

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет физико-математических и естественных наук
Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 9

дисциплина: Архитектура компьютера

Студент: Бусардыков Артур Валерьевич

Группа: НпМБВ-02-21

МОСКВА

2023 г.

Цель Работы:

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

Выполнение работы

```
root@arturfedora:~# mkdir ~/work/arch-pc/lab09
cd ~/work/arch-pc/lab09
touch lab09-1.as
root@arturfedora:~/work/arch-pc/lab09# nano lab09-1.as
root@arturfedora:~/work/arch-pc/lab09# nasm -f elf lab9-1.asm
nasm: fatal: unable to open input file `lab9-1.asm' No such file or directory
root@arturfedora:~/work/arch-pc/lab09# cp ~/work/arch-pc/lab06/in_out.asm ~/work/arch-pc/lab09
root@arturfedora:~/work/arch-pc/lab09# ./lab09-1
-bash: ./lab09-1: Нет такого файла или каталога
root@arturfedora:~/work/arch-pc/lab09# nasm -f elf lab09-1.asm
nasm: fatal: unable to open input file `lab09-1.asm' No such file or directory
root@arturfedora:~/work/arch-pc/lab09# mv lab09-1.as lab09-1.asm
root@arturfedora:~/work/arch-pc/lab09# nasm -f elf lab09-1.asm
root@arturfedora:~/work/arch-pc/lab09# ld -m elf_i386 -o lab09-1 lab09-1.o
root@arturfedora:~/work/arch-pc/lab09# ./lab09-1
Введите x: 2
2x+7=11
root@arturfedora:~/work/arch-pc/lab09#
```

Создал каталог и файл, выполнил и ввел X.

```
root@arturfedora:~/work/arch-pc/lab09# touch lab09-2.asm
root@arturfedora:~/work/arch-pc/lab09# nano lab09-2.asm
root@arturfedora:~/work/arch-pc/lab09# nasm -f elf -g -l lab09-2.lst lab09-2.asm
root@arturfedora:~/work/arch-pc/lab09# ld -m elf_i386 -o lab09-2 lab09-2.o
root@arturfedora:~/work/arch-pc/lab09# gdb lab09-2
GNU gdb (Fedora Linux) 14.1-4.fc39
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".
```

Создал другой файл - lab09-2.asm и с помощью дополнительных параметров -g -l исполнил, и запустил с gdb. Результат:

```
<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: /root/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 separate debug info for system-supplied DSO at 0xf7ffc000
Hello, world!
[Inferior 1 (process 8476) exited normally]
(gdb)
```

Зашел в gdb, добавил точки останова

```
root@arturfedora:~/work/arch-pc/lab09 [ 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
native process 8481 In: _start L9 PC: 0x8049000
(gdb) layout regs
(gdb)

root@arturfedora:~/work/arch-pc/lab09 [ Register Values Unavailable ]
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
b+ 0x8049038 add BYTE PTR [eax],al
native process 8481 In: _start L9 PC: 0x8049000
(gdb) layout regs
(gdb) info breakpoints
Num Type Disp Enb Address What
1 breakpoint keep y 0x08049000 lab09-2.asm:9
breakpoint already hit 1 time
(gdb) break *0x8049038
Breakpoint 2 at 0x8049038
(gdb)
```

Провел работы по изучению команд: Вывел на экран:

```
root@arturfedora:~/work/arch-pc/lab09

[ Register Values Unavailable ]

lab09-2.asm
 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

exec No process in: L?? PC: ??
(gdb) info registers
The program has no registers now.
(gdb) x/1sb &msg1
0x804a000 <msg1>: "Hello, "
(gdb)
```

```
root@arturfedora:~/work/arch-pc/lab09

Register group: general
eax      0x0      0
ecx      0x0      0
edx      0x0      0
ebx      0x2      2
esp      0xffffd460 0xffffd460
ebp      0x0      0x0
esi      0x0      0
edi      0x0      0
eip      0x8049000 0x8049000 <_start>
eflags   0x202     [ IF ]
cs       0x23     35
ss       0x2b     43
ds       0x2b     43

B+>  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

native process 8528 In: _start L9 PC: 0x8049000
ebp      0x0      0x0
esi      0x0      0
edi      0x0      0
eip      0x8049000 0x8049000 <_start>
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) set $ebx=2
(gdb) p/s $ebx
$1 = 2
(gdb)
```

Вывод:

Приобрел навыки написания программ с использованием подпрограмм. Познакомился с методами отладки при помощи GDB и его основными возможностями

