a0:  "/home/aasafin/work/arch-pc/lab10/lab10-3"
(gdb) x/s *(void**)(esp + 8)
0xffffd3c9:  "аргумент1"
(gdb) x/s *(void**)(esp + 12)
0xffffd3db:  "аргумент"
(gdb) x/s *(void**)(esp + 16)
0xffffd3ec:  "2"
(gdb) x/s *(void**)(esp + 20)
0xffffd3ee:  "аргумент 3"
(gdb) x/s *(void**)(esp + 24)
0x0:  <error: Cannot access memory at address 0x0>
(gdb)
```

Рис. 3.24: Выведение значений стека

4 Самостоятельная работа

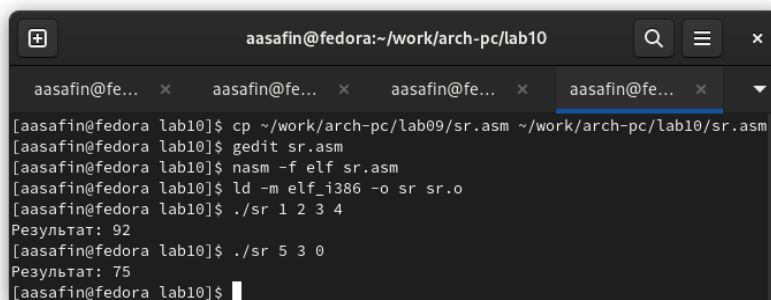
1. Изменена программа из лабораторной работы 9 так, что $f(x)$ вычисляется в подпрограмме (рис. 4.1, 4.2). Результат полностью соответствует таковому из лабораторной работы 9.



```
1 %include 'in_out.asm'
2
3 SECTION .data
4 msg db "Результат: ",0
5
6 SECTION .text
7 global _start
8 _start:
9
10 pop ecx
11 pop edx
12 sub ecx, 1
13
14 mov ebx,0
15
16 cmp ecx,0
17 jz _end
18
19 next:
20 pop eax
21 call atoi
22 call _calc
23 loop next
24
25 _end:
26 mov eax, msg
27 call sprint
28 mov eax, ebx
29 call iprintLF
30 call quit
31
32 _calc:
33 mov edx, 12
34 mul edx
35 sub eax, 7
36 add ebx, eax
37
38 ret
```

Matlab Ширина табуляции: 8 Стр 38, Стлб 4 ВСТ

Рис. 4.1: Текст новой программы sr.asm

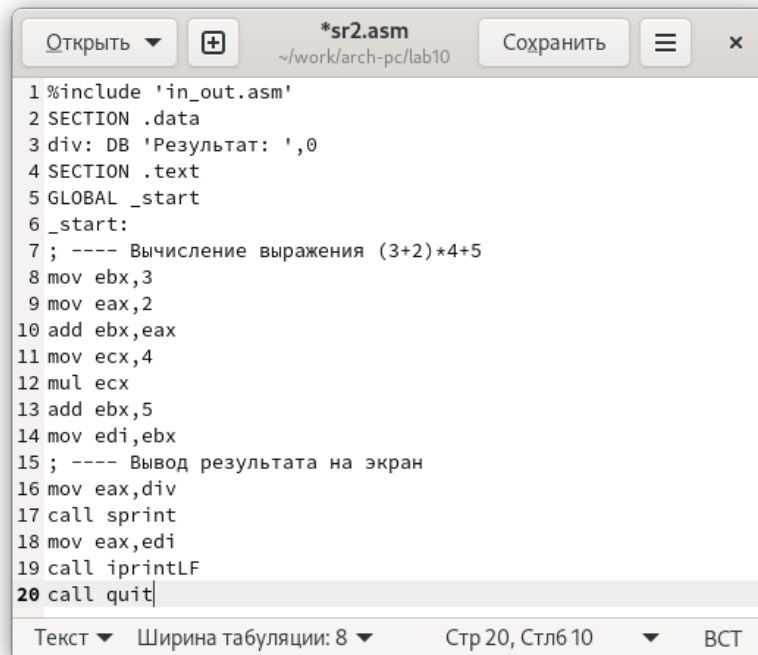


```
aasafin@fedora:~/work/arch-pc/lab10
aasafin@fe... x aasafin@fe... x aasafin@fe... x aasafin@fe... x
[aasafin@fedora lab10]$ cp ~/work/arch-pc/lab09/sr.asm ~/work/arch-pc/lab10/sr.asm
[aasafin@fedora lab10]$ gedit sr.asm
[aasafin@fedora lab10]$ nasm -f elf sr.asm
[aasafin@fedora lab10]$ ld -m elf_i386 -o sr sr.o
[aasafin@fedora lab10]$ ./sr 1 2 3 4
Результат: 92
[aasafin@fedora lab10]$ ./sr 5 3 0
Результат: 75
[aasafin@fedora lab10]$
```

Рис. 4.2: Выполнение sr

2. Создан файл sr2.asm (рис. 4.3), в который введена программа из листинга

10.3, вычисляющая $4(3+2)+5$ с ошибкой (рис. 4.4). С помощью отладчика рассмотрены изменения в регистрах (рис. 4.5-4.8). По ним видно, что ошибка возникает из-за сохранения результата суммы в `ebx` и продолжения работы с этим регистром несмотря на то, что умножение выполняется с `eax`. Ошибка исправлена (рис. 4.9). Программа выполняется верно (рис. 4.10).



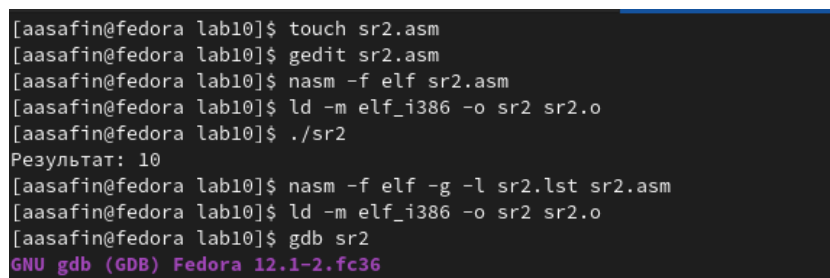
```

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 ebx,eax
11 mov ecx,4
12 mul ecx
13 add ebx,5
14 mov edi,ebx
15 ; ---- Вывод результата на экран
16 mov eax,div
17 call sprint
18 mov eax,edi
19 call iprintLF
20 call quit

```

Текст ▾ Ширина табуляции: 8 ▾ Стр 20, Стлб 10 ▾ ВСТ

Рис. 4.3: Текст новой программы `sr2.asm`



```

[aasafin@fedora lab10]$ touch sr2.asm
[aasafin@fedora lab10]$ gedit sr2.asm
[aasafin@fedora lab10]$ nasm -f elf sr2.asm
[aasafin@fedora lab10]$ ld -m elf_i386 -o sr2 sr2.o
[aasafin@fedora lab10]$ ./sr2
Результат: 10
[aasafin@fedora lab10]$ nasm -f elf -g -l sr2.lst sr2.asm
[aasafin@fedora lab10]$ ld -m elf_i386 -o sr2 sr2.o
[aasafin@fedora lab10]$ gdb sr2
GNU gdb (GDB) Fedora 12.1-2.fc36

```

Рис. 4.4: Выполнение `sr2`


```
aasafin@fedora:~/work/arch-pc/lab10 — gdb sr2
aasafin@fedora:~/work/... x aasafin@fedora:~/work/... x aasafin@fedora:~/work/... x
Register group: general
eax      0x2      2
ecx      0x0      0
edx      0x0      0
ebx      0x3      3
esp      0xffffd240 0xffffd240
ebp      0x0      0
esi      0x0      0

B+ 0x80490e8 <_start> mov $0x3,%ebx
0x80490ed <_start+5> mov $0x2,%eax
> 0x80490f2 <_start+10> add %eax,%ebx
0x80490f4 <_start+12> mov $0x4,%ecx
0x80490f9 <_start+17> mul %ecx
0x80490fb <_start+19> add $0x5,%ebx
0x80490fe <_start+22> mov %ebx,%edi

native process 8834 In: _start L10 PC: 0x80490f2
Breakpoint 1 at 0x80490e8: file sr2.asm, line 8.
(gdb) run
Starting program: /home/aasafin/work/arch-pc/lab10/sr2

Breakpoint 1, _start () at sr2.asm:8
(gdb) si
(gdb) ni
(gdb) █
```

Рис. 4.5: Рассмотрение работы регистров в GDB (1)

```
aasafin@fedora:~/work/arch-pc/lab10 — gdb sr2
aasafin@fedora:~/work/... x aasafin@fedora:~/work/... x aasafin@fedora:~/work/... x
Register group: general
eax      0x2      2
ecx      0x0      0
edx      0x0      0
ebx      0x5      5
esp      0xffffd240 0xffffd240
ebp      0x0      0
esi      0x0      0

B+ 0x80490e8 <_start> mov $0x3,%ebx
0x80490ed <_start+5> mov $0x2,%eax
0x80490f2 <_start+10> add %eax,%ebx
> 0x80490f4 <_start+12> mov $0x4,%ecx
0x80490f9 <_start+17> mul %ecx
0x80490fb <_start+19> add $0x5,%ebx
0x80490fe <_start+22> mov %ebx,%edi

native process 8834 In: _start L11 PC: 0x80490f4
Breakpoint 1 at 0x80490e8: file sr2.asm, line 8.
(gdb) run
Starting program: /home/aasafin/work/arch-pc/lab10/sr2

Breakpoint 1, _start () at sr2.asm:8
(gdb) si
(gdb) ni
(gdb) ni
(gdb) █
```

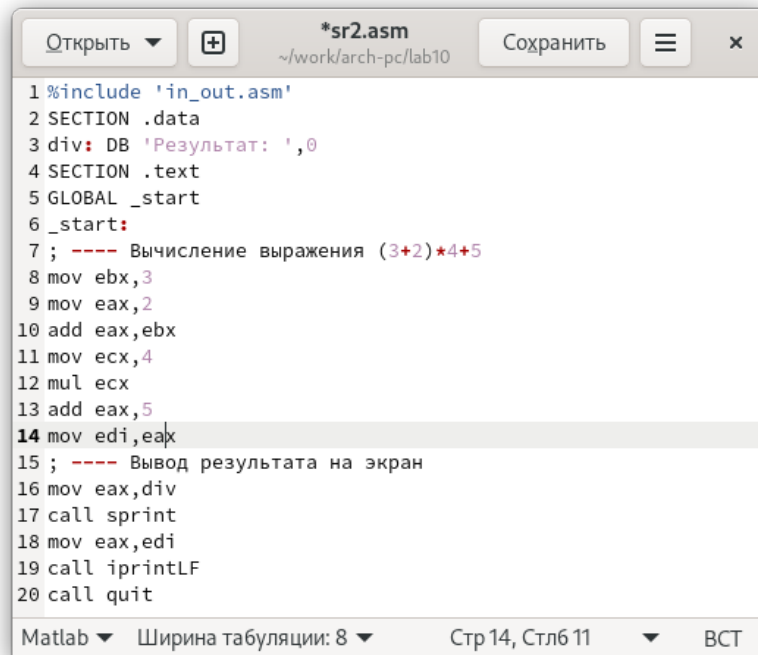
Рис. 4.6: Рассмотрение работы регистров в GDB (2)

```
aasafin@fedora:~/work/arch-pc/lab10 — gdb sr2
aasafin@fedora:~/work/... x aasafin@fedora:~/work/... x aasafin@fedora:~/work/... x
Register group: general
eax      0x8      8
ecx      0x4      4
edx      0x0      0
ebx      0x5      5
esp      0xffffd240 0xffffd240
ebp      0x0      0x0
esi      0x0      0
0x80490f2 <_start+10> add    %eax,%ebx
0x80490f4 <_start+12> mov    $0x4,%ecx
0x80490f9 <_start+17> mul    %ecx
> 0x80490fb <_start+19> add    $0x5,%ebx
0x80490fe <_start+22> mov    %ebx,%edi
0x8049100 <_start+24> mov    $0x804a000,%eax
0x8049105 <_start+29> call   0x804900f <sprint>
native process 8834 In: _start L13 PC: 0x80490fb
Breakpoint 1 at 0x80490e8: file sr2.asm, line 8.
(gdb) run
Starting program: /home/aasafin/work/arch-pc/lab10/sr2
Breakpoint 1, _start () at sr2.asm:8
(gdb) si
(gdb) ni
(gdb) ni
(gdb) ni
(gdb) ni
(gdb) ni
(gdb) 
```

Рис. 4.7: Рассмотрение работы регистров в GDB (3)

```
aasafin@fedora:~/work/arch-pc/lab10 — gdb sr2
aasafin@fedora:~/work/... x aasafin@fedora:~/work/... x aasafin@fedora:~/work/... x
Register group: general
eax      0x8      8
ecx      0x4      4
edx      0x0      0
ebx      0xa      10
esp      0xffffd240 0xffffd240
ebp      0x0      0x0
esi      0x0      0
0x80490f2 <_start+10> add    %eax,%ebx
0x80490f4 <_start+12> mov    $0x4,%ecx
0x80490f9 <_start+17> mul    %ecx
0x80490fb <_start+19> add    $0x5,%ebx
> 0x80490fe <_start+22> mov    %ebx,%edi
0x8049100 <_start+24> mov    $0x804a000,%eax
0x8049105 <_start+29> call   0x804900f <sprint>
native process 8834 In: _start L14 PC: 0x80490fe
(gdb) run
Starting program: /home/aasafin/work/arch-pc/lab10/sr2
Breakpoint 1, _start () at sr2.asm:8
(gdb) si
(gdb) ni
(gdb) ni
(gdb) ni
(gdb) ni
(gdb) ni
(gdb) 
```

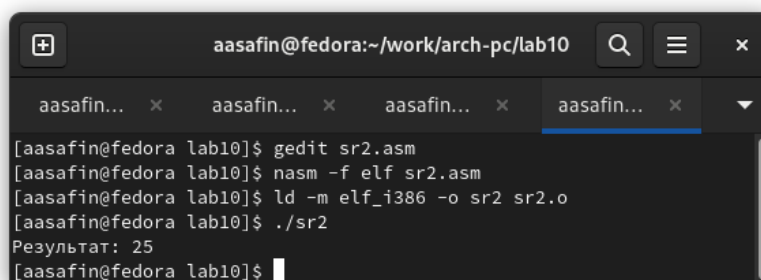
Рис. 4.8: Рассмотрение работы регистров в GDB (4)



```
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
```

Matlab ▾ Ширина табуляции: 8 ▾ Стр 14, Стлб 11 ▾ ВСТ

Рис. 4.9: Текст измененной программы sr2.asm



```
aasafin@fedora:~/work/arch-pc/lab10
aasafin... x aasafin... x aasafin... x aasafin... x
[aasafin@fedora lab10]$ gedit sr2.asm
[aasafin@fedora lab10]$ nasm -f elf sr2.asm
[aasafin@fedora lab10]$ ld -m elf_i386 -o sr2 sr2.o
[aasafin@fedora lab10]$ ./sr2
Результат: 25
[aasafin@fedora lab10]$
```

Рис. 4.10: Работа sr2

5 Выводы

Все программы с подпрограммами составлены. Задания по работе с отладчиком выполнены. Навык работы приобретен.

Список литературы