

Лабараторная работа №09.

НПИбд-03-24

Подготовил:

Гелдиев Ыхлас. Студенческий номер: 1032249184

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Реализация подпрограмм в NASM	6
2.2	Отладка программ с помощью GDB	10
2.2.1	Добавление точек останова	15
2.2.2	Работа с данными программы в GDB	16
3	Выполнение заданий для самостоятельной работы	21
4	Выводы	26

Список иллюстраций

2.1	Создал lab09-1.asm	6
2.2	Заполнил lab09-1	7
2.3	Запуск lab09-1	8
2.4	Изменил текст lab09-1	9
2.5	Запуск измененного lab09-1	10
2.6	Создание lab09-2	10
2.7	Заполнение lab09-2	11
2.8	Загрузка файла в отладчик	13
2.9	Переключение на синтаксис intel	14
2.10	Режим псевдографики	15
2.11	Проверка и установка точек останова	16
2.12	Просмотр регистров	17
2.13	Просмотр значения msg1	18
2.14	Изменение данных msg	18
2.15	Вывод edx	18
2.16	Изменение значения ebx	18
2.17	Создание lab09-3	19
2.18	Проверка стека	20
3.1	Первая самостоятельная работа	22
3.2	Проверка первой самостоятельной	23
3.3	Заполнение неправильной hw2	24
3.4	Запуск неправильной программы	24
3.5	Исправленный hw2	25
3.6	Проверка выполнения hw2	25

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

2.1 Реализация подпрограмм в NASM

1. Создал каталог для программ, перешел в него и создал файл lab09-1.asm (рис. 2.1)

```
igeldiev@dk2n27 ~ $ mkdir ~/work/arch-pc/lab09
igeldiev@dk2n27 ~ $ cd ~/work/arch-pc/lab09
igeldiev@dk2n27 ~/work/arch-pc/lab09 $ touch lab09-1.asm
igeldiev@dk2n27 ~/work/arch-pc/lab09 $ ls
lab09-1.asm
igeldiev@dk2n27 ~/work/arch-pc/lab09 $
```

Рис. 2.1: Создал lab09-1.asm

2. Заполнил lab09-1.asm (рис. 2.2)

```

1 %include 'in_out.asm'
2
3 SECTION .data
4     msg: DB 'Введите x: ',0
5     result: DB '2x+7=',0
6
7 SECTION .bss
8     x: RESB 80
9     res: RESB 80
10
11 SECTION .text
12 GLOBAL _start
13     _start:
14         mov eax,msg
15         call sprint
16
17         mov ecx,x
18         mov edx,80
19         call sread
20
21         mov eax,x
22         call atoi
23
24         call _calcul
25
26         mov eax,result
27         call sprint
28         mov eax,[res]
29         call iprintLF
30
31         call quit
32
33     _calcul:
34         mov ebx,2
35         mul ebx
36         add eax,7
37         mov [res],eax
38
39         ret

```

Рис. 2.2: Заполнил lab09-1

3. Создал исполняемый файл и запустил его (рис. 2.3)

```
igeldiev@dk2n27 ~/work/arch-pc/lab09 $ nasm -f elf lab09-1.asm
igeldiev@dk2n27 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab09-1 lab09-1.o
igeldiev@dk2n27 ~/work/arch-pc/lab09 $ ./lab09-1
Введите x: 12
2x+7=31
igeldiev@dk2n27 ~/work/arch-pc/lab09 $
```

Рис. 2.3: Запуск lab09-1

4. Изменил текст программы добавив _subcalcul(рис. 2.4)


```

1 %include 'in_out.asm'
2
3 SECTION .data
4     msg: DB 'Введите x: ',0
5     result: DB '2x+7=',0
6
7 SECTION .bss
8     x: RESB 80
9     res: RESB 80
10
11 SECTION .text
12 GLOBAL _start
13     _start:
14         mov eax,msg
15         call sprint
16
17         mov ecx,x
18         mov edx,80
19         call sread
20
21         mov eax,x
22         call atoi
23
24         call _calcul
25
26         mov eax,result
27         call sprint
28         mov eax,[res]
29         call iprintLF
30
31         call quit
32
33     _calcul:
34         call _subcalcul
35         mov ebx,2
36         mul ebx
37         add eax,7
38         mov [res],eax
39
40         ret
41
42     _subcalcul: ; 3x-1
43         mov ebx,3
44         mul ebx
45         sub eax,1
46
47         ret

```

Рис. 2.4: Изменил текст lab09-1

5. Создал исполняемый файл и запустил его (рис. 2.5)

```
igeldiev@dk2n27 ~/work/arch-pc/lab09 $ gedit lab09-1.asm
igeldiev@dk2n27 ~/work/arch-pc/lab09 $ nasm -f elf lab09-1.asm
igeldiev@dk2n27 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab09-1 lab09-1.o
igeldiev@dk2n27 ~/work/arch-pc/lab09 $ ./lab09-1
Введите x: 12
2x+7=77
igeldiev@dk2n27 ~/work/arch-pc/lab09 $
```

Рис. 2.5: Запуск измененного lab09-1

2.2 Отладка программ с помощью GDB

8. Создал файл lab09-2.asm (рис. 2.6)

```
igeldiev@dk2n27 ~/work/arch-pc/lab09 $ touch lab09-2.asm
igeldiev@dk2n27 ~/work/arch-pc/lab09 $
```

Рис. 2.6: Создание lab09-2

9. Заполнил lab09-2.asm (рис. 2.7)

```

1 SECTION .data
2     msg1: db "Hello, ",0x0
3     msg1Len: equ $ - msg1
4
5     msg2: db "world!",0xa
6     msg2Len: equ $ - msg2
7
8 SECTION .text
9     global _start
10
11 _start:
12     mov eax, 4
13     mov ebx, 1
14     mov ecx, msg1
15     mov edx, msg1Len
16     int 0x80
17
18     mov eax, 4
19     mov ebx, 1
20     mov ecx, msg2
21     mov edx, msg2Len
22     int 0x80
23
24     mov eax, 1
25     mov ebx, 0
26     int 0x80
27

```

Рис. 2.7: Заполнение lab09-2

10. Получил исполняемый файл с отладочной информацией при помощи

ключа -g при трансляции и загрузил исполняемый файл в отладчик. Также проверил работу программы при помощи команды run, поставил точку останова при помощи break и запустил его, а так же посмотрел дисассимблированный код (рис. 2.8).

```

igeldiev@dk2n27 ~/work/arch-pc/lab09 $ gdb lab09-2
GNU gdb (Gentoo 14.2 vanilla) 14.2
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 lab09-2...
(gdb) run
Starting program: /afs/.dk.sci.pfu.edu.ru/home/i/g/igeldiev/work/arch-pc/lab09/lab09-2
Hello, world!
[Inferior 1 (process 5326) exited normally]
(gdb) break _start
Breakpoint 1 at 0x8049000: file lab09-2.asm, line 12.
(gdb) run
Starting program: /afs/.dk.sci.pfu.edu.ru/home/i/g/igeldiev/work/arch-pc/lab09/lab09-2

Breakpoint 1, _start () at lab09-2.asm:12
12      mov eax, 4
(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.

```

Рис. 2.8: Загрузка файла в отладчик

11. Переключаюсь на отображение команд с Intel'овским синтаксисом. (рис. 2.9)

```

(gdb) set disassembly-flavor intel
(gdb) disassembly _start
Undefined command: "disassembly". Try "help".
(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. 2.9: Переключение на синтаксис intel

12. Включил режим псевдографики (рис. 2.10)

```
[ Register Values Unavailable ]

0x80491a6      add     BYTE PTR [eax],al
0x80491a8      add     BYTE PTR [eax],al
0x80491aa      add     BYTE PTR [eax],al
0x80491ac      add     BYTE PTR [eax],al
0x80491ae      add     BYTE PTR [eax],al
0x80491b0      add     BYTE PTR [eax],al
0x80491b2      add     BYTE PTR [eax],al
0x80491b4      add     BYTE PTR [eax],al
0x80491b6      add     BYTE PTR [eax],al
0x80491b8      add     BYTE PTR [eax],al
0x80491ba      add     BYTE PTR [eax],al
0x80491bc      add     BYTE PTR [eax],al
0x80491be      add     BYTE PTR [eax],al

native process 5644 In: _start                                L12    PC: 0x8049000
(gdb) layout regs
(gdb) █
```

Рис. 2.10: Режим псевдографики

2.2.1 Добавление точек останова

13. Проверка устоновки точки остонова и установления еще одного в конце программы (рис. 2.11)

```
[ Register Values Unavailable ]

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
0x8049036 <_start+54>   int     0x80

native process 5644 In: _start L12 PC: 0x8049000
No breakpoint at *0x8049031.
No breakpoint at *0x8049031.
No breakpoint at *0x8049031.
(gdb) break _start
Breakpoint 5 at 0x8049000: file lab09-2.asm, line 12.
(gdb) info breakpoints
Num   Type             Disp Enb Address      What
5     breakpoint        keep y  0x08049000 lab09-2.asm:12
(gdb) break *8049031
Breakpoint 6 at 0x7ad187
(gdb) i b
Num   Type             Disp Enb Address      What
5     breakpoint        keep y  0x08049000 lab09-2.asm:12
6     breakpoint        keep y  0x007ad187
(gdb) □
```

Рис. 2.11: Проверка и установка точек останова

2.2.2 Работа с данными программы в GDB

15. Посмотрел содержимое регистров (рис. 2.12)


```

Register group: general
eax      0x8      8
ecx      0x804a000 134520832
edx      0x8      8
ebx      0x1      1
esp      0xffffc4c0 0xffffc4c0
ebp      0x0      0x0
esi      0x0      0
edi      0x0      0
eip      0x8049016 0x8049016 <_start+22>
eflags   0x202    [ IF ]
cs       0x23     35
ss       0x2b     43
ds       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 6843 In: _start L18 PC: 0x8049016
edx      0x8      8
ebx      0x1      1
esp      0xffffc4c0 0xffffc4c0
ebp      0x0      0x0
esi      0x0      0
edi      0x0      0
eip      0x8049016 0x8049016 <_start+22>
eflags   0x202    [ IF ]
cs       0x23     35
ss       0x2b     43
ds       0x2b     43
es       0x2b     43
--Type <RET> for more, q to quit, c to continue without paging--q
Quit
(gdb) 

```

Рис. 2.12: Просмотр регистров

16. Просмотр значения переменной msg1 (рис. 2.13)

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

Рис. 2.13: Просмотр значения msg1

17. Изменил первый символ msg1 и msg2 (рис. 2.14)

```
(gdb) set {char}&msg1='h'
(gdb) x/1sb &msg1
0x804a000 <msg1>: "hello, "
(gdb) set {char}&msg2='w'
(gdb) x/1sb &msg2
0x804a008 <msg2>: "world!\n\034"
(gdb) set {char}&msg2='W'
(gdb) x/1sb &msg2
0x804a008 <msg2>: "World!\n\034"
(gdb) □
```

Рис. 2.14: Изменение данных msg

18. Вывел в различных форматах edx (рис. 2.15)

```
(gdb) p/s $edx
$1 = 8
(gdb) p/t $edx
$2 = 1000
(gdb) p/x $edx
$3 = 0x8
(gdb) p/s $edx
$4 = 8
(gdb) □
```

Рис. 2.15: Вывод edx

19. Изменение значения ebx (рис. 2.16)

```
(gdb) set $ebx='2'
(gdb) p/s $ebx
$1 = 50
(gdb) set $ebx=2
(gdb) p/s $ebx
$2 = 2
(gdb) □
```

Рис. 2.16: Изменение значения ebx

20. Создание lab09-3.asm и создал исполняемый файл, так же загрузил исполняемый файл в gdb с аргументами (рис. 2.17)

```
igeldiev@dk2n27 ~/work/arch-pc/lab09 $ cp ~/work/arch-pc/lab08/lab8-2.asm ~/work/arch-pc/lab09/lab09-3.asm
igeldiev@dk2n27 ~/work/arch-pc/lab09 $ nasm -f elf -g -l lab09-3.lst lab09-3.asm
igeldiev@dk2n27 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab09-3 lab09-3.o
igeldiev@dk2n27 ~/work/arch-pc/lab09 $ gdb --args lab09-3 аргумент1 аргумент 2 'аргумент 3'
GNU gdb (Gentoo 14.2 vanilla) 14.2
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 lab09-3...
(gdb) █
```

Рис. 2.17: Создание lab09-3

21. Установил точку останова в начале файла и запустил ее. Проверил все позиции стека (рис. 2.18)

```

(gdb) b _start
Breakpoint 1 at 0x80490e8: file lab09-3.asm, line 10.
(gdb) run
Starting program: /afs/.dk.sci.pfu.edu.ru/home/i/g/igeldiev/work/arch-pc/lab09/lab09-3 аргумент1 аргумен
т 2 аргумент\ 3

Breakpoint 1, _start () at lab09-3.asm:10
10      pop ecx                                ; Извлекаем из стека в `ecx` количество
(gdb) x/x $esp
0xfffffc480: 0x00000005
(gdb) x/s *(void**)(esp+4)
0xfffffc6d1: "/afs/.dk.sci.pfu.edu.ru/home/i/g/igeldiev/work/arch-pc/lab09/lab09-3"
(gdb) x/s *(void**)(esp+8)
0xfffffc716: "аргумент1"
(gdb) x/s *(void**)(esp+12)
0xfffffc728: "аргумент"
(gdb) x/s *(void**)(esp+16)
0xfffffc739: "2"
(gdb) x/s *(void**)(esp+20)
0xfffffc73b: "аргумент 3"
(gdb) x/s *(void**)(esp+24)
0x0: <error: Cannot access memory at address 0x0>
(gdb) 

```

Рис. 2.18: Проверка стека

3 Выполнение заданий для самостоятельной работы

1. Преобразовал программу самостоятельной работы №8 (рис. 3.1) (рис. 3.2).

```

1 %include 'in_out.asm'
2
3 SECTION .data
4     msg1 db 'Функция: f(x)=4x+2',0
5     msg2 db 'Результат: ',0
6
7 SECTION .text
8 global _start
9
10 _start:
11     mov eax, msg1
12     call sprintf
13
14     pop ecx
15     pop edx
16     dec ecx
17     mov esi,0
18
19 next:
20     cmp ecx,0h
21     jz _end
22
23     pop eax
24     call atoi
25
26     call _calculate
27
28     add esi, eax
29
30     loop next
31
32 _end:
33     mov eax,msg2
34     call sprintf
35
36     mov eax,esi
37     call iprintLF
38
39     call quit
40
41 _calculate:
42     mov edx, 4
43     mul edx
44     add eax, 2
45
46     ret
47

```

Рис. 3.1: Первая самостоятельная работа

```
igeldiev@dk2n27 ~/work/arch-pc/lab09 $ nasm -f elf hw1.asm
igeldiev@dk2n27 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o hw1 hw1.o
igeldiev@dk2n27 ~/work/arch-pc/lab09 $ ./hw1
Функция:  $f(x)=4x+2$ 
Результат: 0
igeldiev@dk2n27 ~/work/arch-pc/lab09 $ ./hw1 12
Функция:  $f(x)=4x+2$ 
Результат: 50
igeldiev@dk2n27 ~/work/arch-pc/lab09 $ ./hw1 1 2 3
Функция:  $f(x)=4x+2$ 
Результат: 30
igeldiev@dk2n27 ~/work/arch-pc/lab09 $
```

Рис. 3.2: Проверка первой самостоятельной

2. Заполнение неправильной программы hw2.asm и запуск программы в gdb для нахождения ошибки. (рис. 3.3) (рис. 3.4)

```

1 %include 'in_out.asm'
2 SECTION .data
3     div: DB 'Результат: ',0
4
5 SECTION .text
6     GLOBAL _start
7 _start:
8     ; ---- Вычисление выражения (3+2)*4+5
9     mov ebx,3
10    mov eax,2
11    add ebx,eax
12    mov ecx,4
13    mul ecx
14    add ebx,5
15    mov edi,ebx
16
17    mov eax,div
18    call sprint
19    mov eax,edi
20    call iprintLF
21
22    call quit

```

Рис. 3.3: Заполнение неправильной hw2

```

igeldiev@dk2n27 ~/work/arch-pc/lab09 $ nasm -f elf hw2.asm
igeldiev@dk2n27 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o hw2 hw2.o
igeldiev@dk2n27 ~/work/arch-pc/lab09 $ ./hw2
Результат: 10
igeldiev@dk2n27 ~/work/arch-pc/lab09 $ 

```

Рис. 3.4: Запуск неправильной программы

3. Нахождение ошибка в неправильном использовании mul и замены ebx на

eax в некоторых местах и проверка программы (рис. 3.5) (рис. 3.6)

```
1 %include 'in_out.asm'
2 SECTION .data
3     div: DB 'Результат: ',0
4
5 SECTION .text
6     GLOBAL _start
7 _start:
8     ; ---- Вычисление выражения (3+2)*4+5
9     mov ebx,3
10    mov eax,2
11    add eax,ebx
12    mov ecx,4
13    mul ecx
14    add eax,5
15    mov edi,eax
16
17    mov eax,div
18    call sprint
19    mov eax,edi
20    call iprintLF
21
22    call quit
```

Рис. 3.5: Исправленный hw2

```
igeldiev@dk2n27 ~/work/arch-pc/lab09 $ nasm -f elf hw2.asm
igeldiev@dk2n27 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o hw2 hw2.o
igeldiev@dk2n27 ~/work/arch-pc/lab09 $ ./hw2
Результат: 25
igeldiev@dk2n27 ~/work/arch-pc/lab09 $
```

Рис. 3.6: Проверка выполнения hw2

4 Выводы

Я научился писать программы с использованием подпрограмм. Я познакомился с методами отладки при помощи GDB и его основными возможностями.