

Шаблон отчёта по лабораторной работе

архитектура компьютера

мохамед Муса

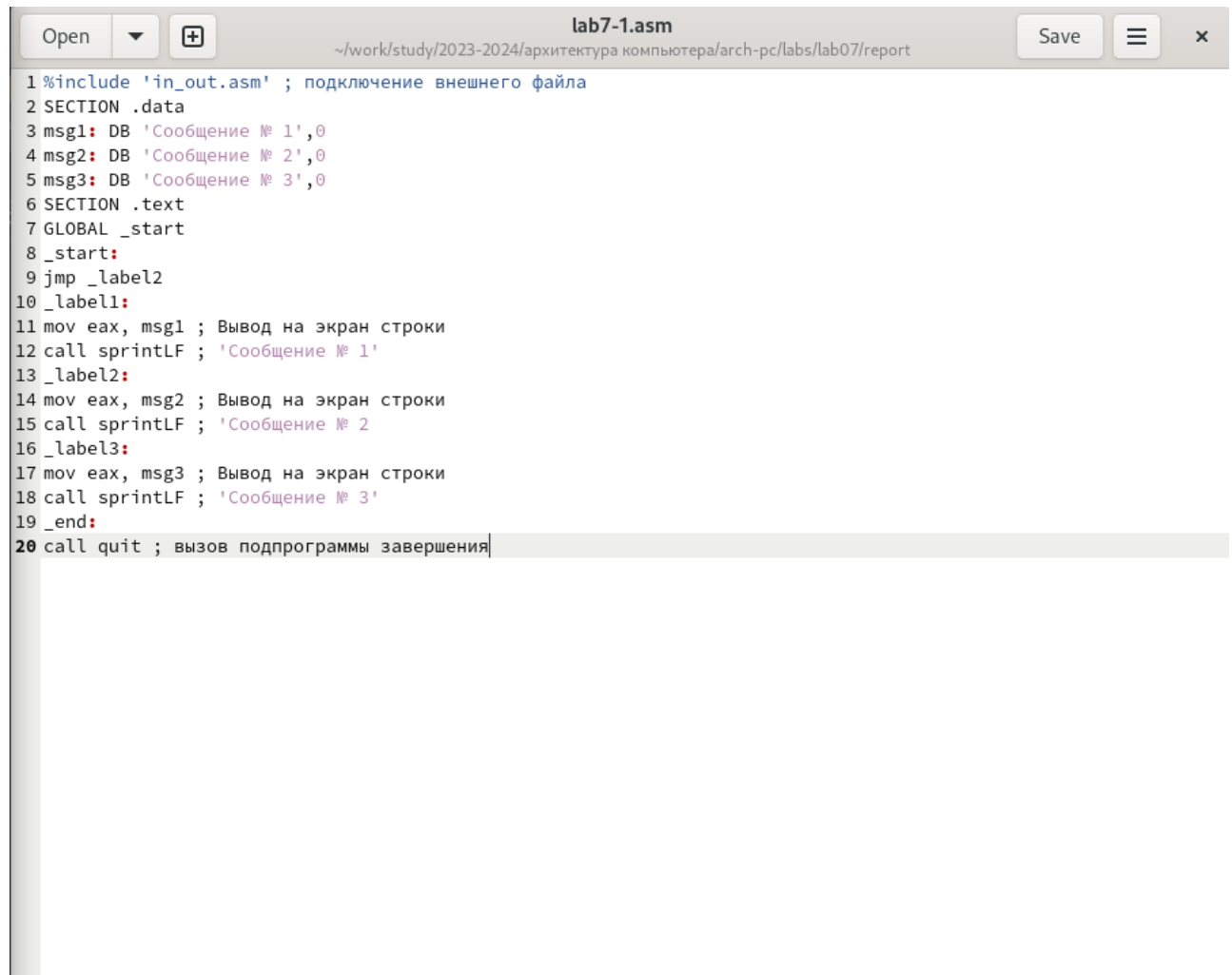
Цель работы

Цель этой работы - углубиться в изучение языка ассемблера, и мы будем больше практиковаться в том, как создавать программы, способные выполнять уравнения

выполнения лабораторной работы

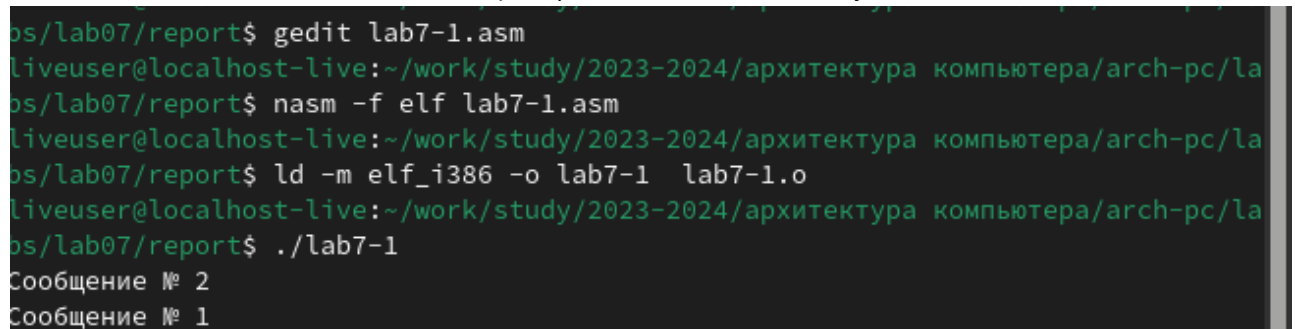
- Сначала я создал файл lab7-1.asm, скопировал код из pdf и запустил его :

```
bs/lab07/report$ touch lab7-1.asm
liveuser@localhost-live:~/work/study/2023-2024/архитектура компьютера/arch-pc/la
bs/lab07/report$ gedit lab7-1.asm
liveuser@localhost-live:~/work/study/2023-2024/архитектура компьютера/arch-pc/la
bs/lab07/report$ nasm -f elf lab7-1.asm
liveuser@localhost-live:~/work/study/2023-2024/архитектура компьютера/arch-pc/la
bs/lab07/report$ ld -m elf_i386 -o lab7-1 lab7-1.o
liveuser@localhost-live:~/work/study/2023-2024/архитектура компьютера/arch-pc/la
bs/lab07/report$ ./lab7-1
Сообщение № 2
Сообщение № 3
```



```
1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data
3 msg1: DB 'Сообщение № 1',0
4 msg2: DB 'Сообщение № 2',0
5 msg3: DB 'Сообщение № 3',0
6 SECTION .text
7 GLOBAL _start
8 _start:
9 jmp _label2
10 _label1:
11 mov eax, msg1 ; Вывод на экран строки
12 call sprintf ; 'Сообщение № 1'
13 _label2:
14 mov eax, msg2 ; Вывод на экран строки
15 call sprintf ; 'Сообщение № 2'
16 _label3:
17 mov eax, msg3 ; Вывод на экран строки
18 call sprintf ; 'Сообщение № 3'
19 _end:
20 call quit ; вызов подпрограммы завершения
```

- И я внес необходимые изменения из pdf-файл в lab7-1.asm и запустил его снова :



```
bs/lab07/report$ gedit lab7-1.asm
liveuser@localhost-live:~/work/study/2023-2024/архитектура компьютера/arch-pc/la
bs/lab07/report$ nasm -f elf lab7-1.asm
liveuser@localhost-live:~/work/study/2023-2024/архитектура компьютера/arch-pc/la
bs/lab07/report$ ld -m elf_i386 -o lab7-1 lab7-1.o
liveuser@localhost-live:~/work/study/2023-2024/архитектура компьютера/arch-pc/la
bs/lab07/report$ ./lab7-1
Сообщение № 2
Сообщение № 1
```

```

1 %include 'in_out.asm' ; подключение внешнего файла
2 section .data
3 msg1: DB 'Сообщение № 1',0
4 msg2: DB 'Сообщение № 2',0
5 msg3: DB 'Сообщение № 3',0
6 SECTION .text
7 GLOBAL _start
8 _start:
9 jmp _label2
10 _label1:
11 mov eax, msg1 ; Вывод на экран строки
12 call sprintf ; 'Сообщение № 1'
13 jmp _end
14 _label2:
15 mov eax, msg2 ; Вывод на экран строки
16 call sprintf ; 'Сообщение № 2'
17 jmp _label1
18 _label3:
19 mov eax, msg3 ; Вывод на экран строки
20 call sprintf ; 'Сообщение № 3'
21 _end:
22 call quit ; вызов подпрограммы завершения

```

- Я также запустил файл lab7-2 и отредактировал его в соответствии с инструкцией в формате pdf :

```

1 %include 'in_out.asm'
2 section .data
3 msg1 db 'Введите B: ',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 B: '
14 mov eax,msg1
15 call sprint
16 ; ----- Ввод 'B'
17 mov ecx,B
18 mov edx,10
19 call sread
20 ; ----- Преобразование 'B' 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 'B'
24 ; ----- Записываем 'A' в переменную 'max'
25 mov ecx,[A] ; 'ecx = A'
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)

```

{#fig:001 width=70%}

```
bs/lab07/report$ touch lab7-2.asm
liveuser@localhost-live:~/work/study/2023-2024/архитектура компьютера/arch-pc/la
bs/lab07/report$ gedit lab7-2.asm
liveuser@localhost-live:~/work/study/2023-2024/архитектура компьютера/arch-pc/la
bs/lab07/report$ gedit lab7-2.asm
liveuser@localhost-live:~/work/study/2023-2024/архитектура компьютера/arch-pc/la
bs/lab07/report$ nasm -f elf lab7-2.asm
liveuser@localhost-live:~/work/study/2023-2024/архитектура компьютера/arch-pc/la
bs/lab07/report$ ld -m elf_i386 -o lab7-2 lab7-2.o
liveuser@localhost-live:~/work/study/2023-2024/архитектура компьютера/arch-pc/la
bs/lab07/report$ ./lab7-2
Введите B: 4
Наибольшее число: 50
```

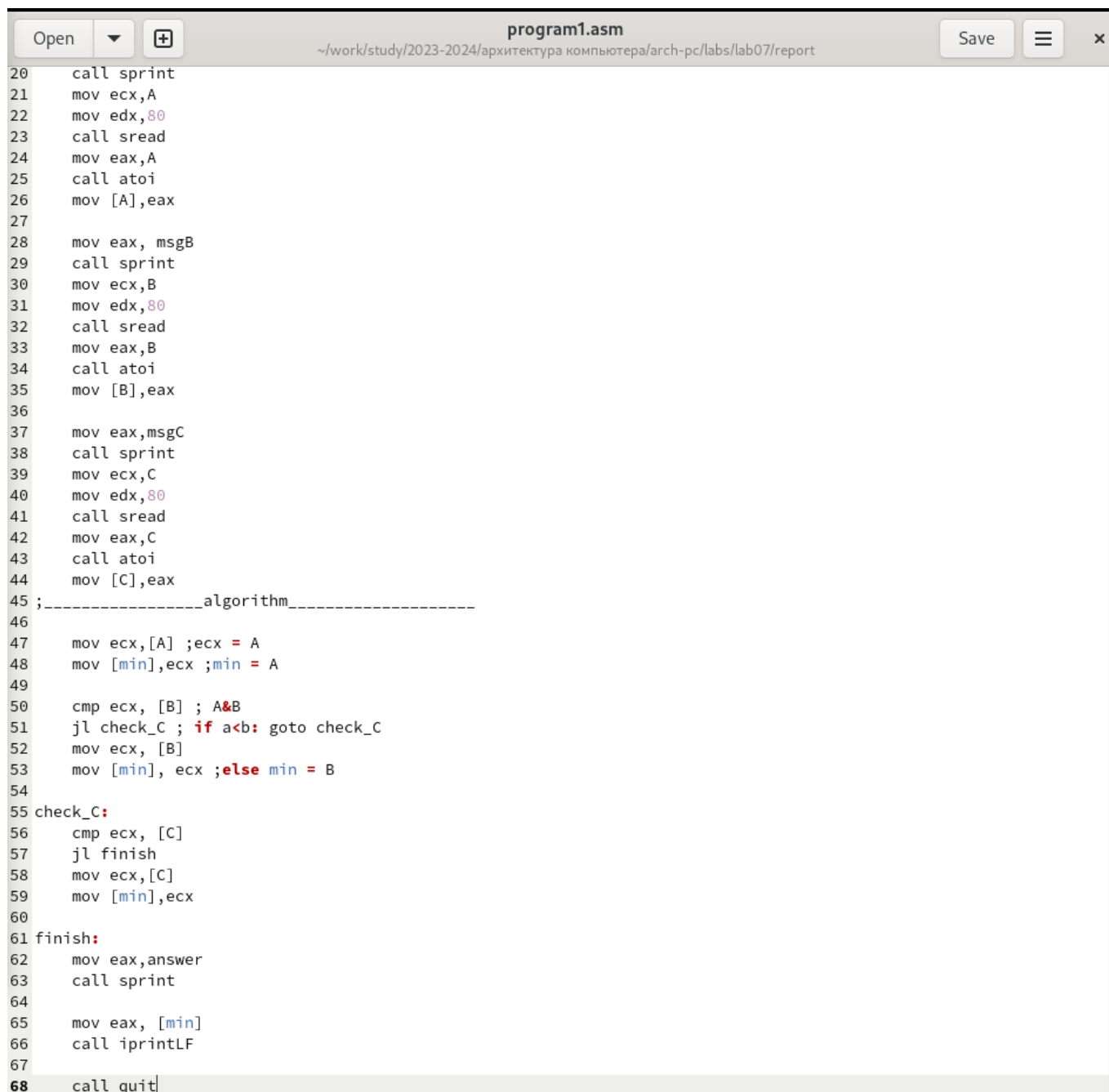
- В-третьих, я создал файл lab7-2.lst с помощью команды nasm :

```
lab7-2.lst      [-M--]  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>
1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn10Quit
```

Выполнения заданий для самостоятельной работы:

- сначала я написал первую программу и запустил ее :

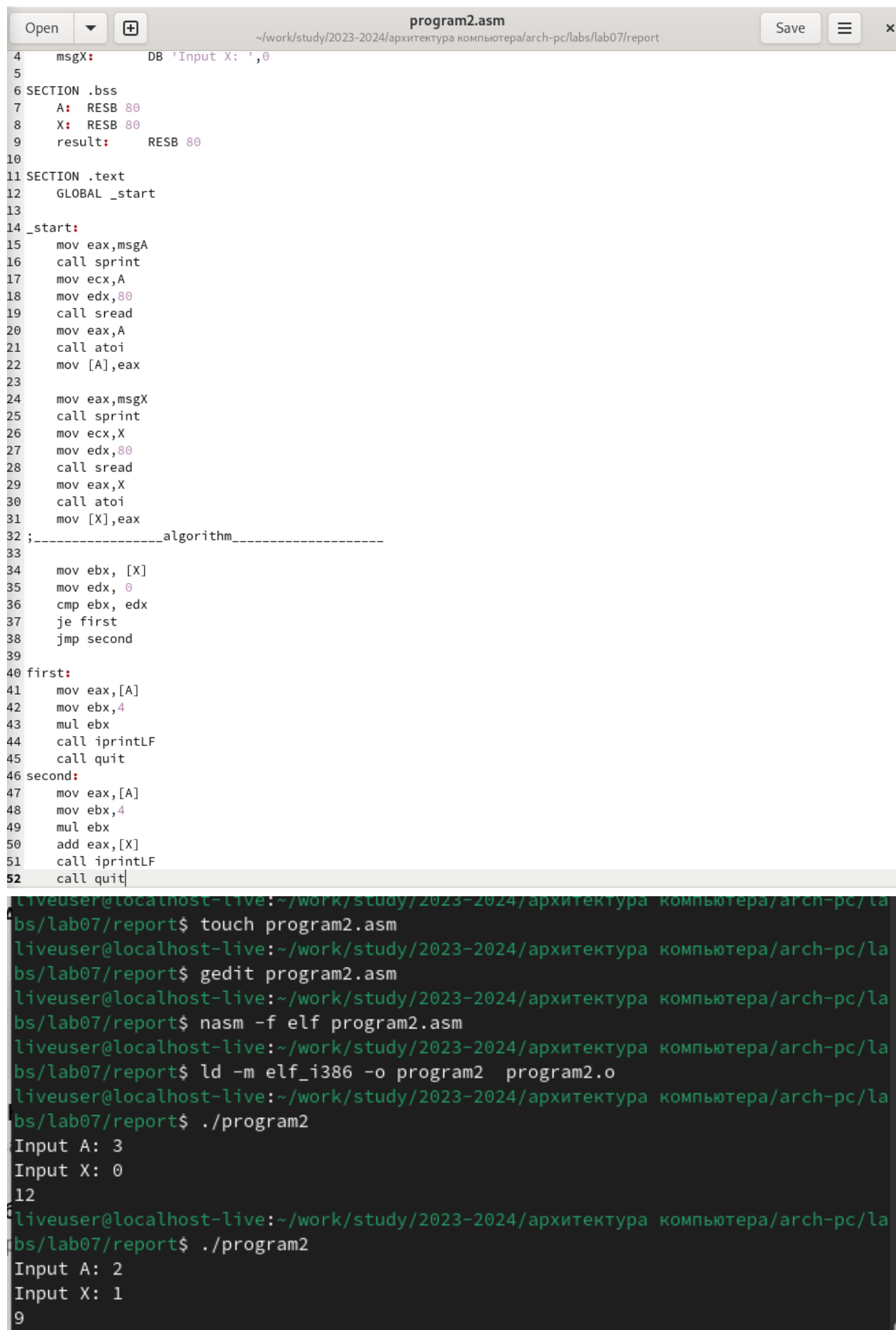
```
bs/lab07/report$ gedit program1.asm
liveuser@localhost-live:~/work/study/2023-2024/архитектура компьютера/arch-pc/lab07/report$ nasm -f elf program1.asm
liveuser@localhost-live:~/work/study/2023-2024/архитектура компьютера/arch-pc/lab07/report$ ld -m elf_i386 -o program1 program1.o
liveuser@localhost-live:~/work/study/2023-2024/архитектура компьютера/arch-pc/lab07/report$ ./program1
Input A: 4
Input B: 3
Input C: 5
Smallest: 3
```



```
program1.asm
~/work/study/2023-2024/архитектура компьютера/arch-pc/labs/lab07/report

20  call sprint
21  mov ecx,A
22  mov edx,80
23  call sread
24  mov eax,A
25  call atoi
26  mov [A],eax
27
28  mov eax, msgB
29  call sprint
30  mov ecx,B
31  mov edx,80
32  call sread
33  mov eax,B
34  call atoi
35  mov [B],eax
36
37  mov eax,msgC
38  call sprint
39  mov ecx,C
40  mov edx,80
41  call sread
42  mov eax,C
43  call atoi
44  mov [C],eax
45 ;-----algorithm-----
46
47  mov ecx,[A] ;ecx = A
48  mov [min],ecx ;min = A
49
50  cmp ecx, [B] ; A<B
51  jl check_C ; if a<b: goto check_C
52  mov ecx, [B]
53  mov [min], ecx ;else min = B
54
55 check_C:
56  cmp ecx, [C]
57  jl finish
58  mov ecx,[C]
59  mov [min],ecx
60
61 finish:
62  mov eax,answer
63  call sprint
64
65  mov eax, [min]
66  call iprintLF
67
68  call quit
```

- и я написал вторую программу в соответствии с инструкциями, приведенными в pdf-файле и запустил ее:



```
program2.asm
~/work/study/2023-2024/архитектура компьютера/arch-pc/labs/lab07/report

4 msgX: DB 'Input X: ',0
5
6 SECTION .bss
7 A: RESB 80
8 X: RESB 80
9 result: RESB 80
10
11 SECTION .text
12 GLOBAL _start
13
14 _start:
15 mov eax,msgA
16 call sprint
17 mov ecx,A
18 mov edx,80
19 call sread
20 mov eax,A
21 call atoi
22 mov [A],eax
23
24 mov eax,msgX
25 call sprint
26 mov ecx,X
27 mov edx,80
28 call sread
29 mov eax,X
30 call atoi
31 mov [X],eax
32 ;-----algorithm-----
33
34 mov ebx, [X]
35 mov edx, 0
36 cmp ebx, edx
37 je first
38 jmp second
39
40 first:
41 mov eax,[A]
42 mov ebx,4
43 mul ebx
44 call iprintLF
45 call quit
46 second:
47 mov eax,[A]
48 mov ebx,4
49 mul ebx
50 add eax,[X]
51 call iprintLF
52 call quit
```

```
liveuser@localhost-live:~/work/study/2023-2024/архитектура компьютера/arch-pc/labs/lab07/report$ touch program2.asm
liveuser@localhost-live:~/work/study/2023-2024/архитектура компьютера/arch-pc/labs/lab07/report$ gedit program2.asm
liveuser@localhost-live:~/work/study/2023-2024/архитектура компьютера/arch-pc/labs/lab07/report$ nasm -f elf program2.asm
liveuser@localhost-live:~/work/study/2023-2024/архитектура компьютера/arch-pc/labs/lab07/report$ ld -m elf_i386 -o program2 program2.o
liveuser@localhost-live:~/work/study/2023-2024/архитектура компьютера/arch-pc/labs/lab07/report$ ./program2
Input A: 3
Input X: 0
12
liveuser@localhost-live:~/work/study/2023-2024/архитектура компьютера/arch-pc/labs/lab07/report$ ./program2
Input A: 2
Input X: 1
9
```

Выводы

В этом pdf-файле мы подробнее попрактиковались в написании кода на ассемблере и узнали о типе файла .lst.