

Отчёт по лабораторной работе 7

Архитектура компьютера

Джеймс НКАбд-05-24

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Реализация переходов в NASM	6
2.2	Изучение структуры файлы листинга	12
2.3	Самостоятельное задание	15
3	Выводы	20

Список иллюстраций

2.1	Создан каталог	6
2.2	Программа lab7-1.asm	7
2.3	Запуск программы lab7-1.asm	7
2.4	Программа lab7-1.asm	8
2.5	Запуск программы lab7-1.asm	9
2.6	Программа lab7-1.asm	10
2.7	Запуск программы lab7-1.asm	10
2.8	Программа lab7-2.asm	11
2.9	Запуск программы lab7-2.asm	12
2.10	Файл листинга lab7-2	13
2.11	Ошибка трансляции lab7-2	14
2.12	Файл листинга с ошибкой lab7-2	15
2.13	Программа lab7-task1.asm	16
2.14	Запуск программы lab7-task1.asm	16
2.15	Программа lab7-task2.asm	18
2.16	Запуск программы lab7-task2.asm	19

Список таблиц

1 Цель работы

Целью работы является изучение команд условного и безусловного переходов. Приобретение навыков написания программ с использованием переходов. Знакомство с назначением и структурой файла листинга.

2 Выполнение лабораторной работы

2.1 Реализация переходов в NASM

Создаю каталог для программ лабораторной работы № 7 и файл lab7-1.asm.
(рис. 2.1)

```
jamespangestu@vbox:~$ cd ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc  
/  
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ mkdir  
~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/lab07  
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ cd ~/w  
ork/study/2024-2025/"Архитектура компьютера"/arch-pc/lab-7  
bash: cd: /home/jamespangestu/work/study/2024-2025/Архитектура компьютера/arch-p  
c/lab-7: No such file or directory  
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ cd ~/w  
ork/study/2024-2025/"Архитектура компьютера"/arch-pc/lab07  
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$  
touch lab7-1.asm
```

Рис. 2.1: Создан каталог

Инструкция `jmp` в NASM используется для реализации безусловных переходов. Рассмотрим пример программы с использованием инструкции `jmp`. Написал в файл lab7-1.asm текст программы из листинга 7.1. (рис. 2.2)

```
~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07/lab7-1.asm - Mousepad
File Edit Search View Document Help
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
msg1: DB 'Сообщение № 1',0
msg2: DB 'Сообщение № 2',0
msg3: DB 'Сообщение № 3',0

SECTION .text
GLOBAL _start
_start:

jmp _label2

_label1:
mov eax, msg1 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 1'

_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 2'
78 Демидова А. В.
Архитектура ЭВМ

_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 3'

_end:
call quit ; вызов подпрограммы завершения
```

Рис. 2.2: Программа lab7-1.asm

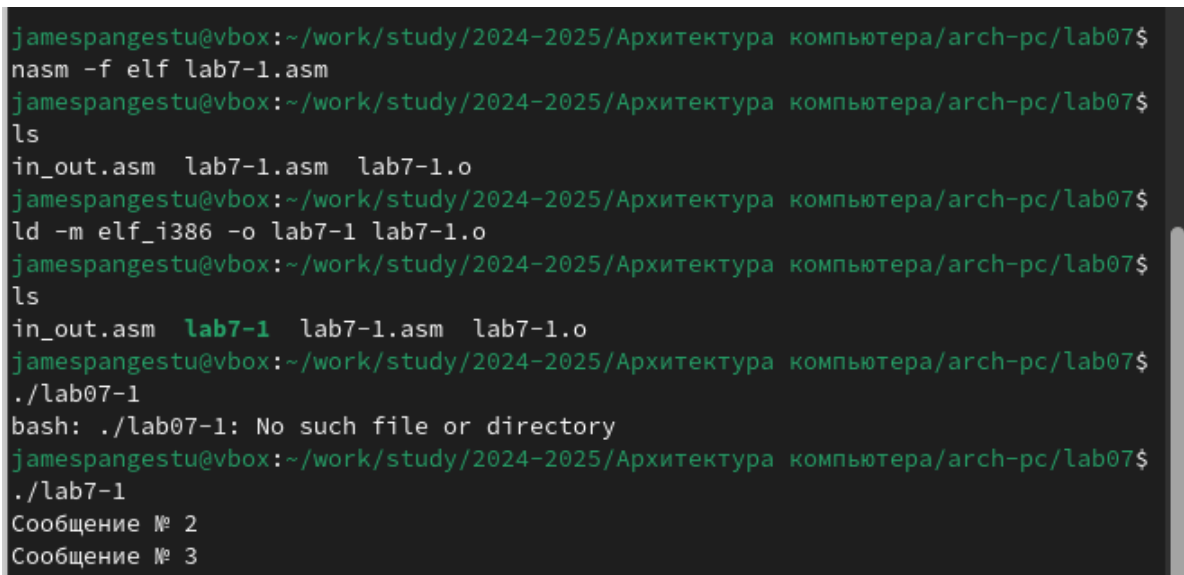
Создаю исполняемый файл и запускаю его. (рис. 2.3)

```
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
nasm -f elf lab7-1.asm
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
ls
in_out.asm lab7-1.asm lab7-1.o
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
ld -m elf_i386 -o lab7-1 lab7-1.o
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
ls
in_out.asm lab7-1 lab7-1.asm lab7-1.o
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
./lab07-1
bash: ./lab07-1: No such file or directory
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
./lab7-1
Сообщение № 2
Сообщение № 3
```

Рис. 2.3: Запуск программы lab7-1.asm

Инструкция `jmp` позволяет осуществлять переходы не только вперед но и назад. Изменим программу таким образом, чтобы она выводила сначала ‘Сообщение № 2’, потом ‘Сообщение № 1’ и завершала работу. Для этого в текст программы после вывода сообщения № 2 добавим инструкцию `jmp` с меткой `_label1` (т.е. переход к инструкциям вывода сообщения № 1) и после вывода сообщения № 1 добавим инструкцию `jmp` с меткой `_end` (т.е. переход к инструкции `call quit`).

Изменяю текст программы в соответствии с листингом 7.2. (рис. 2.4) (рис. 2.5)



```
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
nasm -f elf lab7-1.asm
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
ls
in_out.asm  lab7-1.asm  lab7-1.o
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
ld -m elf_i386 -o lab7-1 lab7-1.o
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
ls
in_out.asm  lab7-1  lab7-1.asm  lab7-1.o
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
./lab07-1
bash: ./lab07-1: No such file or directory
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
./lab7-1
Сообщение № 2
Сообщение № 3
```

Рис. 2.4: Программа lab7-1.asm

```
%include 'in_out.asm'
SECTION .data
msg1: DB 'Сообщение № 1',0
msg2: DB 'Сообщение № 2',0
msg3: DB 'Сообщение № 3',0
SECTION .text
GLOBAL _start

_start:
jmp _label3

_label1:
mov eax, msg1
call sprintLF
jmp _end

_label2:
mov eax, msg2
call sprintLF
jmp _label1

_label3:
mov eax, msg3
call sprintLF
jmp _label2

_end:
call quit
```

Рис. 2.5: Запуск программы lab7-1.asm

Изменила текст программы, чтобы вывод программы был следующим (рис. 2.6) (рис. 2.7):

Сообщение № 3

Сообщение № 2

Сообщение № 1

```

%include 'in_out.asm'
SECTION .data
msg1: DB 'Сообщение № 1',0
msg2: DB 'Сообщение № 2',0
msg3: DB 'Сообщение № 3',0
SECTION .text
GLOBAL _start

_start:
jmp _label3

_label1:
mov eax, msg1
call sprintLF
jmp _end

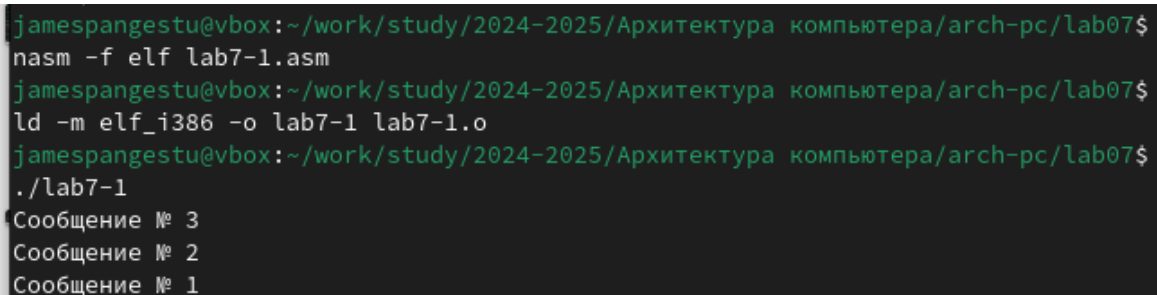
_label2:
mov eax, msg2
call sprintLF
jmp _label1

_label3:
mov eax, msg3
call sprintLF
jmp _label2

_end:
call quit

```

Рис. 2.6: Программа lab7-1.asm



```

jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
nasm -f elf lab7-1.asm
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
ld -m elf_i386 -o lab7-1 lab7-1.o
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1

```

Рис. 2.7: Запуск программы lab7-1.asm

Использование инструкции `jmp` приводит к переходу в любом случае. Однако, часто при написании программ необходимо использовать условные переходы, т.е. переход должен происходить если выполнено какое-либо условие. В качестве примера рассмотрим программу, которая определяет и выводит на экран наибольшую из 3 целочисленных переменных: A, B и C. Значения для A и C задаются в программе, значение B вводится с клавиатуры.

Создала исполняемый файл и проверила его работу для разных значений B (рис. 2.8) (рис. 2.9).

```
26 mov [max],ecx ; 'max = A'
27 ; ----- Сравниваем 'A' и 'C' (как символы)
28 cmp ecx,[C] ; Сравниваем 'A' и 'C'
29 jg check_B ; если 'A>C', то переход на метку 'check_B',
30 mov ecx,[C] ; иначе 'ecx = C'
31 mov [max],ecx ; 'max = C'
32 ; ----- Преобразование 'max(A,C)' из символа в число
33 check_B:
34 mov eax,max
35 call atoi ; Вызов подпрограммы перевода символа в число
36 mov [max],eax ; запись преобразованного числа в 'max'
37 ; ----- Сравниваем 'max(A,C)' и 'B' (как числа)
38 mov ecx,[max]
39 cmp ecx,[B] ; Сравниваем 'max(A,C)' и 'B'
40 jg fin ; если 'max(A,C)>B', то переход на 'fin',
41 mov ecx,[B] ; иначе 'ecx = B'
42 mov [max],ecx
43 ; ----- Вывод результата
44 fin:
45 mov eax, msg2
46 call sprint ; Вывод сообщения 'Наибольшее число: '
47 mov eax,[max]
48 call iprintLF ; Вывод 'max(A,B,C)'
49 call quit ; Выход
50
```

Рис. 2.8: Программа lab7-2.asm

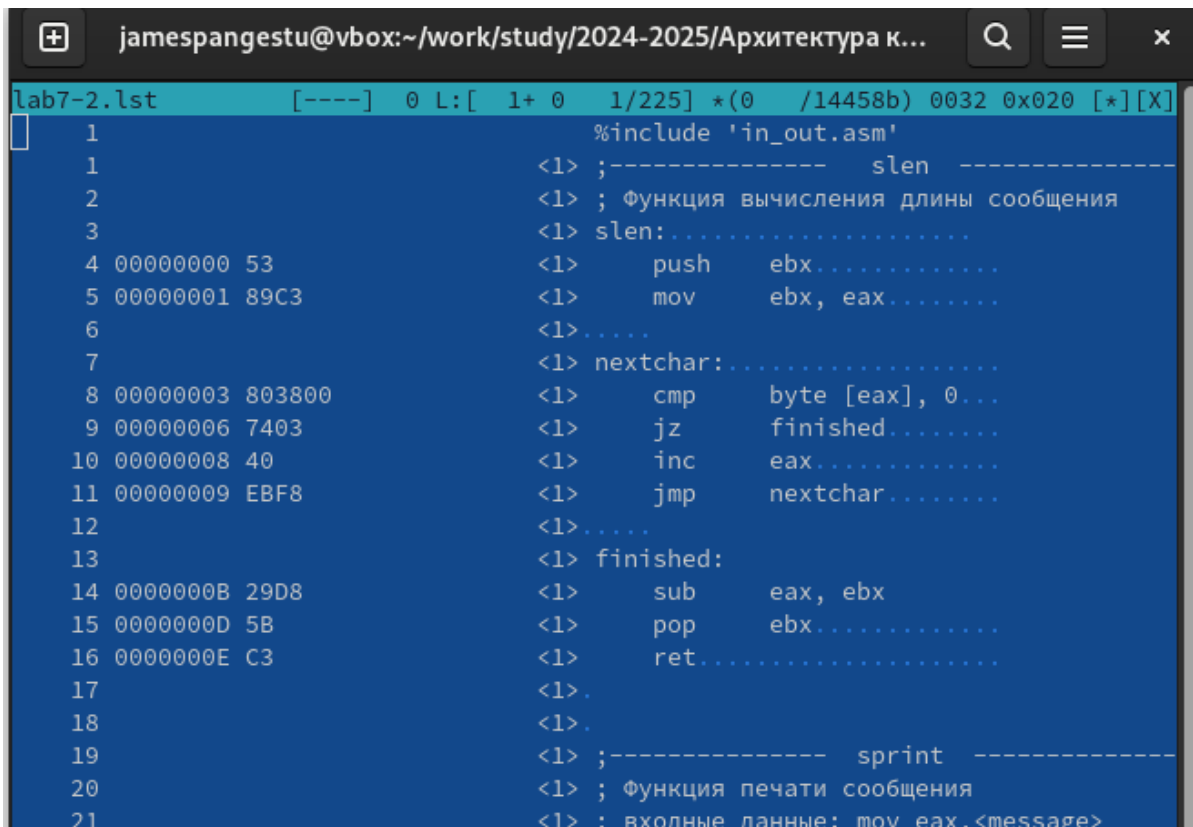
```
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
nasm -f elf lab7-2.asm
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
ld -m elf_i386 -o lab7-2 lab7-2.o
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
./lab07-2
bash: ./lab07-2: No such file or directory
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
./lab7-2
Введите B: 0
Наибольшее число: 50
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
./lab7-2
Введите B: 2
Наибольшее число: 50
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
./lab7-2
Введите B: 55
Наибольшее число: 55
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
./lab7-2
Введите B: 41
Наибольшее число: 50
```

Рис. 2.9: Запуск программы lab7-2.asm

2.2 Изучение структуры файлы листинга

Обычно nasm создаёт в результате ассемблирования только объектный файл. Получить файл листинга можно, указав ключ -l и задав имя файла листинга в командной строке.

Создаю файл листинга для программы из файла lab7-2.asm (рис. 2.10)



```
lab7-2.lst [----] 0 L:[ 1+ 0 1/225] *(0 /14458b) 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> .
19                                     <1> ;----- sprint -----
20                                     <1> ; Функция печати сообщения
21                                     <1> ; входные данные: mov eax,<message>
```

Рис. 2.10: Файл листинга lab7-2

Ознакомимся с его форматом и содержимым.

строка 211

- 34 - номер строки
- 0000012E - адрес
- B8[00000000] - машинный код
- mov eax,ebx - код программы

строка 212

- 35 - номер строки
- 00000133 - адрес

- E864FFFFFF - машинный код
- call atoi - код программы

строка 213

- 36 - номер строки
- 00000138 - адрес
- A3[00000000] - машинный код
- mov [max],eax - код программы

Открыла файл с программой lab7-2.asm и в инструкции с двумя операндами удалила один операнд. Выполню трансляцию с получением файла листинга. (рис. 2.11) (рис. 2.12)

```
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
nasm -f elf -l lab7-2.lst lab7-2.asm
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
mcedit lab7-2.lst
```

Рис. 2.11: Ошибка трансляции lab7-2

```

jamespangestu@vbox:~/work/study/2024-2025/Архитектура к...
lab7-2.lst [----] 0 L:[ 1+ 0 1/225] *(0 /14458b) 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>.
19                                     <1> ;----- sprint -----
20                                     <1> ; Функция печати сообщения
21                                     <1> ; входные данные: mov eax,<message>

```

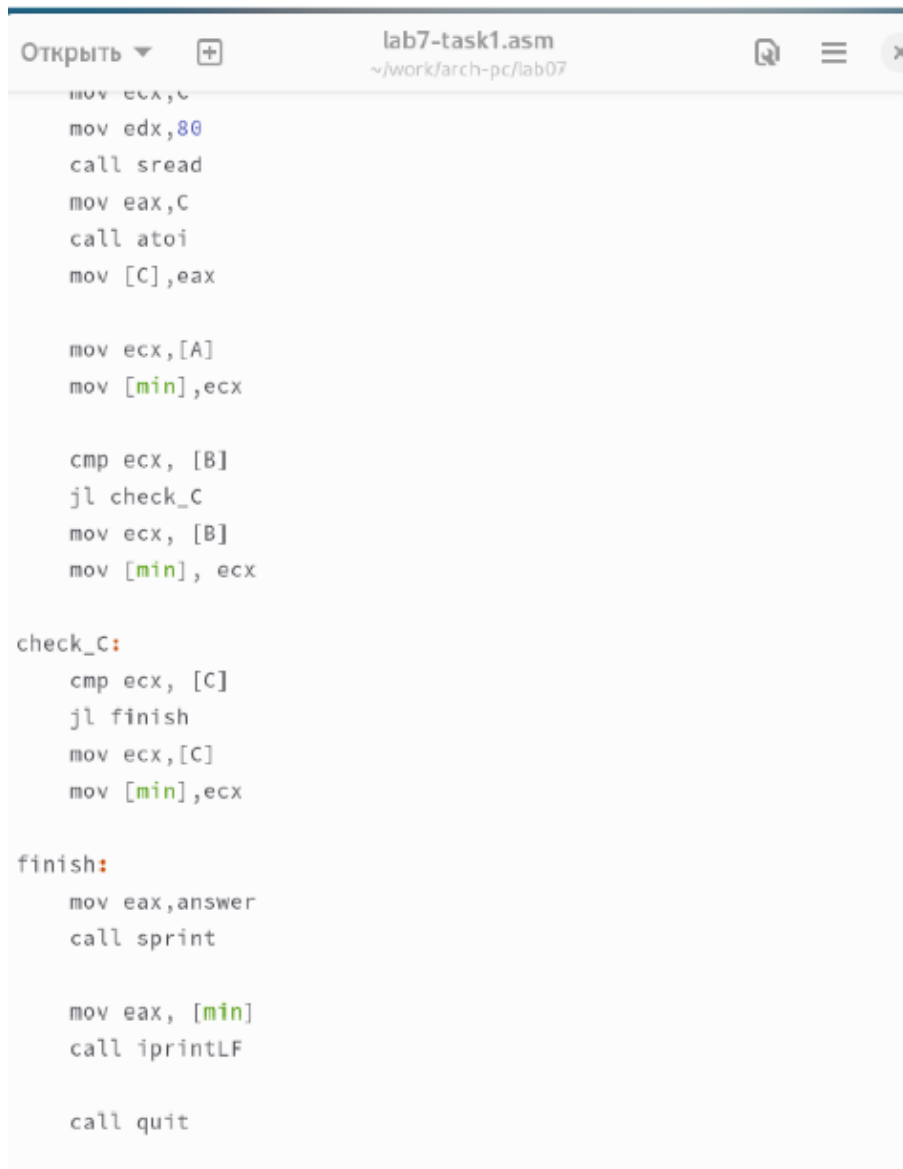
Рис. 2.12: Файл листинга с ошибкой lab7-2

Объектный файл не смог создаться из-за ошибки. Но получился листинг, где выделено место ошибки.

2.3 Самостоятельное задание

Напишите программу нахождения наименьшей из 3 целочисленных переменных a, b и c. Значения переменных выбрать из табл. 7.5 в соответствии с вариантом, полученным при выполнении лабораторной работы № 6. Создайте исполняемый файл и проверьте его работу (рис. 2.13) (рис. 2.14)

для варианта 13 - 84,32,77



```
Открыть ▾ + lab7-task1.asm
~/.work/arch-pc/lab07

mov ecx,C
mov edx,80
call sread
mov eax,C
call atoi
mov [C],eax

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

cmp ecx, [B]
jl check_C
mov ecx, [B]
mov [min], ecx

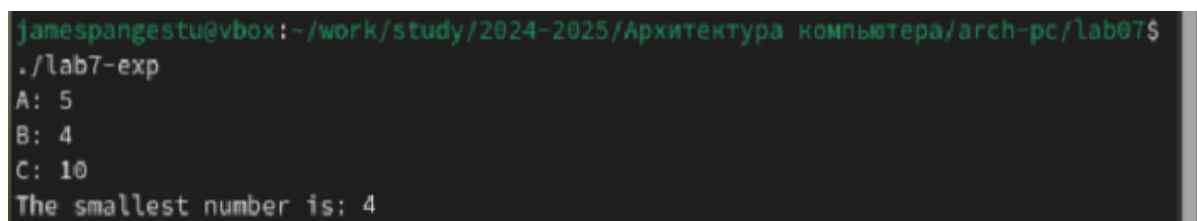
check_C:
cmp ecx, [C]
jl finish
mov ecx,[C]
mov [min],ecx

finish:
mov eax,answer
call sprint

mov eax, [min]
call iprintLF

call quit
```

Рис. 2.13: Программа lab7-task1.asm



```
jamespangestu@vbox: ~/.work/study/2024-2025/Архитектура компьютера/arch-pc/lab07$
./lab7-exp
A: 5
B: 4
C: 10
The smallest number is: 4
```

Рис. 2.14: Запуск программы lab7-task1.asm

Напишите программу, которая для введенных с клавиатуры значений x и a

вычисляет значение заданной функции $f(x)$ и выводит результат вычислений. Вид функции $f(x)$ выбрать из таблицы 7.6 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу для значений X и a из 7.6. (рис. 2.15) (рис. 2.16)

для варианта 13

$$\begin{cases} a - 7, a \geq 7 \\ ax, a < 7 \end{cases}$$

При $x = 3, a = 9$ получается 2.

При $x = 6, a = 4$ получается 24.

```
mov [A], eax

mov eax, msgX
call sprint
mov ecx, X
mov edx, 80
call sread
mov eax, X
call atoi
mov [X], eax

mov ebx, [A]
mov edx, 7
cmp ebx, edx
jge first
jmp second

first:
    mov eax, [A]
    sub eax, 7
    call iprintLF
    call quit
second:
    mov eax, [X]
    mov ebx, [A]
    mul ebx
    call iprintLF
    call quit
```

Рис. 2.15: Программа lab7-task2.asm

```
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-p
c/lab07$ nasm -f elf32 lab7-final.asm -o lab7-final.o
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-p
c/lab07$ ld -m elf_i386 lab7-final.o -o lab7-final
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-p
c/lab07$ ./lab7-final
Enter value for x:5
Enter value for a:6
The value of f(x) is: 6
7
jamespangestu@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-p
c/lab07$ ./lab7-final
Enter value for x:10
Enter value for a:7
The value of f(x) is: 7
```

Рис. 2.16: Запуск программы lab7-task2.asm

3 Выводы

Изучили команды условного и безусловного переходов, познакомились с фалом листинга.