

Лабораторная работа №8

**Команды безусловного и условного переходов в Nasm.
Программирование ветвлений.**

Рассолова Маргарита

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	7
4	Самостоятельная работа (Вариант 9)	12
5	Выводы	15
	Список литературы	16

Список иллюстраций

3.1	Создание каталога и файла	7
3.2	Ввод текста	7
3.3	Создание объектного файла и его запуск	8
3.4	Изменение текста	8
3.5	Создание объектного файла и его запуск	8
3.6	Изменение текста	9
3.7	Создание объектного файла и его запуск	9
3.8	Создание второго файла	9
3.9	Текст программы	10
3.10	Создание объектного файла и проверка его работы	10
3.11	Создание файла листинга и его открытие	10
3.12	push eax	11
3.13	pop eax	11
3.14	dec ecx	11
3.15	Удаление одного из операндов	11
3.16	Ошибка	11
4.1	Создание файла	12
4.2	Текст программы	12
4.3	Создание объектного файла и его запуск	13
4.4	Создание файла	13
4.5	Текст программы	14
4.6	Создание объектного файла и его запуск	14

Список таблиц

1 Цель работы

Изучение команд условного и безусловного переходов. Приобретение навыков написания программ с использованием переходов. Знакомство с назначением и структурой файла листинга.

2 Теоретическое введение

Безусловный переход — это переход, который выполняется всегда. Безусловный переход осуществляется с помощью команды JMP. У этой команды один операнд, который может быть непосредственным адресом, регистром или ячейкой памяти, содержащей адрес. Условный переход осуществляется, если выполняется определённое условие, заданное флагами процессора. Состояние флагов изменяется после выполнения арифметических, логических и некоторых других команд. Если условие не выполняется, то управление переходит к следующей команде.

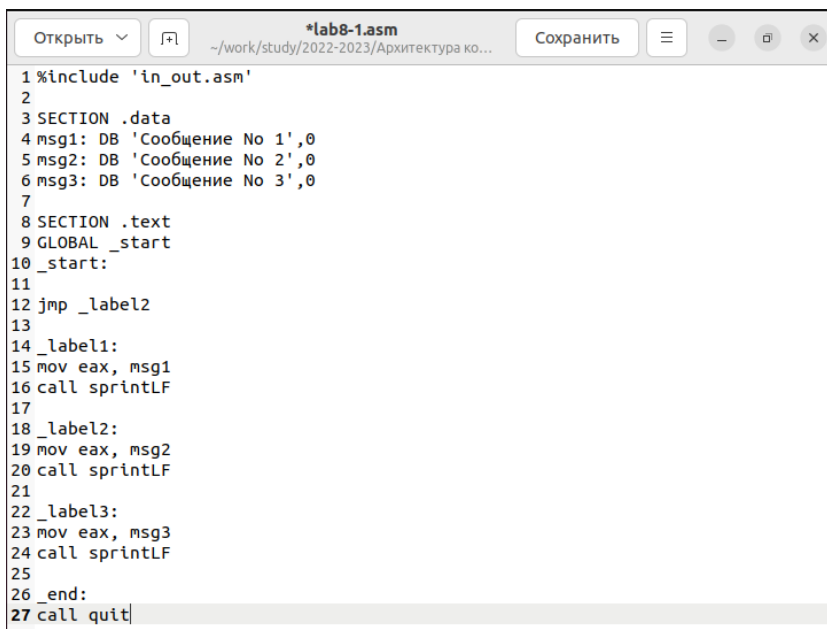
3 Выполнение лабораторной работы

1. Создала каталог для восьмой лабораторной работы и файл к ней. (рис. 3.1)

```
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a
/arch-pc$ mkdir lab08
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc$ cd lab08
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ touch lab8-1.asm
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ ls
lab8-1.asm
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ pwd
/home/mikotseruba/work/study/2022-2023/Архитектура компьютера/arch-pc/lab08
```

Рис. 3.1: Создание каталога и файла

2. Ввела в созданный файл текст программы из листинга 8-1. (рис. 3.2)



The screenshot shows a text editor window titled '*lab8-1.asm' with the file path '~/work/study/2022-2023/Архитектура ко...'. The editor contains the following assembly code:

```
1 %include 'in_out.asm'
2
3 SECTION .data
4 msg1: DB 'Сообщение No 1',0
5 msg2: DB 'Сообщение No 2',0
6 msg3: DB 'Сообщение No 3',0
7
8 SECTION .text
9 GLOBAL _start
10 _start:
11
12 jmp _label2
13
14 _label1:
15 mov eax, msg1
16 call sprintf
17
18 _label2:
19 mov eax, msg2
20 call sprintf
21
22 _label3:
23 mov eax, msg3
24 call sprintf
25
26 _end:
27 call quit
```

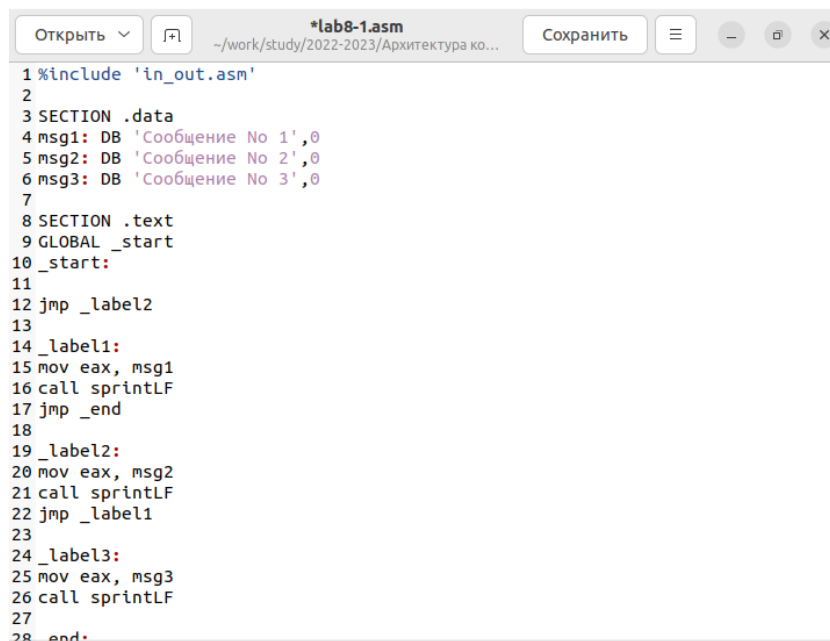
Рис. 3.2: Ввод текста

3. Создала объектный файл и запустила его. (рис. 3.3)

```
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ nasm -f elf lab8-1.asm
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ ./lab8-1
Сообщение No 2
Сообщение No 3
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
```

Рис. 3.3: Создание объектного файла и его запуск

4. Изменила текст программы в соответствии с листингом 8-2. (рис. 3.4)



```
Открыть  [F] *lab8-1.asm Сохранить [Ctrl+S] [Ctrl+Z] [X]
~/work/study/2022-2023/Архитектура ко...

1 %include 'in_out.asm'
2
3 SECTION .data
4 msg1: DB 'Сообщение No 1',0
5 msg2: DB 'Сообщение No 2',0
6 msg3: DB 'Сообщение No 3',0
7
8 SECTION .text
9 GLOBAL _start
10 _start:
11
12 jmp _label2
13
14 _label1:
15 mov eax, msg1
16 call sprintf
17 jmp _end
18
19 _label2:
20 mov eax, msg2
21 call sprintf
22 jmp _label1
23
24 _label3:
25 mov eax, msg3
26 call sprintf
27
28 _end:
```

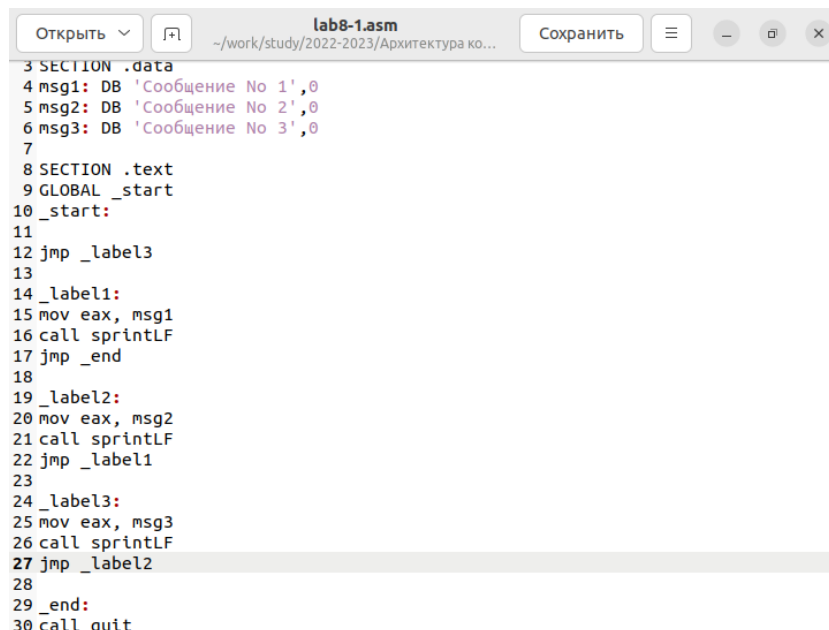
Рис. 3.4: Изменение текста

5. Создала объектный файл и запустила его. (рис. 3.5)

```
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ nasm -f elf lab8-1.asm
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ ./lab8-1
Сообщение No 2
Сообщение No 1
```

Рис. 3.5: Создание объектного файла и его запуск

6. Изменила текст программы снова. (рис. 3.6)

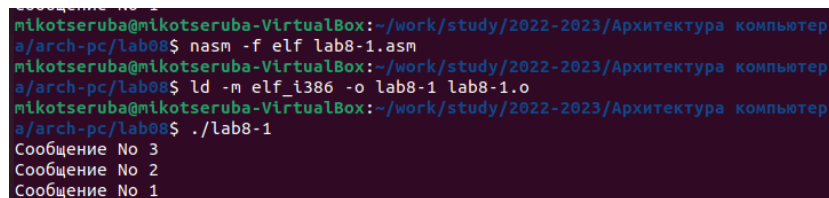


```
lab8-1.asm
~/work/study/2022-2023/Архитектура ко...
Сохранить

3 SECTION .data
4 msg1: DB 'Сообщение No 1',0
5 msg2: DB 'Сообщение No 2',0
6 msg3: DB 'Сообщение No 3',0
7
8 SECTION .text
9 GLOBAL _start
10 _start:
11
12 jmp _label3
13
14 _label1:
15 mov eax, msg1
16 call sprintf
17 jmp _end
18
19 _label2:
20 mov eax, msg2
21 call sprintf
22 jmp _label1
23
24 _label3:
25 mov eax, msg3
26 call sprintf
27 jmp _label2
28
29 _end:
30 call quit
```

Рис. 3.6: Изменение текста

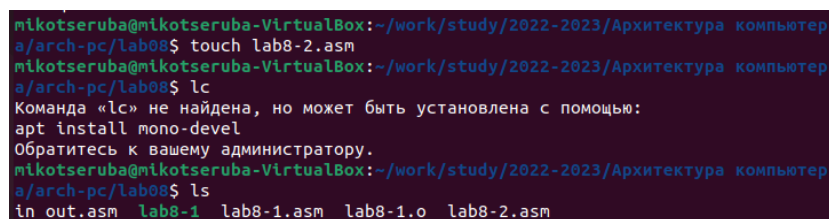
7. Создала объектный файл и запустила его. (рис. 3.7)



```
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ nasm -f elf lab8-1.asm
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ ./lab8-1
Сообщение No 3
Сообщение No 2
Сообщение No 1
```

Рис. 3.7: Создание объектного файла и его запуск

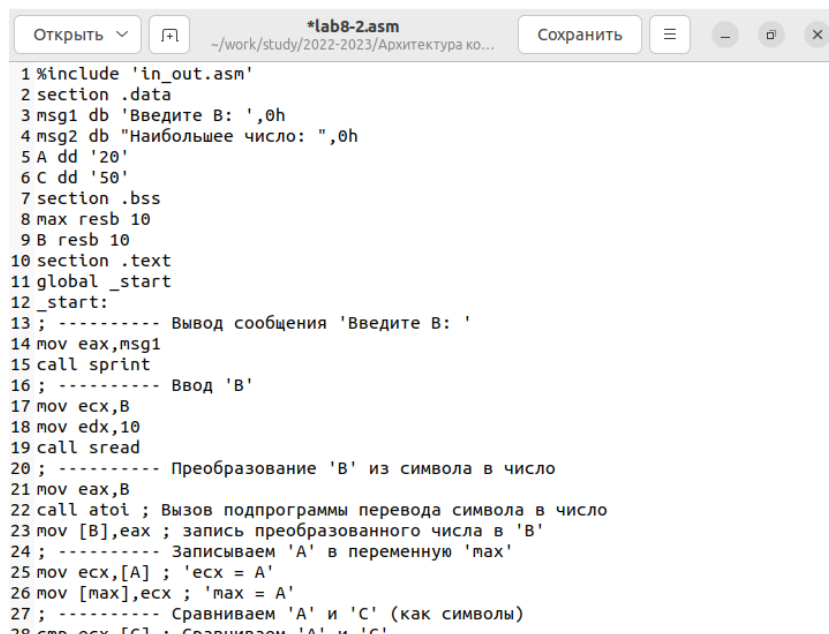
8. Создала файл lab8-2. (рис. 3.8)



```
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ touch lab8-2.asm
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ lc
Команда «lc» не найдена, но может быть установлена с помощью:
apt install mono-devel
Обратитесь к вашему администратору.
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ ls
in_out.asm lab8-1 lab8-1.asm lab8-1.o lab8-2.asm
```

Рис. 3.8: Создание второго файла

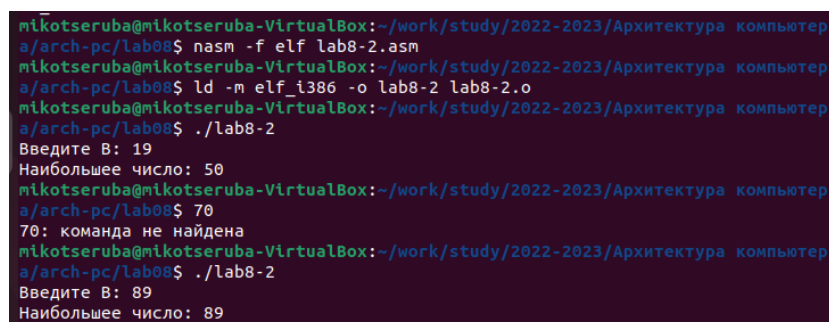
9. Вставила в новый файл текст программы из листинга 8-3. (рис. 3.9)



```
*lab8-2.asm
1 %include 'in_out.asm'
2 section .data
3 msg1 db 'Введите В: ',0h
4 msg2 db "Наибольшее число: ",0h
5 A dd '20'
6 C dd '50'
7 section .bss
8 max resb 10
9 B resb 10
10 section .text
11 global _start
12 _start:
13 ; ----- Вывод сообщения 'Введите В: '
14 mov eax,msg1
15 call sprint
16 ; ----- Ввод 'В'
17 mov ecx,B
18 mov edx,10
19 call sread
20 ; ----- Преобразование 'В' из символа в число
21 mov eax,B
22 call atoi ; Вызов подпрограммы перевода символа в число
23 mov [B],eax ; запись преобразованного числа в 'В'
24 ; ----- Записываем 'А' в переменную 'max'
25 mov ecx,[A] ; 'ecx = A'
26 mov [max],ecx ; 'max = A'
27 ; ----- Сравниваем 'А' и 'С' (как символы)
28 cmp ecx,[C] ; Сравниваем 'А' и 'С'
```

Рис. 3.9: Текст программы

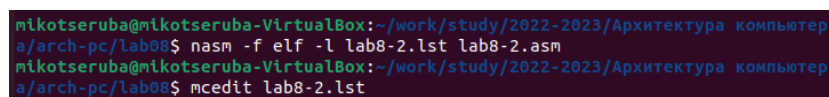
10. Создала объектный файл и проверила его работу. (рис. 3.10)



```
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ nasm -f elf lab8-2.asm
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ ld -m elf_i386 -o lab8-2 lab8-2.o
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ ./lab8-2
Введите В: 19
Наибольшее число: 50
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ 70
70: команда не найдена
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ ./lab8-2
Введите В: 89
Наибольшее число: 89
```

Рис. 3.10: Создание объектного файла и проверка его работы

11. Создала файл листинга для программы из файла lab8-2.asm. Открыла его с помощью текстового редактора mcedit. (рис. 3.11)



```
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ nasm -f elf -l lab8-2.lst lab8-2.asm
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ mcedit lab8-2.lst
```

Рис. 3.11: Создание файла листинга и его открытие

12. 27 строка. Адрес 00000012. Машинный код 50. push eax (исходный текст программы) - выделяет место наверху стека и помещает туда значение из регистра eax. (рис. 3.12)

```
27 00000012 50          <1>    push    eax
```

Рис. 3.12: push eax

13. 55 строка. Адрес 00000040. Машинный код 58. pop eax (исходный текст программы) - переносит любые данные из верхней части стека в eax и освобождает эту область памяти. (рис. 3.13)

```
55 00000040 58          <1>    pop     eax
```

Рис. 3.13: pop eax

14. 95 строка. Адрес 00000073. Машинный код 49. dec esx (исходный текст программы) - уменьшает значение esx на единицу. (рис. 3.14)

```
95 00000073 49          <1>    dec    esx
```

Рис. 3.14: dec esx

15. В 34 строке удалила max. (рис. 3.15)

```
34 mov eax,max
```

Рис. 3.15: Удаление одного из операндов

16. При попытке создать файл листинга выдалась ошибка. (рис. 3.16)

```
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ nasm -f elf -l lab8-2.lst lab8-2.asm
lab8-2.asm:34: error: invalid combination of opcode and operands
```

Рис. 3.16: Ошибка

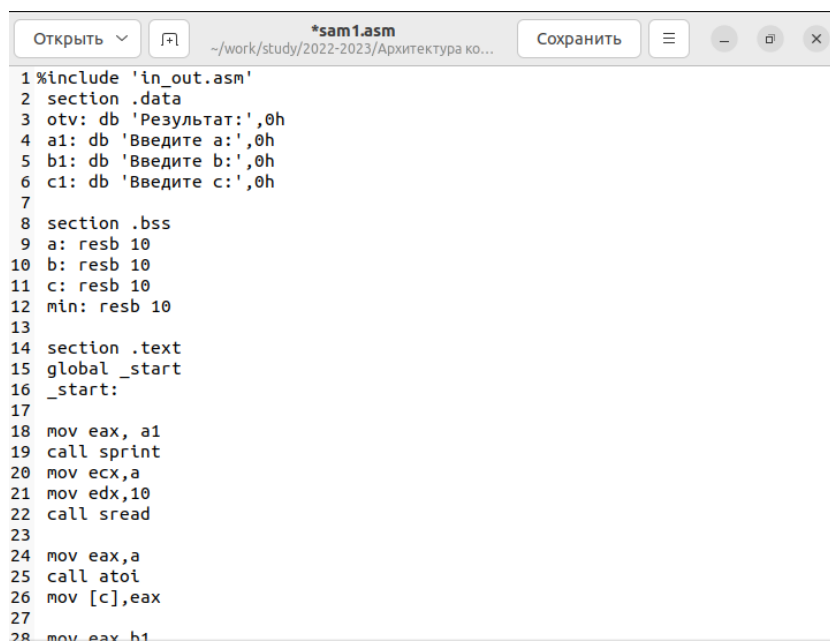
4 Самостоятельная работа (Вариант 9)

17. Создала файл для первого задания. (рис. 4.1)

```
nikotseruba@nikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютера  
a  
/arch-pc/lab08$ touch sam1.asm
```

Рис. 4.1: Создание файла

18. Написала к нему программу. (рис. 4.2)



```
*sam1.asm  
1 %include 'in_out.asm'  
2 section .data  
3 otv: db 'Результат:',0h  
4 a1: db 'Введите a:',0h  
5 b1: db 'Введите b:',0h  
6 c1: db 'Введите c:',0h  
7  
8 section .bss  
9 a: resb 10  
10 b: resb 10  
11 c: resb 10  
12 min: resb 10  
13  
14 section .text  
15 global _start  
16 _start:  
17  
18 mov eax, a1  
19 call sprint  
20 mov ecx, a  
21 mov edx, 10  
22 call sread  
23  
24 mov eax, a  
25 call atoi  
26 mov [c], eax  
27  
28 mov eax, b1
```

Рис. 4.2: Текст программы

19. Создала объектный файл и запустила программу. (рис. 4.3)

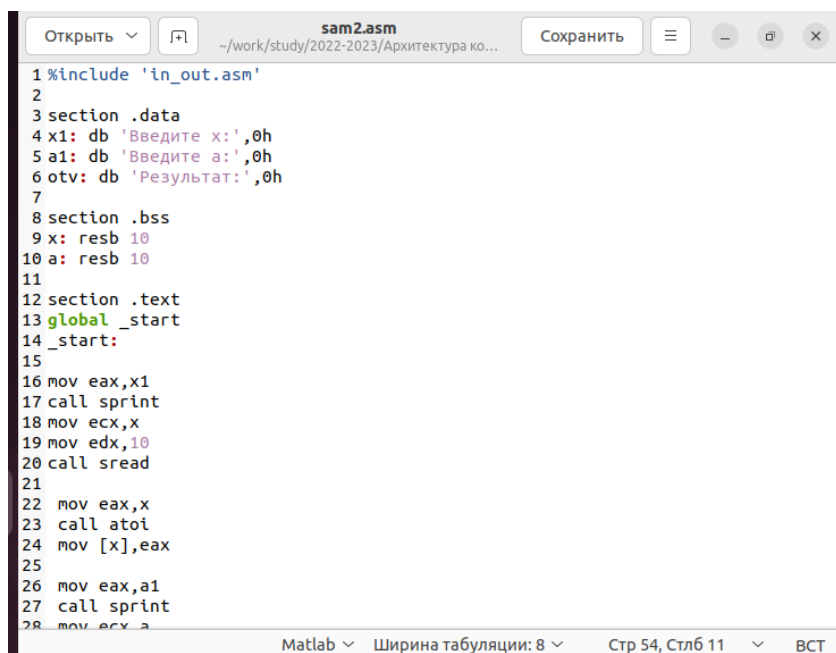
```

mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ nasm -f elf sam1.asm
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ ld -m elf_i386 -o sam1 sam1.o
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ ./sam1
Введите a:24
Введите b:98
Введите c:15
Результат:15

```

Рис. 4.3: Создание объектного файла и его запуск

20. Создала файл для второго задания. (рис. 4.4)



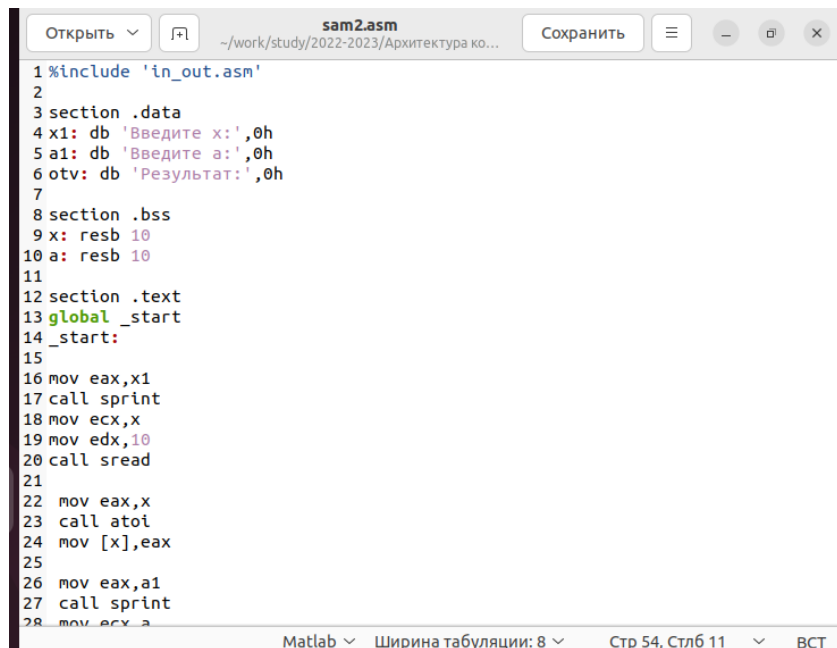
```

1 %include 'in_out.asm'
2
3 section .data
4 x1: db 'Введите x:',0h
5 a1: db 'Введите a:',0h
6 otv: db 'Результат:',0h
7
8 section .bss
9 x: resb 10
10 a: resb 10
11
12 section .text
13 global _start
14 _start:
15
16 mov eax,x1
17 call sprint
18 mov ecx,x
19 mov edx,10
20 call sread
21
22 mov eax,x
23 call atoi
24 mov [x],eax
25
26 mov eax,a1
27 call sprint
28 mov ecx,a

```

Рис. 4.4: Создание файла

21. Написала программу. (рис. 4.5)

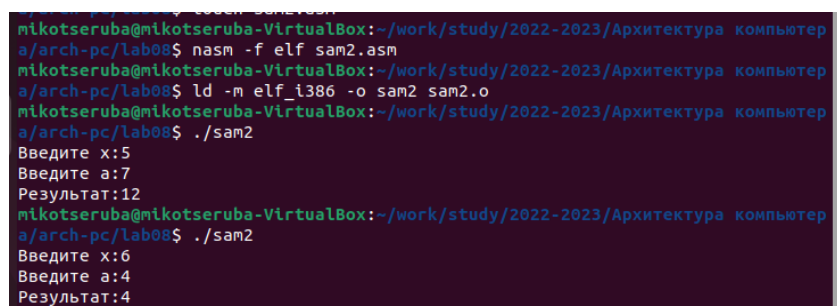
A screenshot of a text editor window titled 'sam2.asm'. The window has a menu bar with 'Открыть', 'Сохранить', and a hamburger menu icon. The file path is '~/.work/study/2022-2023/Архитектура ко...'. The code is as follows:

```
1 %include 'in_out.asm'
2
3 section .data
4 x1: db 'Введите x:',0h
5 a1: db 'Введите a:',0h
6 otv: db 'Результат:',0h
7
8 section .bss
9 x: resb 10
10 a: resb 10
11
12 section .text
13 global _start
14 _start:
15
16 mov eax,x1
17 call sprint
18 mov ecx,x
19 mov edx,10
20 call sread
21
22 mov eax,x
23 call atoi
24 mov [x],eax
25
26 mov eax,a1
27 call sprint
28 mov ecx,a
```

The status bar at the bottom shows 'Matlab', 'Ширина табуляции: 8', 'Стр 54, Стлб 11', and 'ВСТ'.

Рис. 4.5: Текст программы

22. Создала объектный файл и запустила программу. (рис. 4.6)

A screenshot of a terminal window showing the compilation and execution of the assembly program. The commands and output are as follows:

```
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ nasm -f elf sam2.asm
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ ld -m elf_i386 -o sam2 sam2.o
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ ./sam2
Введите x:5
Введите a:7
Результат:12
mikotseruba@mikotseruba-VirtualBox:~/work/study/2022-2023/Архитектура компьютер
a/arch-pc/lab08$ ./sam2
Введите x:6
Введите a:4
Результат:4
```

Рис. 4.6: Создание объектного файла и его запуск

5 Выводы

Изучила команды условного и безусловного переходов. Приобрела навык написания программ с использованием переходов. Познакомилась с назначением и структурой файла листинга.

Список литературы