

# **Отчёт по лабораторной работе №9**

**НПМбв-02-21**

Гугульян Ксения Александровна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>4</b>	<b>Выводы</b>	<b>13</b>
	<b>Список литературы</b>	<b>14</b>

## Список иллюстраций

3.1	Создание каталога . . . . .	7
3.2	Ввод текста в файл . . . . .	8
3.3	Создание исп. файла . . . . .	8
3.4	Создание файла с текстом . . . . .	9
3.5	Загрузка исп. файла в отладчик . . . . .	10
3.6	Проверка . . . . .	10
3.7	Установка брейкпоинт . . . . .	10
3.8	Просмотр программы . . . . .	11
3.9	Переключение . . . . .	11
3.10	Включение режима . . . . .	12

## 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

1. Создайте каталог для выполнения лабораторной работы № 9, перейдите в него и создайте файл lab09-1.asm.
2. Введите в файл lab09-1.asm текст программы из листинга 9.1. Создайте исполняемый файл и проверьте его работу. Измените текст программы, добавив подпрограмму \_subcalcul в подпрограмму \_calcul, для вычисления выражения  $\text{f}(\text{f}(\text{f}(x)))$ , где  $x$  вводится с клавиатуры,  $\text{f}(x) = 2x + 7$ ,  $\text{f}(x) = 3x - 1$ .
3. Создайте файл lab09-2.asm с текстом программы из Листинга 9.2. Получите исполняемый файл. Загрузите исполняемый файл в отладчик gdb. Проверьте работу программы, запустив ее в оболочке GDB с помощью команды run.

### 3 Выполнение лабораторной работы

1. Создаём каталог для выполнения лабораторной работы № 9, перейдём в него и создаём файл lab09-1.asm (рис. 3.1).

```
kaguguljyan@dk6n62 ~ $ mkdir ~/work/arch-pc/lab09
kaguguljyan@dk6n62 ~ $ cd ~/work/arch-pc/lab09
kaguguljyan@dk6n62 ~/work/arch-pc/lab09 $ touch lab09-1.asm
kaguguljyan@dk6n62 ~/work/arch-pc/lab09 $
```

Рис. 3.1: Создание каталога

2. Введём в файл lab09-1.asm текст программы из листинга 9.1 (рис. 3.2).

```

lab09-1.asm      [-M--]  0 L:[ 1+29 30/ 30]
%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
mov eax,result
call sprint
mov eax,[res]
call iprintLF
call quit
_calcul:
mov ebx,2
mul ebx
add eax,7
mov [res],eax
ret

```

Рис. 3.2: Ввод текста в файл

Создаём исполняемый файл и проверяем его работу (рис. 3.3).

```

kaguguljyan@dk6n62 ~/work/arch-pc/lab09 $ nasm -f elf lab09-1.asm
kaguguljyan@dk6n62 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab09-1 lab09-1.o
kaguguljyan@dk6n62 ~/work/arch-pc/lab09 $ ./lab09-1
Введите x: 10
2x+7=27
kaguguljyan@dk6n62 ~/work/arch-pc/lab09 $ █

```

Рис. 3.3: Создание исп. файла



3. Создаём файл lab09-2.asm с текстом программы из Листинга 9.2 (рис. 3.4).



```
lab09-2.asm      [-M--]  0 L:[ 1+21  22/
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
mov edx, msg2Len
int 0x80
mov eax, 1
mov ebx, 0
int 0x80
```

Рис. 3.4: Создание файла с текстом

Получаем исполняемый файл и загружаем исполняемый файл в отладчик gdb (рис. 3.5).

```

kaguguljyan@dk6n62 ~/work/arch-pc/lab09 $ nasm -f elf -g -l lab09-2.lst lab09-2.asm
kaguguljyan@dk6n62 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab09-2 lab09-2.o
kaguguljyan@dk6n62 ~/work/arch-pc/lab09 $ gdb lab09-2
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/>.

```

Рис. 3.5: Загрузка исп. файла в отладчик

Проверяем работу программы, запустив ее в оболочке GDB с помощью команды `run` (рис. 3.6).

```

(gdb) run
Starting program: /afs/.dk.sci.pfu.edu.ru/home/k/a/kaguguljyan/work/arch-pc/lab09/lab09-2
Hello, world!
[Inferior 1 (process 9748) exited normally]
(gdb)

```

Рис. 3.6: Проверка

Установим брейкпоинт на метку `_start`, с которой начинается выполнение любой ассемблерной программы, и запустим её (рис. 3.7).

```

(gdb) break _start
Breakpoint 1 at 0x8049000: file lab09-2.asm, line 9.
(gdb) run
Starting program: /afs/.dk.sci.pfu.edu.ru/home/k/a/kaguguljyan/work/arch-pc/lab09/lab09-2

Breakpoint 1, _start () at lab09-2.asm:9
9      mov eax, 4
(gdb)

```

Рис. 3.7: Установка брейкпоинт

Посмотрим дисассимилированный код программы с помощью команды `disassemble` начиная с метки `_start` (рис. 3.8).

```
(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: Просмотр программы

Переключимся на отображение команд с Intel'овским синтаксисом, введя команду `set disassembly-flavor intel` (рис. 3.9).

```
(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: Переключение

Включаем режим псевдографики для более удобного анализа программы (рис. 3.10).

```
[ 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 9800 In: _start L9
(gdb) layout regs
(gdb) 
```

Рис. 3.10: Включение режима

## 4 Выводы

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

## **Список литературы**