

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №7

дисциплина: Архитектура компьютера

Студент: Володина Алиса Алексеевна

Группа: НКАбд-01-25

МОСКВА

2025 г.

Оглавление

Список таблиц 1 Цель работы	4
2 Задания	6
3 Теоретическое введение	7
4 Выполнение работы	8
Реализация переходов в NASM	8
Изучение структуры файлы листинга	13
Объяснение строк листинга	13
Задание для самостоятельной работы	15
5 Выводы	19
Список литературы	20

Список изображений

Рисунок 0.1 (создание каталога)	7
Рисунок 0.2 (переход в каталог)	7
Рисунок 0.3 (создаем файл)	7
Рисунок 0.4 (текст программы)	7
Рисунок 0.5 (создание исполняемого файла)	8
Рисунок 0.6 (запуск исполняемого файла)	8
Рисунок 0.7 (результат работы файла)	8
Рисунок 0.8 (измененный текст программы)	8
Рисунок 0.9 (создание и запуск исполняемого файла)	9
Рисунок 0.10 (измененный текст программы)	9
Рисунок 0.11 (создание исполняемого файла)	9
Рисунок 0.12(запуск исполняемого файла)	10
Рисунок 0.13(создание файла)	10
Рисунок 0.14 (текст программы)	10
Рисунок 0.15 (текст программы)	11
Рисунок 0.16 (создание исполняемого файла и проверка работы)	11
Рисунок 0.17(создание исполняемого файла и проверка работы)	11
Рисунок 0.18(создание исполняемого файла и проверка работы)	11
Рисунок 0.19 (создание файла листинга)	12
Рисунок 0.20 (открытие листинга)	12
Рисунок 0.21 (открытие листинга)	12
Рисунок 0.22 (текст программы с удалением операнды)	13
Рисунок 0.23 (создание файла листинга)	13
Рисунок 0.24 (открытие листинга)	13
Рисунок 0.25 (текст программы)	14
Рисунок 0.26 (текст программы)	14
Рисунок 0.27 (текст программы)	14
Рисунок 0.28 (создание исполняемого файла и проверка его работы)	15
Рисунок 0.29 (текст программы)	15
Рисунок 0.30 (текст программы)	16
Рисунок 0.31 (создание исполняемого файла)	16
Рисунок 0.32 (создание исполняемого файла)	16
Рисунок 0.33 (запуск программы)	16
Рисунок 0.34 (проверка результатов)	16
Рисунок 0.35 (проверка результатов)	16

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 Задания

1. Реализация переходов в NASM
2. Изучение структуры файлов листинга
3. Самостоятельное написание программ по материалам лабораторной работы

3 Теоретическое введение

Для реализации ветвлений в ассемблере используются так называемые команды передачи управления или команды перехода. Можно выделить 2 типа переходов:

- условный переход–выполнение или не выполнение перехода в определенную точку программы в зависимости от проверки условия.
- безусловный переход–выполнение передачи управления в определенную точку программы без каких-либо условий.

4 Выполнение работы

Реализация переходов в NASM

Создадим каталог для программ лабораторной работы № 7, перейдем в него и создадим файл lab7-1.asm (рисунок 0.1-0.3)

```
aavolodina@fedora:~$ mkdir ~/work/arch-pc/lab07
aavolodina@fedora:~$
```

Рисунок 0.1 (создание каталога)

```
aavolodina@fedora:~$ cd ~/work/arch-pc/lab07
aavolodina@fedora:~/work/arch-pc/lab07$
```

Рисунок 0.2 (переход в каталог)

```
aavolodina@fedora:~/work/arch-pc/lab07$ touch lab7-1.asm
aavolodina@fedora:~/work/arch-pc/lab07$
```

Рисунок 0.3 (создаем файл)

Введем в файл lab7-1.asm текст программы из листинга (рисунок 0.4)

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

Рисунок 0.4 (текст программы)

Создадим исполняемый файл и запустим его (рисунок 0.5-0.6)


```
aavolodina@fedora:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
aavolodina@fedora:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
aavolodina@fedora:~/work/arch-pc/lab07$
```

Рисунок 0.5 (создание исполняемого файла)

```
aavolodina@fedora:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
aavolodina@fedora:~/work/arch-pc/lab07$ ./lab7-1
```

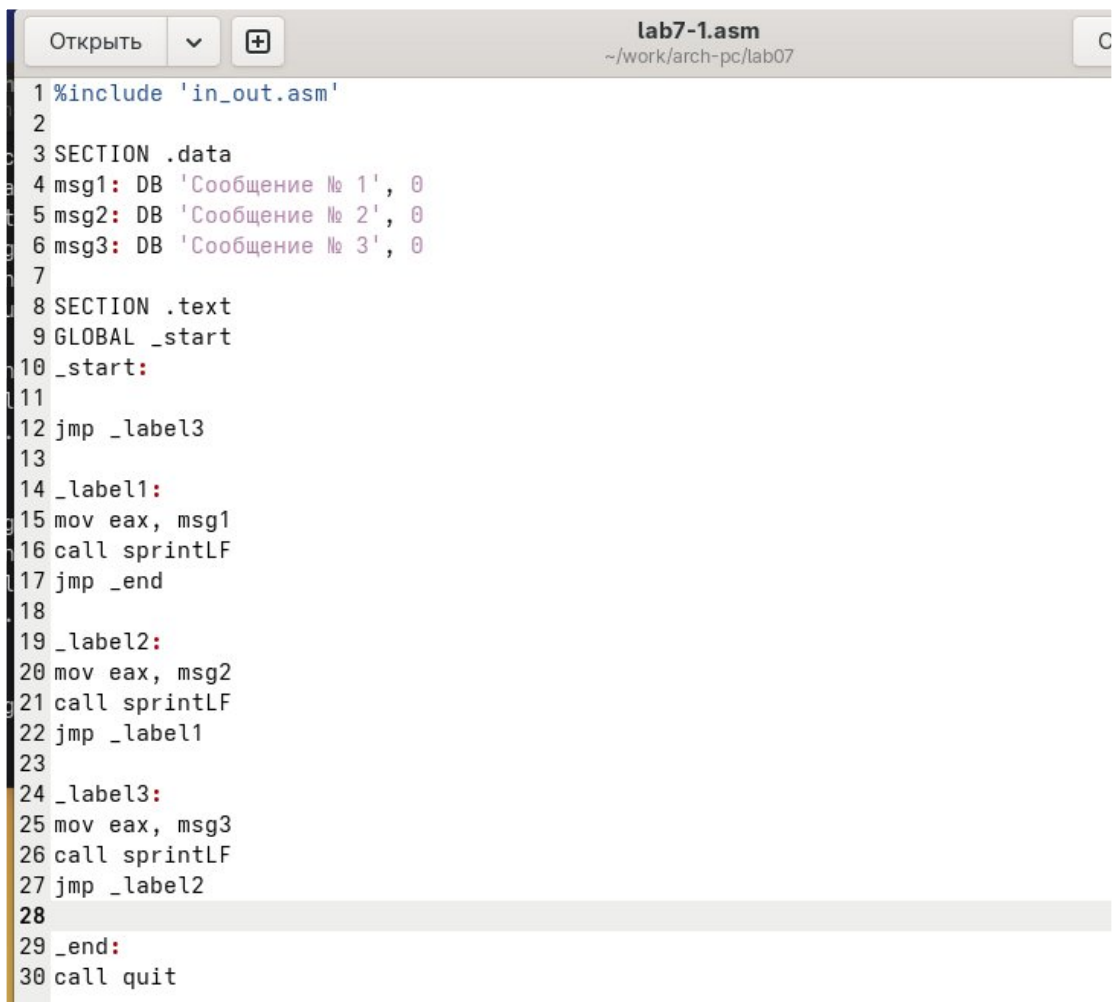
Рисунок 0.6 (запуск исполняемого файла)

Результат работы данной программы будет следующим (рисунок 0.7)

```
aavolodina@fedora:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 3
aavolodina@fedora:~/work/arch-pc/lab07$
```

Рисунок 0.7 (результат работы файла)

Изменим текст программы в соответствии с листингом (рисунок 0.8)



```
lab7-1.asm
~/work/arch-pc/lab07

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

Рисунок 0.8 (измененный текст программы)

Создадим исполняемый файл и проверим его работу (рисунок 0.9)

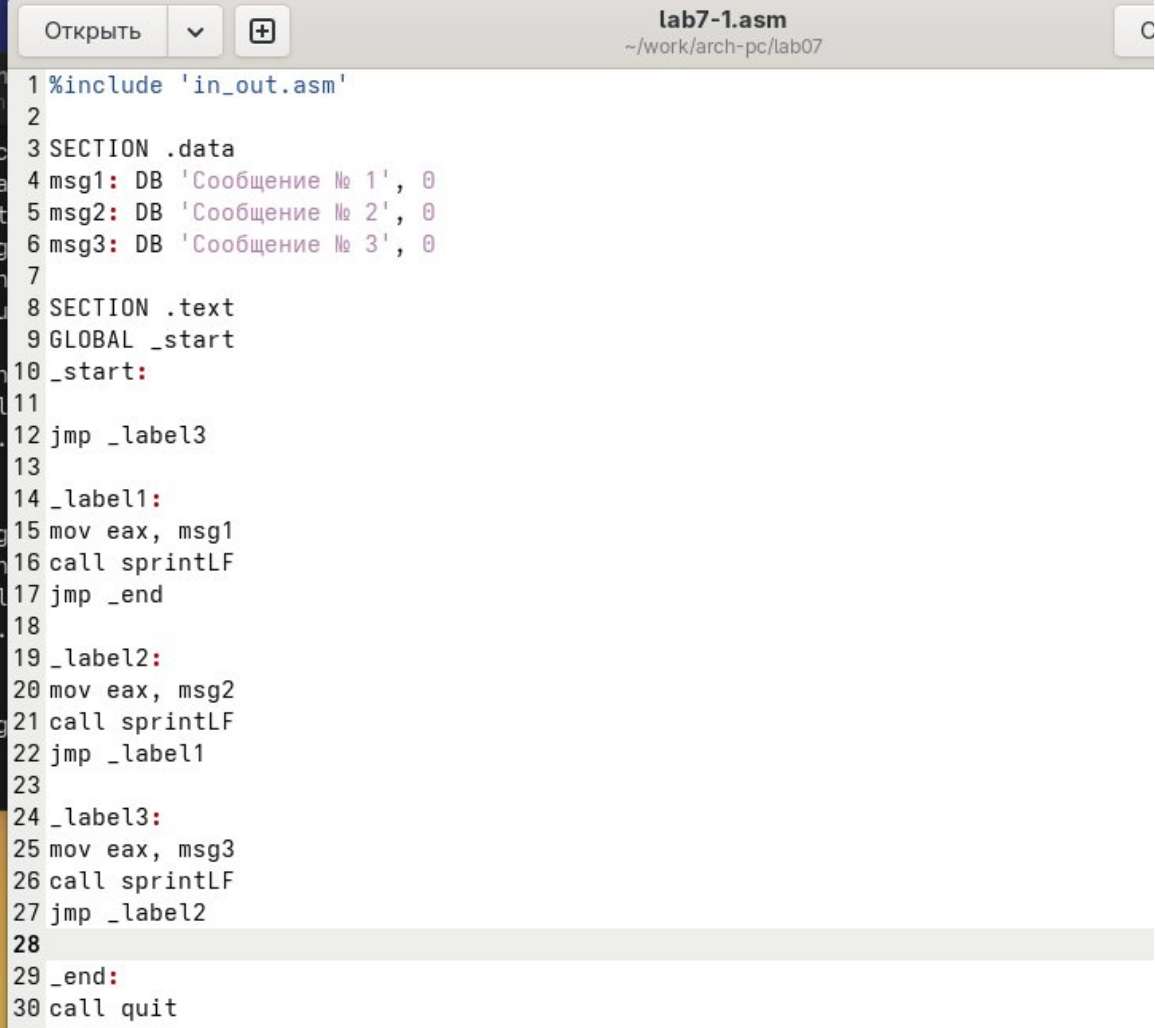
```

aavolodina@fedora:~/work/arch-pc/lab07$ gedit lab7-1.asm
aavolodina@fedora:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
aavolodina@fedora:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
aavolodina@fedora:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 1

```

Рисунок 0.9 (создание и запуск исполняемого файла)

Изменим текст программы добавив или изменив инструкции `jmp`, чтобы вывод программы был следующим (рисунок 0.10)



```

lab7-1.asm
~/work/arch-pc/lab07
1 %include 'in_out.asm'
2
3 SECTION .data
4 msg1: DB 'Сообщение № 1', 0
5 msg2: DB 'Сообщение № 2', 0
6 msg3: DB 'Сообщение № 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

```

Рисунок 0.10 (измененный текст программы)

```

aavolodina@fedora:~/work/arch-pc/lab07$ gedit lab7-1.asm
aavolodina@fedora:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
aavolodina@fedora:~/work/arch-pc/lab07$

```

Рисунок 0.11 (создание исполняемого файла)

```

aavolodina@fedora:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1
aavolodina@fedora:~/work/arch-pc/lab07$

```

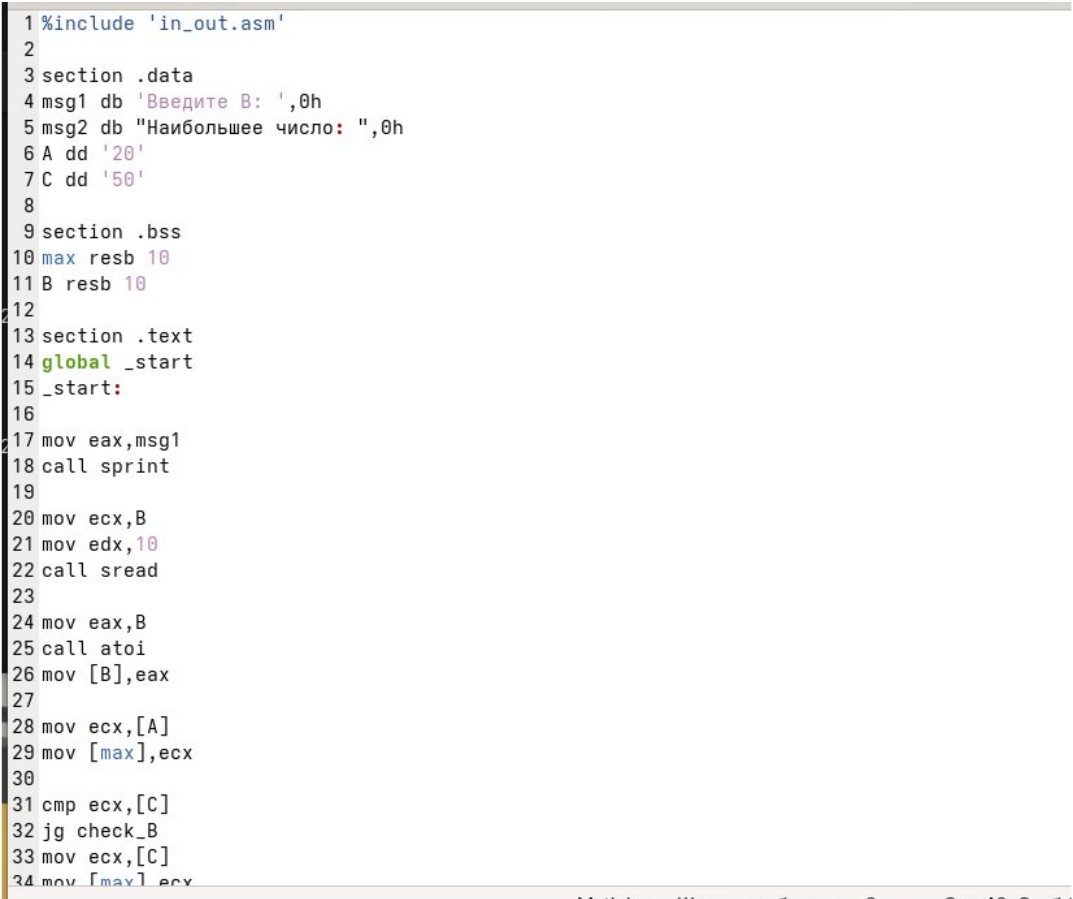
Рисунок 0.12(запуск исполняемого файла)

Создадим файл lab7-2.asm в каталоге ~/work/arch-pc/lab07. Внимательно изучим текст программы из листинга 7.3 и введем в lab7-2.asm (рисунок 0.13-0.15)



```
aavolodina@fedora:~/work/arch-pc/lab07$ touch lab7-2.asm
```

Рисунок 0.13(создание файла)



```
1 %include 'in_out.asm'
2
3 section .data
4 msg1 db 'Введите B: ',0h
5 msg2 db "Наибольшее число: ",0h
6 A dd '20'
7 C dd '50'
8
9 section .bss
10 max resb 10
11 B resb 10
12
13 section .text
14 global _start
15 _start:
16
17 mov eax,msg1
18 call sprint
19
20 mov ecx,B
21 mov edx,10
22 call sread
23
24 mov eax,B
25 call atoi
26 mov [B],eax
27
28 mov ecx,[A]
29 mov [max],ecx
30
31 cmp ecx,[C]
32 jg check_B
33 mov ecx,[C]
34 mov [max],ecx
```

Рисунок 0.14 (текст программы)

```

21 mov edx,10
22 call sread
23
24 mov eax,B
25 call atoi
26 mov [B],eax
27
28 mov ecx,[A]
29 mov [max],ecx
30
31 cmp ecx,[C]
32 jg check_B
33 mov ecx,[C]
34 mov [max],ecx
35
36 check_B:
37 mov eax,max
38 call atoi
39 mov [max],eax
40
41 mov ecx,[max]
42 cmp ecx,[B]
43 jg fin
44 mov ecx,[B]
45 mov [max],ecx
46
47 fin:
48 mov eax,msg2
49 call sprint
50 mov eax,[max]
51 call iprintLF
52 call quit

```

Matlab ▾ Ширина табуляции: 8 ▾ Стр 46, Стлб 1

Рисунок 0.15 (текст программы)

Создадим исполняемый файл и проверим его работу для разных значений В (рисунок 0.16-0.18)

```

aavolodina@fedora:~/work/arch-pc/lab07$ gedit lab7-2.asm
aavolodina@fedora:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
aavolodina@fedora:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
aavolodina@fedora:~/work/arch-pc/lab07$ ./lab7-2
Введите В: 25
Наибольшее число: 50
aavolodina@fedora:~/work/arch-pc/lab07$

```

Рисунок 0.16 (создание исполняемого файла и проверка работы)

```

aavolodina@fedora:~/work/arch-pc/lab07$ ./lab7-2
Введите В: 25
Наибольшее число: 50
aavolodina@fedora:~/work/arch-pc/lab07$ ./lab7-2
Введите В: 80
Наибольшее число: 80
aavolodina@fedora:~/work/arch-pc/lab07$ 

```

Рисунок 0.17(создание исполняемого файла и проверка работы)

```

aavolodina@fedora:~/work/arch-pc/lab07$ ./lab7-2
Введите В: 5
Наибольшее число: 50
aavolodina@fedora:~/work/arch-pc/lab07$

```

Рисунок 0.18(создание исполняемого файла и проверка работы)

Изучение структуры файлы листинга

Создадим файл листинга для программы из файла lab7-2.asm (рисунок 0.19-


0.21)

```
aavolodina@fedora:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
aavolodina@fedora:~/work/arch-pc/lab07$
```

Рисунок 0.19 (создание файла листинга)

```
aavolodina@fedora:~/work/arch-pc/lab07$ mcedit lab7-2.lst
```

Рисунок 0.20 (открытие листинга)



```
lab7-2.lst      [----]  0 L:[ 1+ 0  1/228] *(0  /13398b) 0032 0x020 [*][X]
1                                     %include 'in_out.asm'
1                                     <1> ;----- slen -----
2                                     <1> ; Функция вычисления длины сообщения
3                                     <1> slen:.....
4 00000000 53                         <1>      push     ebx.....
5 00000001 89C3                       <1>      mov      ebx, eax.....
6                                     <1>.....
7                                     <1> nextchar:.....
8 00000003 803800                     <1>      cmp      byte [eax], 0...
9 00000006 7403                       <1>      jz       finished.....
10 00000008 40                        <1>      inc      eax.....
11 00000009 EBF8                      <1>      jmp      nextchar.....
12                                     <1>.....
13                                     <1> finished:
14 0000000B 29D8                     <1>      sub      eax, ebx
15 0000000D 5B                        <1>      pop      ebx.....
16 0000000E C3                       <1>      ret.....
17                                     <1>.
18                                     <1>.

1Помощь 2Сохран 3Блок 4Замена 5Копия 6Переть 7Поиск 8Удасть 9МенюМС10Выход
```

Рисунок 0.21 (открытие листинга)

Объяснение строк листинга:

Самое первое значение в файле листинга – номер строки—это номер строки файла листинга (нужно помнить, что номер строки в файле листинга может не соответствовать номеру строки в файле с исходным текстом программы)

Второе вхождение- адрес—это смещение машинного кода от начала текущего сегмента

Далее идут машинный код, представляющий собой ассемблированную исходную строку в виде шестнадцатеричной последовательности, и исходный текст программы — это просто строка исходной программы вместе с комментариями

Откроем файл с программой lab7-2.asm и в любой инструкции с двумя операндами

удалим один операнд. Выполним трансляцию с получением файла листинга (рисунок 0.22-0.24)

```
20 mov ecx,B
21 mov edx,10
22 call sread
23
24 mov eax,B
25 call atoi
26 mov [B],eax
27
28 mov ecx,[A]
29 mov [max],ecx
30
31 cmp ecx,[C]
32 jg check_B
33 mov ecx,[C]
34 mov [max],ecx
35
36 check_B:
37 mov eax,max
38 call atoi
39 mov [max],eax
40
41 mov ecx,[max]
42 cmp ecx,[B]
43 jg fin
44 mov ecx,[B]
45 mov [max],ecx
46
47 fin:
48 mov eax,
49 call sprint
50 mov eax,[max]
51 call iprintLF
52 call quit
```

Рисунок 0.22 (текст программы с удалением операнды)

В 48 строке удалим один операнд

```
aavolodina@fedora:~/work/arch-pc/lab07$ gedit lab7-2.asm
aavolodina@fedora:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
lab7-2.asm:48: error: invalid combination of opcode and operands
aavolodina@fedora:~/work/arch-pc/lab07$
```

Рисунок 0.23 (создание файла листинга)

```
45 00000153 890D[00000000]      mov [max],ecx
46 .....
47                               fin:
48                               mov eax,.
48                               *****      error: invalid combination of opcode an
49 00000159 E8B1FEFFFF          call sprint.
50 0000015E A1[00000000]        mov eax,[max]
51 00000163 E81EFFFFFF          call iprintLF
52 00000168 E86EFFFFFF          call quit
```

Рисунок 0.24 (открытие листинга)

В новом файле листинга показывает ошибку, возникшую при попытке трансляции файла. При этом никакие выходные файлы, помимо файла листинга, не появляются

Задание для самостоятельной работы

Напишем программу нахождения наименьшей из 3 целочисленных

переменных a , b и c . Значения переменных 82,59,61 (рисунок 0.25-0.27)

```
%include 'in_out.asm'

SECTION .data
msg1 db 'Введите B: ', 0h
msg2 db 'Наименьшее число: ', 0h
A dd '82'
C dd '61'

SECTION .bss
min resb 10
B resb 10

SECTION .text
GLOBAL _start
_start:

mov eax, msg1
call sprint

mov ecx, B
mov edx, 10
```

Рисунок 0.25 (текст программы)

```
mov ecx, B
mov edx, 10
call sread

mov eax, B
call atoi
mov [B], eax

mov ecx, [A]
mov [min], ecx

cmp ecx, [C]
jg check_B
mov ecx, [C]
mov [min], ecx

check_B:
mov eax, min
call atoi
mov [min], eax
```

Рисунок 0.26 (текст программы)

```
mov [min], eax

mov ecx, [min]
cmp ecx, [B]
jb fin
mov ecx, [B]
mov [min], ecx

fin:
mov eax, msg2
call sprint
mov eax, [min]
call iprintLF
call quit
```

Рисунок 0.27 (текст программы)

Создайте исполняемый файл и проверьте его работу (рисунок 0.28)

```
aavolodina@fedora:~/work/arch-pc/lab07$ nasm -f elf lab7-3.asm
aavolodina@fedora:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-3 lab7-3.o
aavolodina@fedora:~/work/arch-pc/lab07$ ./lab7-3
Введите В: 59
Наименьшее число: 59
aavolodina@fedora:~/work/arch-pc/lab07$
```

Рисунок 0.28 (создание исполняемого файла и проверка его работы)

Напишите программу, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции $f(x)$ и выводит результат вычислений. Вид функции $f(x)$ ($a-1, x < a$;
 $x-1, x \geq a$) (рисунок 0.29-0.30)

```
1 %include 'in_out.asm'
2 SECTION .data
3 msg_x: DB 'Введите значение x: ', 0
4 msg_a: DB 'Введите значение a: ', 0
5 res: DB 'Результат: ', 0
6 SECTION .bss
7 x: RESB 80
8 a: RESB 80
9
10 SECTION .text
11 GLOBAL _start
12 _start:
13
14 mov eax, msg_x
15 call sprint
16
17 mov ecx, x
18 mov edx, 80
19 call sread
20
21 mov eax, x
22 call atoi
23 mov edi, eax
24
25 mov eax, msg_a
26 call sprint
27 mov ecx, a
28 mov edx, 80
29 call sread
30 mov eax, a
31 call atoi
32 mov esi, eax
33
```

Рисунок 0.29 (текст программы)


```

call sprint
mov ecx, a
mov edx, 80
call sread
mov eax, a
call atoi
mov esi, eax

cmp edi, esi
jl case_less
mov eax, edi
sub eax, 1
jmp print_result

case_less:
mov eax, esi
sub eax, 1

print_result:
mov edi, eax
mov eax, res
call sprint
mov eax, edi
call iprintLF
call quit

```

Рисунок 0.30 (текст программы)

Создадим исполняемый файл и проверим его работу для значений x и a (рисунок 0.31-0.35)

```

avolodina@fedora:~/work/arch-pc/lab07$ gedit lab7-4.asm
avolodina@fedora:~/work/arch-pc/lab07$ nasm -f elf lab7-4.asm
avolodina@fedora:~/work/arch-pc/lab07$

```

Рисунок 0.31 (создание исполняемого файла)

```

avolodina@fedora:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-4 lab7-4.o
avolodina@fedora:~/work/arch-pc/lab07$

```

Рисунок 0.32 (создание исполняемого файла)

```

aavolodina@fedora:~/work/arch-pc/lab07$ ./lab7-4
Введите значение x:

```

Рисунок 0.33 (запуск программы)

```

aavolodina@fedora:~/work/arch-pc/lab07$ ./lab7-4
Введите значение x: 5
Введите значение a: 7
Результат: 4
aavolodina@fedora:~/work/arch-pc/lab07$

```

Рисунок 0.34 (проверка результатов)

```

aavolodina@fedora:~/work/arch-pc/lab07$ ./lab7-4
Введите значение x: 6
Введите значение a: 4
Результат: 5
aavolodina@fedora:~/work/arch-pc/lab07$

```

Рисунок 0.35 (проверка результатов)

5 Выводы

Мы изучили команды условного и безусловного перехода, приобрели навыки написания программ с использованием переходов, ознакомились с назначением и структурой файла листинга

Список литературы

1. GDB: The GNU Project Debugger.—URL: <https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual.—2016.—URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center.—2021.—URL: <https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials.—2021.—URL: <https://asmtutor.com/>.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658.—URL: <http://www.amazon.com/Learning-bash-Shell-Programming-Nutshell/dp/0596009658>.
6. Robbins A. Bash Pocket Reference.—O'Reilly Media, 2016.—156 с.—ISBN 978-1491941591.
7. The NASM documentation.—2021.—URL: <https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash.—Packt Publishing, 2017.—502 с.—ISBN 9781784396879.
9. Колдаев В. Д., Lupin C. A. Архитектура ЭВМ.—М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER.—М. : Солон-Пресс, 2017.
11. Новожилов О. П. Архитектура ЭВМ систем.—М.: Юрайт, 2016.
12. Расширенный ассемблер: NASM.—2021.—URL: <https://www.opennet.ru/docs/RUS/nasm/>.
13. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX.—2-е изд.—БХВ Петербург, 2010.—656 с.—ISBN 978-5-94157-538-1.
14. Столяров А. Программирование на языке ассемблера NASM для ОС Unix.—2-е изд.—М. : МАКС Пресс, 2011.—URL: http://www.stolyarov.info/books/asm_unix.
15. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
16. Таненбаум Э., Бос Х. Современные операционные системы.—4-е изд.—СПб.: Питер, 2015. —1120 с.—(Классика Computer Science)