

Отчёт по лабораторной работе №9

Дисциплина: Архитектура компьютера

Арсений Андреевич Шалин

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выполнение самостоятельной работы	16
4	Выводы	17
	Список литературы	18

Список иллюстраций

2.1	Задание 9.4.1А	6
2.2	Задание 9.4.1Б	6
2.3	Задание 9.4.1В	7
2.4	Задание 9.4.2А	7
2.5	Задание 9.4.2Б	7
2.6	Задание 9.4.2В	7
2.7	Задание 9.4.2Г	8
2.8	Задание 9.4.2Д	8
2.9	Задание 9.4.2.1А	9
2.10	Задание 9.4.2.1Б	9
2.11	Задание 9.4.2.2А	9
2.12	Задание 9.4.2.2Б	10
2.13	Задание ЛР №9	10
2.14	Задание ЛР №9	11
2.15	Задание ЛР №9	11
2.16	Задание ЛР №9	11
2.17	Задание ЛР №9	11
2.18	Задание ЛР №9	12
2.19	Задание ЛР №9	12
2.20	Задание ЛР №9	13
2.21	Задание ЛР №9	13
2.22	Задание ЛР №9	13
2.23	Задание ЛР №9	14
2.24	Задание ЛР №9	14
2.25	Задание ЛР №9	14
2.26	Задание ЛР №9	15
2.27	Задание ЛР №9	15

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

Ввёл в файл lab9-1.asm текст программы из листинга 9.1. Создал исполняемый файл и проверил его работу (рис. 2.1).

```
[aashalin@localhost lab09]$ nasm -f elf lab9-1.asm
[aashalin@localhost lab09]$ ls
in_out.asm  lab9-1.asm  lab9-1.o
[aashalin@localhost lab09]$ ld -m elf_i386 lab9-1.o -o lab9-1
[aashalin@localhost lab09]$ ./lab9-1
Введите x: 3
2x+7=13
[aashalin@localhost lab09]$
```

Рис. 2.1: Задание 9.4.1А

Модифицировал программу lab9-1.asm, добавив подпрограмму. (рис. 2.2).

```
-----
; 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
    mov [res],eax

    ret
```

Рис. 2.2: Задание 9.4.1Б

Создал исполняемый файл и проверил его работу (рис. 2.3).

```
[aashalin@localhost lab09]$ nasm -f elf lab9-1.asm
[aashalin@localhost lab09]$ ld -m elf_i386 lab9-1.o -o lab9-1
[aashalin@localhost lab09]$ ./lab9-1
Введите x: 3
2(3x-1)+7=23
[aashalin@localhost lab09]$
```

Рис. 2.3: Задание 9.4.1В

Создал файл lab9-2.asm с текстом программы из Листинга 9.2, получил исполняемый файл с отладочной информацией, загрузил его в отладчик gdb (рис. 2.4).

```
[aashalin@localhost lab09]$ nasm -f elf -g -l lab9-2.lst lab9-2.asm
[aashalin@localhost lab09]$ ld -m elf_i386 -o lab9-2 lab9-2.o
[aashalin@localhost lab09]$ gdb lab9-2
GNU gdb (CentOS Stream) 14.2-3.el9
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-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. 2.4: Задание 9.4.2А

Проверил работу программы, запустив её в оболочке GDB с помощью команды run (рис. 2.5).

```
(gdb) run
Starting program: /home/aashalin/work/study/2023-2024/Архитектура компьютера/study_2024-2025_arh-pc/labs/lab09/lab9-2
This GDB supports auto-downloading debuginfo from the following URLs:
  <https://debuginfod.centos.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 47.65 K separate debug info for system supplied GDB at 6x11fcb00
Hello, world!
[Inferior 1 (process 3911) exited normally]
(gdb)
```

Рис. 2.5: Задание 9.4.2Б

Установил брейкпоинт на метку _start, запустил программу (рис. 2.6).

```
(gdb) break _start
Breakpoint 1 at 0x0400000: file lab9-2.asm, line 12.
(gdb) run
Starting program: /home/aashalin/work/study/2023-2024/Архитектура компьютера/study_2024-2025_arh-pc/labs/lab09/lab9-2
Breakpoint 1, _start () at lab9-2.asm:12
12      mov eax, 4
(gdb)
(gdb)
```

Рис. 2.6: Задание 9.4.2В

Посмотрел дисассимилированный код программы с помощью команды disassemble начиная с метки _start (рис. 2.7).

```
(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: Задание 9.4.2Г

Переключился на отображение команд с синтаксисом Intel (рис. 2.8). Отличается оно тем, что регистры меняются местами с адресами, а также тем, что перед регистрами не написано %, перед адресами не написано \$.

```
(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: Задание 9.4.2Д

Включил режим псевдографики, проверил, что была установлена точка останова (рис. 2.9).


```

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
0x8049031 <_start+49>     mov     ebx,0x0

exec No process In:
(gdb) layout regs
(gdb) info breakpoints
Num      Type             Disp Enb Address      What
1        breakpoint       keep y   0x08049000 lab9-2.asm:12
(gdb) 

```

Рис. 2.9: Задание 9.4.2.1А

Определил адрес предпоследней инструкции (`mov ebx,0x0`) и установил точку останова, посмотрел информацию о всех установленных точках останова (рис. 2.10). При пошаговом выполнении программы меняются значения `eax`, `ebx`, `ecx`, `edx`.

```

(gdb) b *0x8049031
Breakpoint 2 at 0x8049031: file lab9-2.asm, line 25.
(gdb) i b
Num      Type             Disp Enb Address      What
1        breakpoint       keep y   0x08049000 lab9-2.asm:12
          breakpoint already hit 1 time
2        breakpoint       keep y   0x08049031 lab9-2.asm:25
(gdb) 

```

Рис. 2.10: Задание 9.4.2.1Б

Посмотрел значение переменной `msg1` и переменной `msg2` по адресу (рис. 2.11).

```

(gdb) x/1sb &msg1
0x804a000 <msg1>:      "Hello, "
(gdb) x/1sb 0x804a008
0x804a008 <msg2>:      "world!\n\034"
(gdb) 

```

Рис. 2.11: Задание 9.4.2.2А

Изменил первый символ переменной msg1 (рис. 2.12).

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

Рис. 2.12: Задание 9.4.2.2Б

```
(gdb) p/s $ebx  
$3 = 50  
(gdb) p/t $ebx  
$4 = 110010  
(gdb) p/x $ebx  
$5 = 0x32  
(gdb) 
```

Рис. 2.13: Задание ЛР №9

48	30	0
49	31	1
50	32	2

Рис. 2.14: Задание ЛР №9

```
(gdb) set $ebx=2
(gdb) p/s $ebx
$6 = 2
(gdb) 
```

Рис. 2.15: Задание ЛР №9

```
(gdb) c
Continuing.
hello, world!
[Inferior 1 (process 7527) exited normally]
(gdb) q
[aashalin@localhost lab09]$ 
```

Рис. 2.16: Задание ЛР №9

```
(gdb) x/s *(void**)( $esp + 4)
0xffffcfe4: "/home/aashalin/work/study/2023-2024/Архитектура компьютера/study_2024-2025_arh-pc/labs/lab09/lab9-3"
(gdb) x/s *(void**)( $esp + 8)
0xffffd05d: "аргумент1"
(gdb) x/s *(void**)( $esp + 12)
0xffffd06f: "аргумент"
(gdb) x/s *(void**)( $esp + 16)
0xffffd080: "2"
(gdb) x/s *(void**)( $esp + 20)
0xffffd082: "аргумент 3"
(gdb) x/s *(void**)( $esp + 24)
0x0: <error: Cannot access memory at address 0x0>
```

Рис. 2.17: Задание ЛР №9

```
[aashalin@localhost lab09]$ ./lab9-4 1
Функция:  $f(x)=15x-9$ 
Результат: 6
[aashalin@localhost lab09]$
```

Рис. 2.18: Задание ЛР №9

```
next:
    cmp ecx,0h ; проверяем, есть ли еще аргументы
    jz _end ; если аргументов нет выходим из цикла
    ; (переход на метку `_end`)
    pop eax ; иначе извлекаем следующий аргумент из стека
    call atoi ; преобразуем символ в число
    call _f
    loop next ; переход к обработке следующего аргумента

_end:
    mov eax, msg ; вывод сообщения "Результат: "
    call sprint
    mov eax, esi ; записываем сумму в регистр `eax`
    call iprintLF ; печать результата
    call quit ; завершение программы

_f:
    mov ebx,15
    mul ebx
    sub eax,9
    add esi,eax

ret
```

Рис. 2.19: Задание ЛР №9

```
(gdb) info registers
eax            0x8            8
ecx            0x4            4
edx            0x0            0
ebx            0x5            5
esp            0xffffce50     0xffffce50
ebp            0x0            0x0
esi            0x0            0
edi            0x0            0
eip            0x80490fb       0x80490fb <_start+19>
eflags         0x10206        [ PF IF RF ]
cs             0x23            35
ss             0x2b            43
ds             0x2b            43
es             0x2b            43
fs             0x0            0
gs             0x0            0
(gdb) █
```

Рис. 2.20: Задание ЛР №9

```
(gdb) stepi
13      add ebx,5
(gdb) info registers
eax            0x8            8
ecx            0x4            4
edx            0x0            0
ebx            0x5            5
```

Рис. 2.21: Задание ЛР №9

```
(gdb) stepi
12      mul ecx
(gdb) info registers
eax            0x2            2
ecx            0x4            4
edx            0x0            0
ebx            0x5            5
```

Рис. 2.22: Задание ЛР №9

```

(gdb) stepi
11      mov ecx,4
(gdb) info registers
eax                0x2                2
ecx                0x0                0
edx                0x0                0
ebx                0x5                5

```

Рис. 2.23: Задание ЛР №9

```

(gdb) stepi
10      add ebx,eax
(gdb) info registers
eax                0x2                2
ecx                0x0                0
edx                0x0                0
ebx                0x3                3

```

Рис. 2.24: Задание ЛР №9

```

[aashalin@localhost lab09]$ ./lab9-5
Результат: 25
[aashalin@localhost lab09]$ 

```

Рис. 2.25: Задание ЛР №9

```

#include 'in_out.asm'
SECTION .data
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. 2.26: Задание ЛР №9

```

[aashalin@localhost lab09]$ ./lab9-1
Введите x: 3
2(3x-1)+7=23
[aashalin@localhost lab09]$

```

Рис. 2.27: Задание ЛР №9

3 Выполнение самостоятельной работы

Выполнены все задания лабораторной работы, путём дебагинга в gdb найдены ошибки в исходной программе в строках 10, 14, 15. Скриншот (рис. 2.26). Ошибки исправлены (рис. 2.27).

4 Выводы

- Навыки написания программ с использованием подпрограмм приобретены.
- С методами отладки при помощи GDB и его основными возможностями ознакомился.

Список литературы