

Отчёт по лабораторной работе №10

Дисциплина: Архитектура компьютера

Рыжкова Ульяна Валерьевна

Содержание

1 Цель работы

Научиться работать с отладчиком gdb.

2 Выполнение лабораторной работы

1. С помощью терминала создадим подкаталог, создадим файл lab10-1.asm.

Изучим и запишем в него код из листинга, откомпилируем и запустим файл

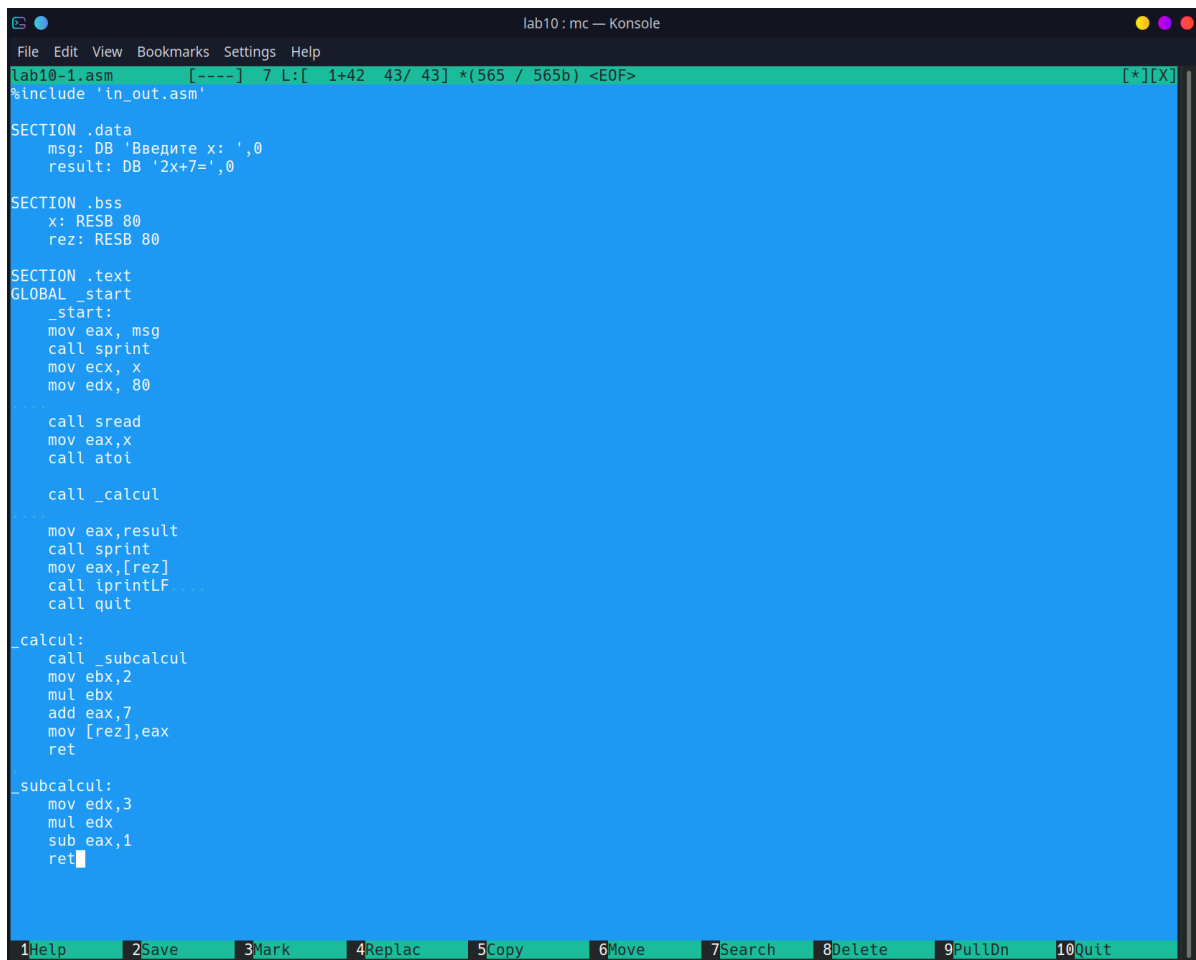
```
uvrihzhkova@barmaglot:/home/ru> cd "work/study/2022-2023/Архитектура компьютера/arch-pc"
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc> mkdir lab10
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc> cd lab10
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> touch lab10-1.asm
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> mc

uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> nasm -f elf lab10-1.asm
lab10-1.asm:26: error: symbol `res' undefined
lab10-1.asm:31: error: label `_calcul' changed during code generation [-w+error=label-redef-late]
lab10-1.asm:35: error: symbol `rez' undefined
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> mc

uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> nasm -f elf lab10-1.asm
lab10-1.asm:27: error: symbol `res' undefined
lab10-1.asm:31: error: label `_calcul' changed during code generation [-w+error=label-redef-late]
lab10-1.asm:35: error: symbol `rez' undefined
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> mc

uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> nasm -f elf lab10-1.asm
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> ld -m elf_i386 -o lab10-1 lab10-1.o

uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> ./lab10-1
Введите x: 2
2x+7=11
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> █
```



```
lab10-1.asm [----] 7 L: [ 1+42 43/ 43] *(565 / 565b) <EOF> [*][X]
%include 'in_out.asm'

SECTION .data
    msg: DB 'Введите x: ',0
    result: DB '2x+7=',0

SECTION .bss
    x: RESB 80
    rez: 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

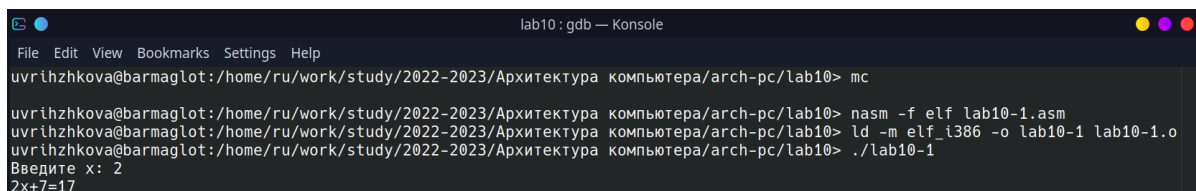
    call _calcul

    mov eax, result
    call sprint
    mov eax, [rez]
    call iprintfLF
    call quit

_calcul:
    call _subcalcul
    mov ebx, 2
    mul ebx
    add eax, 7
    mov [rez], eax
    ret

_subcalcul:
    mov edx, 3
    mul edx
    sub eax, 1
    ret
```

3. Добавим в подпрограмму ещё одну подпрограмму, проверим корректность работы



```
lab10: gdb — Konsole
File Edit View Bookmarks Settings Help
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> mc
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> nasm -f elf lab10-1.asm
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> ld -m elf_i386 -o lab10-1 lab10-1.o
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> ./lab10-1
Введите x: 2
2x+7=17
```

4. Создадим новый файл, запишем в него предложенный код, запустим отладчик и в нем запустим программу

```
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> touch lab10-2.asm
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> mc

uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> nasm -f elf -g -l lab10-2.lst lab10-2.asm
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> ld -m elf_i386 -o lab10-2 lab10-2.o

uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> gdb lab10-2
GNU gdb (GDB; openSUSE Leap 15.2) 11.1
Copyright (C) 2021 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-suse-linux".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://bugs.opensuse.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 lab10-2...
(gdb) █
```

```
lab10 : gdb — Konsole
File Edit View Bookmarks Settings Help
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> mc
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> nasm -f elf lab10-1.asm
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> ld -m elf_i386 -o lab10-1 lab10-1.o
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> ./lab10-1
Введите x: 2
2x+7=17
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> touch lab10-2.asm
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> mc
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> nasm -f elf -g -l lab10-2.lst lab10-2.asm
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> ld -m elf_i386 -o lab10-2 lab10-2.o
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> gdb lab10-2
GNU gdb (GDB; openSUSE Leap 15.2) 11.1
Copyright (C) 2021 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-suse-linux".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://bugs.opensuse.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 lab10-2...
(gdb) run
Starting program: /home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10/lab10-2
Hello, world!
[Inferior 1 (process 9297) exited normally]
(gdb) break _start
Breakpoint 1 at 0x0048080
(gdb) run
Starting program: /home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10/lab10-2
Breakpoint 1, 0x0048080 in _start ()
(gdb) █
```

5. Установим брейкпоинт

```
lab10: gdb — Konsole
File Edit View Bookmarks Settings Help

uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> nasm -f elf -g -l lab10-2.lst lab10-2.asm
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> ld -m elf_i386 -o lab10-2 lab10-2.o

uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> gdb lab10-2
GNU gdb (GDB; openSUSE Leap 15.2) 11.1
Copyright (C) 2021 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-suse-linux".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://bugs.opensuse.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 lab10-2...
(gdb) run
Starting program: /home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10/lab10-2
Hello, world!
[Inferior 1 (process 9297) exited normally]
(gdb) break _start
Breakpoint 1 at 0x08048080
(gdb) run
Starting program: /home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10/lab10-2

Breakpoint 1, 0x08048080 in _start ()
(gdb) disassemble _start
Dump of assembler code for function _start:
=> 0x08048080 <+0>: mov $0x4,%eax
0x08048085 <+5>: mov $0x1,%ebx
0x0804808a <+10>: mov $0x80490b8,%ecx
0x0804808f <+15>: mov $0x8,%edx
0x08048094 <+20>: int $0x80
0x08048096 <+22>: mov $0x4,%eax
0x0804809b <+27>: mov $0x1,%ebx
0x080480a0 <+32>: mov $0x80490c0,%ecx
0x080480a5 <+37>: mov $0x7,%edx
0x080480aa <+42>: int $0x80
0x080480ac <+44>: mov $0x1,%eax
0x080480b1 <+49>: mov $0x0,%ebx
0x080480b6 <+54>: int $0x80
End of assembler dump.
(gdb) █
```

6. Рассмотрим отличия между синтаксисами. Ячейки памяти находятся с разных сторон от значений в них и в АТТ добавляются символы \$ и %


```
lab10: gdb — Konsole
File Edit View Bookmarks Settings Help
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from lab10-2...
(gdb) run
Starting program: /home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10/lab10-2
Hello, world!
[Inferior 1 (process 9297) exited normally]
(gdb) break _start
Breakpoint 1 at 0x8048080
(gdb) run
Starting program: /home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10/lab10-2

Breakpoint 1, 0x8048080 in _start ()
(gdb) disassemble _start
Dump of assembler code for function _start:
=> 0x08048080 <+0>: mov     $0x4,%eax
    0x08048085 <+5>: mov     $0x1,%ebx
    0x0804808a <+10>: mov     $0x80490b8,%ecx
    0x0804808f <+15>: mov     $0x8,%edx
    0x08048094 <+20>: int     $0x80
    0x08048096 <+22>: mov     $0x4,%eax
    0x0804809b <+27>: mov     $0x1,%ebx
    0x080480a0 <+32>: mov     $0x80490c0,%ecx
    0x080480a5 <+37>: mov     $0x7,%edx
    0x080480aa <+42>: int     $0x80
    0x080480ac <+44>: mov     $0x1,%eax
    0x080480b1 <+49>: mov     $0x0,%ebx
    0x080480b6 <+54>: int     $0x80
End of assembler dump.
(gdb) set disassembly-flavor intel
(gdb) disassemble _start
Dump of assembler code for function _start:
=> 0x08048080 <+0>: mov     eax,0x4
    0x08048085 <+5>: mov     ebx,0x1
    0x0804808a <+10>: mov     ecx,0x80490b8
    0x0804808f <+15>: mov     edx,0x8
    0x08048094 <+20>: int     0x80
    0x08048096 <+22>: mov     eax,0x4
    0x0804809b <+27>: mov     ebx,0x1
    0x080480a0 <+32>: mov     ecx,0x80490c0
    0x080480a5 <+37>: mov     edx,0x7
    0x080480aa <+42>: int     0x80
    0x080480ac <+44>: mov     eax,0x1
    0x080480b1 <+49>: mov     ebx,0x0
    0x080480b6 <+54>: int     0x80
End of assembler dump.
(gdb) █
```

7. Выведем режимы псевдографики, по началу layout regs будет пустой

```
lab10: gdb — Konsole
File Edit View Bookmarks Settings Help

[ Register Values Unavailable ]

B+> 0x8048080 <_start> mov eax,0x4
0x8048085 <_start+5> mov ebx,0x1
0x804808a <_start+10> mov ecx,0x80490b8
0x804808f <_start+15> mov edx,0x8
0x8048094 <_start+20> int 0x80
0x8048096 <_start+22> mov eax,0x4
0x804809b <_start+27> mov ebx,0x1
0x80480a0 <_start+32> mov ecx,0x80490c0
0x80480a5 <_start+37> mov edx,0x7
0x80480aa <_start+42> int 0x80
0x80480ac <_start+44> mov eax,0x1
0x80480b1 <_start+49> mov ebx,0x0
0x80480b6 <_start+54> int 0x80
0x80480b8 dec eax
0x80480b9 gs ins BYTE PTR es:[edi],dx

native process 9310 In: _start L?? PC: 0x8048080
(gdb) layout regs
(gdb) █
```

8. Добавим точки останова

```
lab10: gdb — Konsole
File Edit View Bookmarks Settings Help

[ Register Values Unavailable ]

0x80492f5 add BYTE PTR [eax],al
0x80492f7 add BYTE PTR [eax],dl
0x80492f9 add BYTE PTR [eax],al
0x80492fb add BYTE PTR [eax],al
0x80492fd add BYTE PTR [eax],al
0x80492ff add BYTE PTR [ecx],ah
0x8049301 add BYTE PTR [eax],al
0x8049303 add BYTE PTR [ecx],al
0x8049305 add BYTE PTR [eax],al
0x8049307 add BYTE PTR [ebx],al
0x8049309 add BYTE PTR [eax],al
0x804930b add BYTE PTR [eax-0x47f7fb70],bh
0x8049311 add BYTE PTR [eax],al
0x8049313 add BYTE PTR [edi],cl
0x8049315 add BYTE PTR [eax],al

native process 9310 In: _start L?? PC: 0x8048080
(gdb) layout regs
(gdb) info breakpoints
Num Type Disp Enb Address What
1 breakpoint keep y 0x08048080 <_start>
breakpoint already hit 1 time
(gdb) break *0x8049031
Breakpoint 2 at 0x8049031
(gdb) i b
Num Type Disp Enb Address What
1 breakpoint keep y 0x08048080 <_start>
breakpoint already hit 1 time
2 breakpoint keep y 0x8049031
(gdb) █
```

9. С помощью команды `i r` посмотрим содержимое регистров

```
lab10 : gdb — Konsole
File Edit View Bookmarks Settings Help

[ Register Values Unavailable ]

0x80492f5 add BYTE PTR [eax],al
0x80492f7 add BYTE PTR [eax],dl
0x80492f9 add BYTE PTR [eax],al
0x80492fb add BYTE PTR [eax],al
0x80492fd add BYTE PTR [eax],al
0x80492ff add BYTE PTR [ecx],ah
0x8049301 add BYTE PTR [eax],al
0x8049303 add BYTE PTR [ecx],al
0x8049305 add BYTE PTR [eax],al
0x8049307 add BYTE PTR [ebx],al
0x8049309 add BYTE PTR [eax],al
0x804930b add BYTE PTR [eax-0x47f7fb70],bh
0x8049311 add BYTE PTR [eax],al
0x8049313 add BYTE PTR [edi],cl
0x8049315 add BYTE PTR [eax],al

native process 9310 In: _start L?? PC: 0x8048080
eax 0x0 0
ecx 0x0 0
edx 0x0 0
ebx 0x0 0
esp 0xffffce50 0xffffce50
ebp 0x0 0x0
esi 0x0 0
edi 0x0 0
eip 0x8048080 0x8048080 <_start>
eflags 0x202 [ IF ]
cs 0x23 35
ss 0x2b 43
ds 0x2b 43
es 0x2b 43
fs 0x0 0
--Type <RET> for more, q to quit, c to continue without paging--
```

10. Теперь поменяем значение в 1 регистре на другое

```
lab10: gdb — Konsole
File Edit View Bookmarks Settings Help

[ Register Values Unavailable ]

0x8049417    add    BYTE PTR [eax],al
0x8049419    add    BYTE PTR [eax],al
0x804941b    add    BYTE PTR [eax],al
0x804941d    add    BYTE PTR [eax],al
0x804941f    add    BYTE PTR [eax],al
0x8049421    add    BYTE PTR [eax],al
0x8049423    add    BYTE PTR [eax],al
0x8049425    add    BYTE PTR [eax],al
0x8049427    add    BYTE PTR [eax],al
0x8049429    add    BYTE PTR [eax],al
0x804942b    add    BYTE PTR [eax],al
0x804942d    add    BYTE PTR [eax],al
0x804942f    add    BYTE PTR [eax],al
0x8049431    add    BYTE PTR [eax],al
0x8049433    add    BYTE PTR [eax],al

native process 9310 In: _start                                L??  PC: 0x8048080
ds      0x2b      43
es      0x2b      43
fs      0x0       0
--Type <RET> for more, q to quit, c to continue without paging--qQuit
(gdb) x/1sb &msg1
0x80490b8 <msg1>:      "Hello, "
(gdb) x/1sb 0x804a008
0x804a008:      <error: Cannot access memory at address 0x804a008>
(gdb) set {char}msg1='h'
'msg1' has unknown type; cast it to its declared type
(gdb) x/1sb &msg1
0x80490b8 <msg1>:      "Hello, "
(gdb) set {char}&msg1='h'
(gdb) x/1sb &msg1
0x80490b8 <msg1>:      "hello, "
(gdb) █
```

```
lab10: gdb — Konsole
File Edit View Bookmarks Settings Help

[ Register Values Unavailable ]

0x8049417    add    BYTE PTR [eax],al
0x8049419    add    BYTE PTR [eax],al
0x804941b    add    BYTE PTR [eax],al
0x804941d    add    BYTE PTR [eax],al
0x804941f    add    BYTE PTR [eax],al
0x8049421    add    BYTE PTR [eax],al
0x8049423    add    BYTE PTR [eax],al
0x8049425    add    BYTE PTR [eax],al
0x8049427    add    BYTE PTR [eax],al
0x8049429    add    BYTE PTR [eax],al
0x804942b    add    BYTE PTR [eax],al
0x804942d    add    BYTE PTR [eax],al
0x804942f    add    BYTE PTR [eax],al
0x8049431    add    BYTE PTR [eax],al
0x8049433    add    BYTE PTR [eax],al

native process 9310 In: _start                                L??  PC: 0x8048080
--Type <RET> for more, q to quit, c to continue without paging--qQuit
(gdb) x/1sb &msg1
0x80490b8 <msg1>:      "Hello, "
(gdb) x/1sb 0x804a008
0x804a008:      <error: Cannot access memory at address 0x804a008>
(gdb) set {char}msg1='h'
'msg1' has unknown type; cast it to its declared type
(gdb) x/1sb &msg1
0x80490b8 <msg1>:      "Hello, "
(gdb) set {char}&msg1='h'
(gdb) x/1sb &msg1
0x80490b8 <msg1>:      "hello, "
(gdb) set {char}&msg2='!'
(gdb) x/1sb &msg2
0x80490c0 <msg2>:      "!orld!\n"
(gdb) █
```

11. Воспользуемся функцией (set) и поменяем значение

```
lab10: gdb — Konsole
File Edit View Bookmarks Settings Help

Register group: general
eax      0x0      0      ecx      0x0      0
edx      0x0      0      ebx      0x2      2
esp      0xffffce50 0xffffce50  ebp      0x0      0x0
esi      0x0      0      edi      0x0      0
eip      0x8048080 0x8048080 <_start>  eflags    0x202    [ IF ]
cs       0x23     35     ss       0x2b     43
ds       0x2b     43     es       0x2b     43
fs       0x0      0      gs       0x0      0

0x8049417 add BYTE PTR [eax],al
0x8049419 add BYTE PTR [eax],al
0x804941b add BYTE PTR [eax],al
0x804941d add BYTE PTR [eax],al
0x804941f add BYTE PTR [eax],al
0x8049421 add BYTE PTR [eax],al
0x8049423 add BYTE PTR [eax],al
0x8049425 add BYTE PTR [eax],al
0x8049427 add BYTE PTR [eax],al
0x8049429 add BYTE PTR [eax],al
0x804942b add BYTE PTR [eax],al
0x804942d add BYTE PTR [eax],al
0x804942f add BYTE PTR [eax],al
0x8049431 add BYTE PTR [eax],al
0x8049433 add BYTE PTR [eax],al

native process 9310 In: _start
0x80490c0 <msg2>: "!\orld!\n"
(gdb) p/F $ecx
No symbol "F" in current context.
(gdb) p/x $ecx
$1 = 0x0
(gdb) p/x $eax
$2 = 0x0
(gdb) p/l $eax
$3 = Undefined output format "l".
(gdb) set $ebx='2'
(gdb) p/s $ebx
$4 = 50
(gdb) set $ebx=2
(gdb) p/s $ebx
$5 = 2
(gdb) █
```

12. Запустим программу из 9 лабораторной, установим брейкпоинт и изучим, что лежит в стэке. Шаг равен 4, потому что в 1 ячейке стэка 4 байта информации

```
lab10: gdb — Konsole
File Edit View Bookmarks Settings Help
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> gdb --args lab10-3 аргумент1 аргумент 2 'аргумент 3'
GNU gdb (GDB; openSUSE Leap 15.2) 11.1
Copyright (C) 2021 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-suse-linux".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://bugs.opensuse.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 lab10-3...
(gdb) b _start
Breakpoint 1 at 0x8048148
(gdb) run
Starting program: /home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10/lab10-3 аргумент1 аргумент 2 аргумент\ 3

Breakpoint 1, 0x8048148 in _start ()
(gdb) x/x $esp
0xfffffce00: 0x00000005
(gdb) x/s *9void**($esp + 4)
Invalid number "9void".
(gdb) x/s *(void**)(esp + 4)
0xfffffd078: "/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10/lab10-3"
(gdb) x/s *(void**)(esp + 8)
0xfffffd0d8: "аргумент1"
(gdb) x/s *(void**)(esp + 12)
0xfffffd0ea: "аргумент"
(gdb) x/s *(void**)(esp + 16)
0xfffffd0fb: "2"
(gdb) x/s *(void**)(esp + 20)
0xfffffd0fd: "аргумент 3"
(gdb) x/s *(void**)(esp + 24)
0x0: <error: Cannot access memory at address 0x0>
(gdb) █
```


3 Самостоятельная работа

1. Скопируем файл и изменим код

```
lab10: bash — Konsole
File Edit View Bookmarks Settings Help
This GDB was configured as "x86_64-suse-linux".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://bugs.opensuse.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 lab10-3...
(gdb) b _start
Breakpoint 1 at 0x8048148
(gdb) run
Starting program: /home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10/lab10-3 аргумент1 аргумент 2 аргумент\ 3

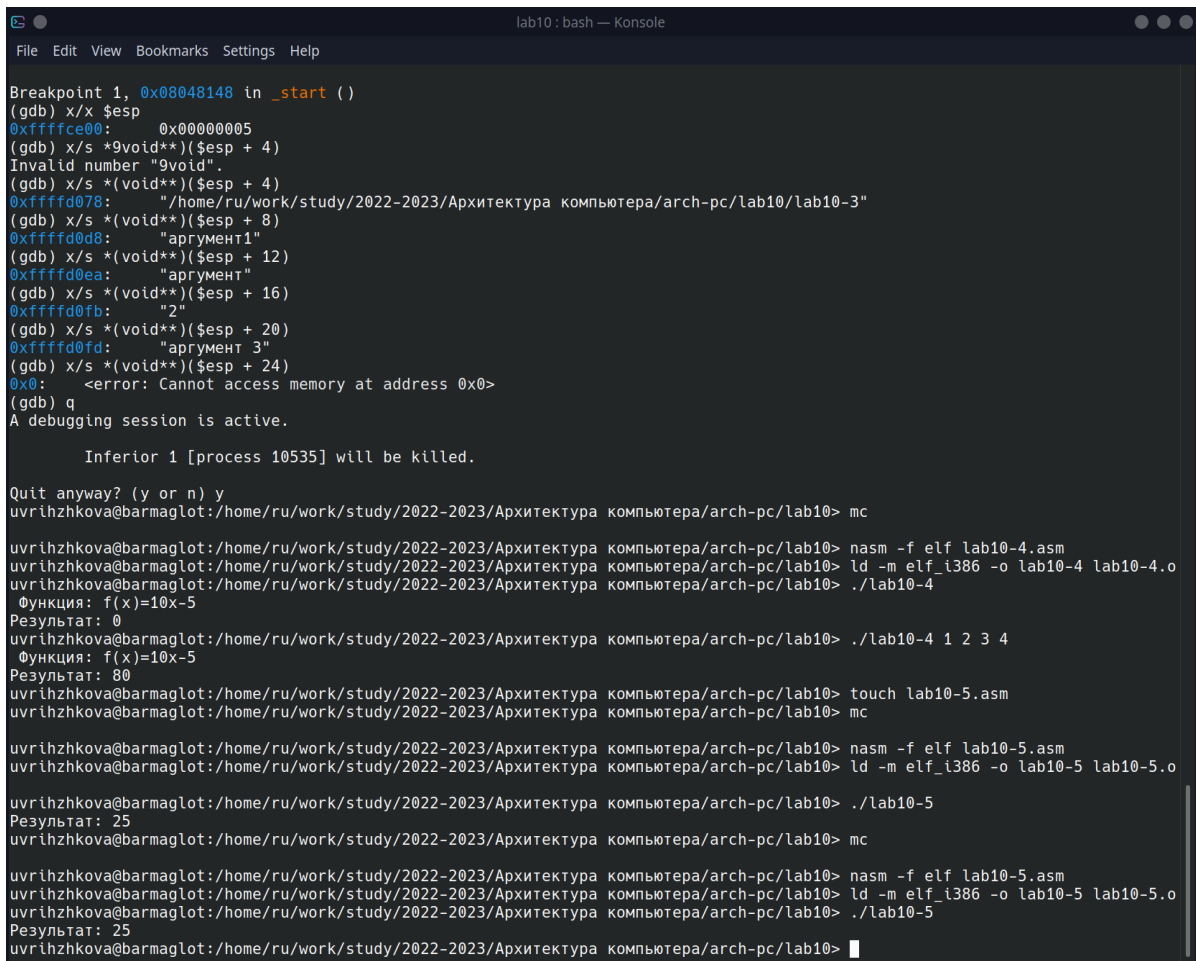
Breakpoint 1, 0x8048148 in _start ()
(gdb) x/x $esp
0xfffffce00: 0x00000005
(gdb) x/s *9void**($esp + 4)
Invalid number "9void".
(gdb) x/s *(void**)( $esp + 4)
0xfffffd078: "/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10/lab10-3"
(gdb) x/s *(void**)( $esp + 8)
0xfffffd0d8: "аргумент1"
(gdb) x/s *(void**)( $esp + 12)
0xfffffd0ea: "аргумент"
(gdb) x/s *(void**)( $esp + 16)
0xfffffd0fb: "2"
(gdb) x/s *(void**)( $esp + 20)
0xfffffd0fd: "аргумент 3"
(gdb) x/s *(void**)( $esp + 24)
0x0: <error: Cannot access memory at address 0x0>
(gdb) q
A debugging session is active.

    Inferior 1 [process 10535] will be killed.

Quit anyway? (y or n) y
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> mc
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> nasm -f elf lab10-4.asm
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> ld -m elf_i386 -o lab10-4 lab10-4.o
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> ./lab10-4
Функция: f(x)=10x-5
Результат: 0
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> ./lab10-4 1 2 3 4
Функция: f(x)=10x-5
Результат: 80
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10>
```

2. Предложенный код выводит ошибку, с помощью gdb и функций X/NFU посмотрим содержание регистра умножения, ещё надо поставить на нем брэйкпоинт, заметим, что в нем изменяется eax, а суммируем мы с ebx и

выводим значение в `ebx`, поэтому заменим в суммирование `ebx` на `eax` и получим правильный ответ 25.



```
lab10: bash — Konsole
File Edit View Bookmarks Settings Help

Breakpoint 1, 0x08048148 in _start ()
(gdb) x/x $esp
0xfffffce00: 0x00000005
(gdb) x/s *9void**($esp + 4)
Invalid number "9void".
(gdb) x/s *(void**)(esp + 4)
0xfffffd078: "/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10/lab10-3"
(gdb) x/s *(void**)(esp + 8)
0xfffffd0d8: "аргумент1"
(gdb) x/s *(void**)(esp + 12)
0xfffffd0ea: "аргумент"
(gdb) x/s *(void**)(esp + 16)
0xfffffd0fb: "2"
(gdb) x/s *(void**)(esp + 20)
0xfffffd0fd: "аргумент 3"
(gdb) x/s *(void**)(esp + 24)
0x0: <error: Cannot access memory at address 0x0>
(gdb) q
A debugging session is active.

    Inferior 1 [process 10535] will be killed.

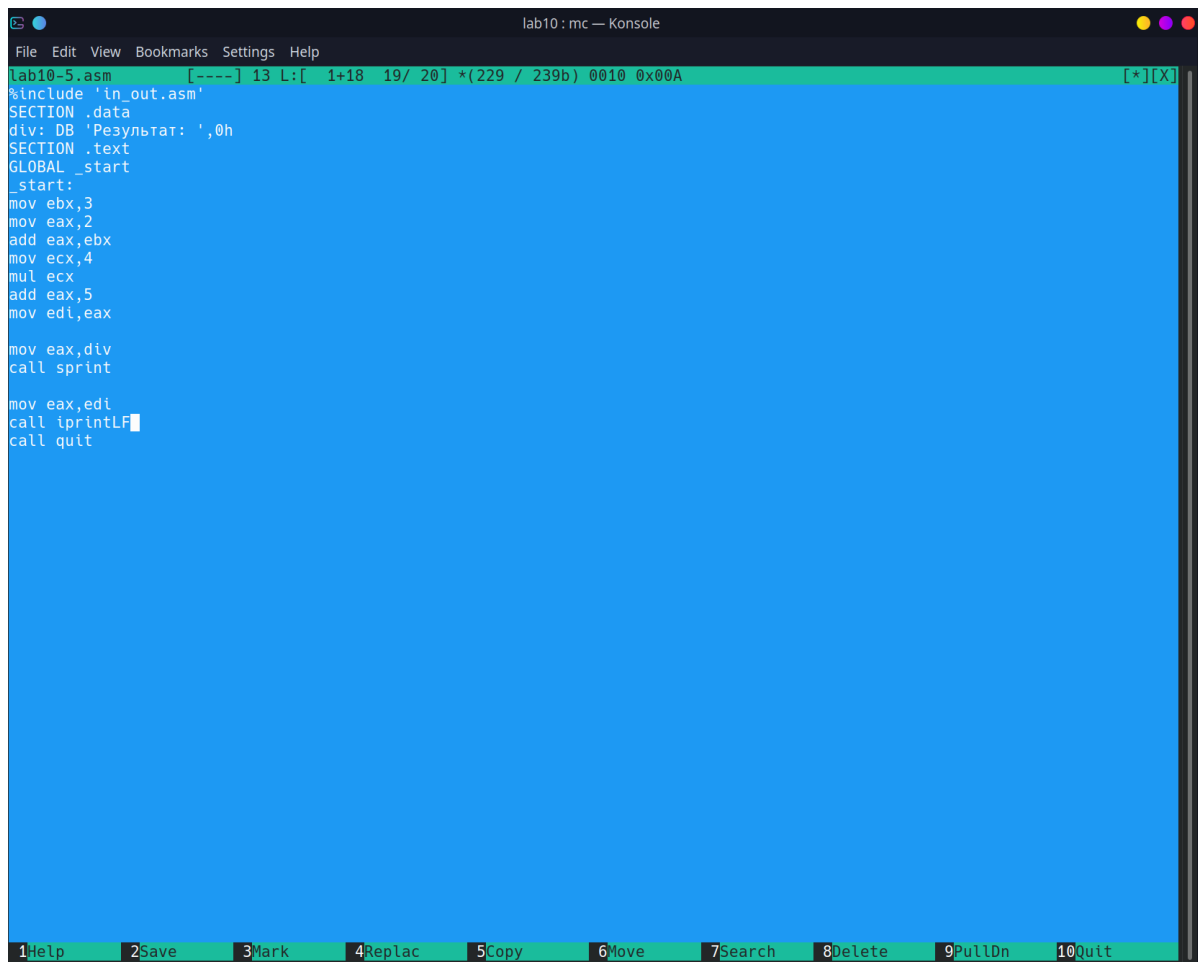
Quit anyway? (y or n) y
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> mc

uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> nasm -f elf lab10-4.asm
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> ld -m elf_i386 -o lab10-4 lab10-4.o
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> ./lab10-4
Функция: f(x)=10x-5
Результат: 0
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> ./lab10-4 1 2 3 4
Функция: f(x)=10x-5
Результат: 80
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> touch lab10-5.asm
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> mc

uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> nasm -f elf lab10-5.asm
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> ld -m elf_i386 -o lab10-5 lab10-5.o
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> ./lab10-5
Результат: 25
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> mc

uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> nasm -f elf lab10-5.asm
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> ld -m elf_i386 -o lab10-5 lab10-5.o
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> ./lab10-5
Результат: 25
uvrihzhkova@barmaglot:/home/ru/work/study/2022-2023/Архитектура компьютера/arch-pc/lab10> █
```

Рис. 3.1: Исправленный вывод



```
lab10-5.asm [-----] 13 L: [ 1+18 19/ 20] *(229 / 239b) 0010 0x00A [*][X]
File Edit View Bookmarks Settings Help
%include 'in_out.asm'
SECTION .data
div: DB 'Результат: ',0h
SECTION .text
GLOBAL _start
_start:
mov ebx,3
mov eax,2
add eax,ebx
mov ecx,4
mul ecx
add eax,5
mov edi,eax

mov eax,div
call sprint

mov eax,edi
call iprintLF
call quit

1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit
```

Рис. 3.2: Исправленный код

4 Выводы

В данной работе мы познакомились с отладчиком и с помощью него научились изменять программу.