# Шаблон отчёта по лабораторной работе №7

## Дисциплина: архитектура компьютера

Нурыева Гулсолтан

## Содержание

*Рис. 1: Созания каталога и файла*

Потом зашла на MC и через него скопировала файл in_out.asm в созданный каталог

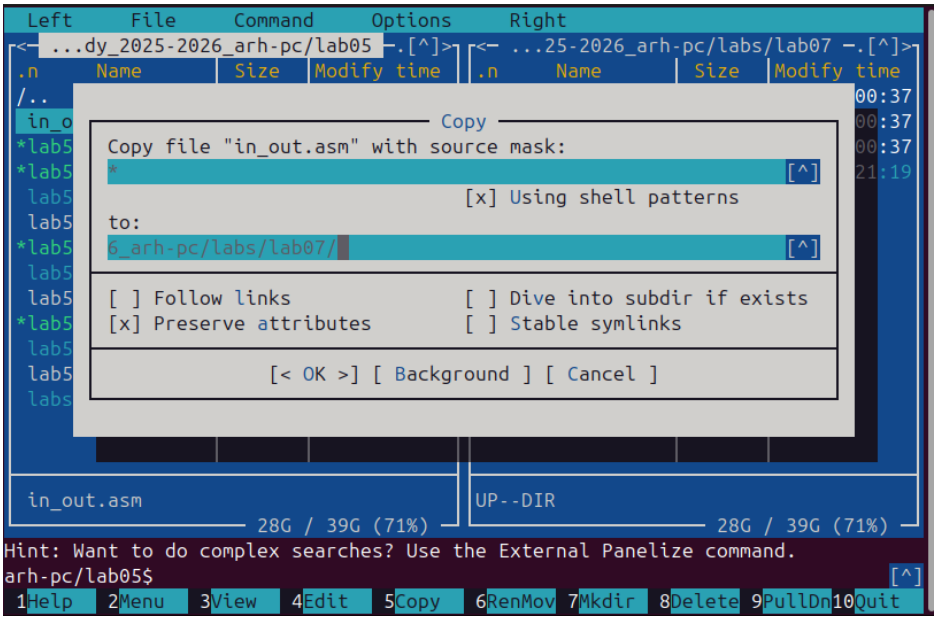



*Рис. 2: Скопирования файла in_out.asm в нужный каталог*

После этого я открыла созданной мною файл с помощью клавиши F4 и ввел туда программу с использованием инструкции jmp

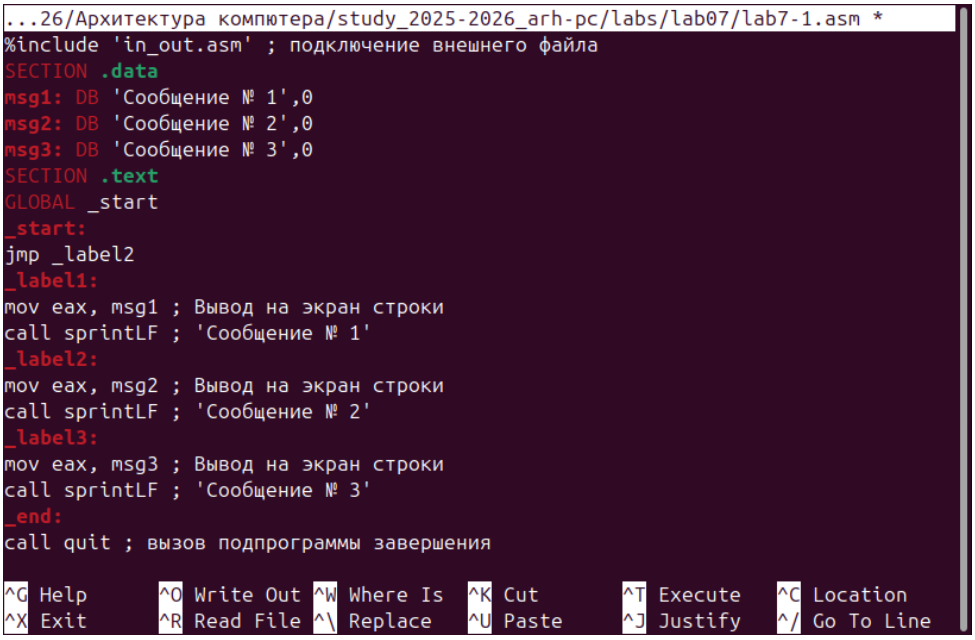



*Рис. 3: Программа с использованием инструкции jmp*

Потом я создала исполняемый файл и запустила его

```
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компютера/study_2025-2026_
arh-pc/labs/lab07$ nasm -f elf lab7-1.asm
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компютера/study_2025-2026_
arh-pc/labs/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компютера/study_2025-2026_
arh-pc/labs/lab07$ .lab7-1
.lab7-1: command not found
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компютера/study_2025-2026_
arh-pc/labs/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 3
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компютера/study_2025-2026_
arh-pc/labs/lab07$
```
ьс

*Рис. 4: Создания испольняемого файла*

Я изменила текст файла чтобы осуществить переход назад в инструкции jmp. Для этого в текст программы после вывода сообщения № 2 добавила инструкцию jmp с меткой _label1, и после вывода сообщения № 1 добавила инструкцию jmp с меткой _end

```
mc [vboxuser@gulsoltan]:~/work/study/2025-2026/Архитектура компют...
...26/Архитектура компютера/study_2025-2026_arh-pc/labs/lab07/lab7-1.asm *
%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 ;
_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 2'
jmp _label1
_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 3'
_end:

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

*Рис. 5: Изменения текста файла*

Создала испольнямый файл и запустила его ещё раз но уже изменённого.

```
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компютера/study_2025-2026_
arh-pc/labs/lab07$ nasm -f elf lab7-1.asm
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компютера/study_2025-2026_
arh-pc/labs/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компютера/study_2025-2026_
arh-pc/labs/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 1
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компютера/study_2025-2026_
arh-pc/labs/lab07$
```

*Рис. 6: Создания(изменённого) испольняемого файла*

Потом я создала новый файл в том же каталоге lab7-2.asm

```
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компютера/study_2025-2026_
arh-pc/labs/lab07$ touch lab7-2.asm
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компютера/study_2025-2026_
arh-pc/labs/lab07$
```

*Рис. 7: Создания файла lab7-2.asm*

После создания я окрыла файл и ввела туда программу, которая определяет и выводит на экран наибольшую из 3 целочисленных переменных: A,B и C

```
vboxuser@gulsoltan: ~/work/study/2025-2026/Ap...   vboxuser@gulsoltan: ~/work/study/2025-2026/Ap...
...5-2026/Архитектура компютера/study_2025-2026_arh-pc/labs/lab07/lab7-2.asm *
%include 'in_out.asm'
section .data
msg1 db 'Введите B: ',0h
msg2 db "Наибольшее число: ",0h
A dd '20'
C dd '50'
section .bss
max resb 10
B resb 10
section .text
global _start
_start:
; ---------- Вывод сообщения 'Введите B: '
mov eax,msg1
call sprint
; ---------- Ввод 'B'
mov ecx,B
mov edx,10
call sread
; ---------- Преобразование 'B' из символа в число
mov eax,B
call atoi ; Вызов подпрограммы перевода символа в число
mov [B],eax ; запись преобразованного числа в 'B'
; ---------- Записываем 'A' в переменную 'max'
mov ecx,[A] ; 'ecx = A'
mov [max],ecx ; 'max = A'

^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```

*Рис. 8: Программа,которая определяет и выводит на экран наибольшую из 3 целочисленных переменных: A,B и C*

Потом создала исполняемый файл и запустил его. И ещё я проверила его работу

```
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компютера/study_2025-2026_arh-
pc/labs/lab07$ nasm -f elf lab7-2.asm
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компютера/study_2025-2026_arh-
pc/labs/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компютера/study_2025-2026_arh-
pc/labs/lab07$ ./lab7-2
Введите B: 20
Наибольшее число: 50
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компютера/study_2025-2026_arh-
pc/labs/lab07$ ./lab7-2
Введите B: 70
Наибольшее число: 70
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компютера/study_2025-2026_arh-
pc/labs/lab07$
```

*Рис. 9: Создания испольняемого файла lab7-2.asm*

## 4.2 Изучение структуры файлы листинга

Я создала файл листинга с помощью nasm указав ключ -l и задала имя лисинга в командной строке

```
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компютера/study_2025-2026_arh-pc/labs/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компютера/study_2025-2026_arh-pc/labs/lab07$
```

*Рис. 10: Создания листинга*

Потом октрыла файл листинга с помощью mcedit и изучила содержимое

```
vboxuser@gulsoltan: ~/work/study/2025-2026/Ap...   vboxuser@gulsoltan: ~/work/study/2025-2026/Ap...
/home/vbox~ab7-2.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>
   22                              <1> sprint:
   23 0000000F 52                  <1>     push    edx
   24 00000010 51                  <1>     push    ecx
   25 00000011 53                  <1>     push    ebx
   26 00000012 50                  <1>     push    eax
   27 00000013 E8E8FFFFFF          <1>     call    slen
1Help  2Save  3Mark  4Replac  5Copy  6Move  7Search  8Delete  9PullDn  10Quit
```

*Рис. 11: Открытие листинга*

Выбрала первую строку и это 112. В строке которая показана в картинке снизу обозначается “00000086” — адрес в памяти, “E8C9FFFFFF” — машинный код для инструкции call а “call inprint” — обозначает вызов функции inprint.

```
112 00000086 E8C9FFFFFF          <1>     call    iprint.....
```

*Рис. 12: 112 строка для объяснения*

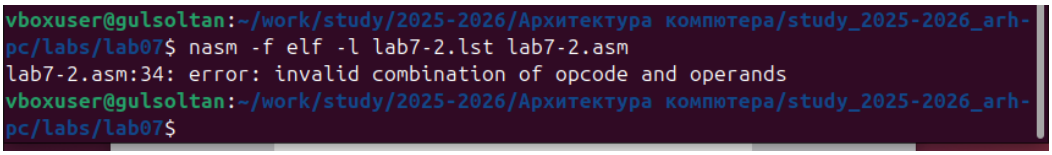
Выбрала вторую строку и это 14. В строке которая показана в картинке снизу обозначается “0000000B” — адрес в памяти, где расположена эта инструкция, 29D8 — машинный код для инструкции sub a “sub eax, ebx” — обозначает операцию, которая вычитает значение регистра ebx из значения регистра eax и сохраняет результат в eax.


*Рис. 13: 14 строка для объяснения*

Выбрала третью строку и это 42. В строке которая показана в картинке снизу обозначается “00000153” — адрес в памяти, где расположена эта инструкция, 890D — машинный код для инструкции mov a “mov [max], ecx”— Обозначает операцию, которая копирует значение из регистра ecx в память по адресу, соответствующему метке или переменной max.


*Рис. 14: 42 строка для объяснения*

Потом в строке mov eax,max я убрала max и попробовала создать файл. Выдало ошибку, так как для программы нужно два операнда.




ьс

*Рис. 15: Ошибка в программе*

В файле листинга показывает где ошибка и с чем оно связана

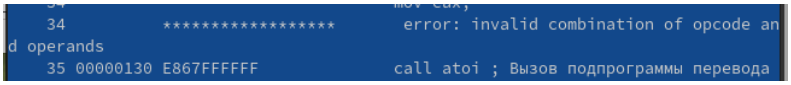



*Рис. 16: Осмотр листинга*

## 4.3 Самостоятельная работа.

### 4.3.1 Напишите программу нахождения наименьшей из 3 целочисленных переменных a,b и c.

Для начала я создала файл и в него я написала программу нахождения наименьшей из 3 целочисленных переменных a,b и c.

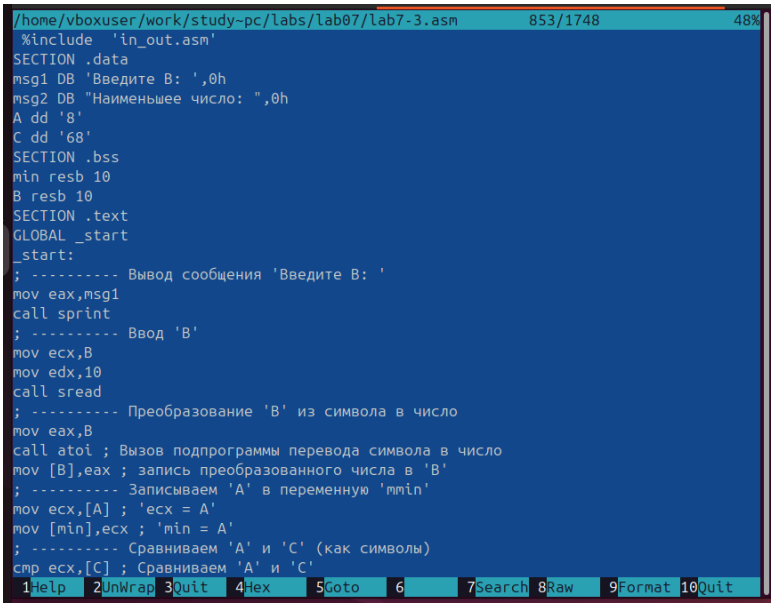



*Рис. 17: Внесения программы в файл*

Потом создала исполняемый файл и запустила его и проверил все ли работает

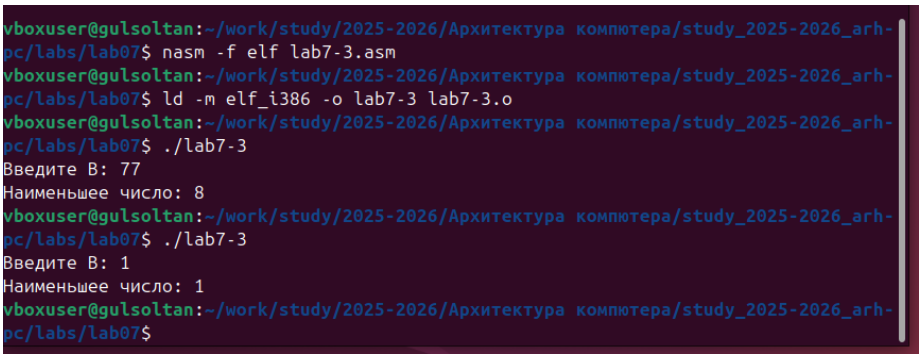



*Рис. 18: Создания исполняемого файла lab7-3.asm*

### 4.3.2 Напишите программу, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции f(x) и выводит результат вычислений.

Для начала я создала файл и в него я написала программу, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции f(x) и выводит результат вычислений.

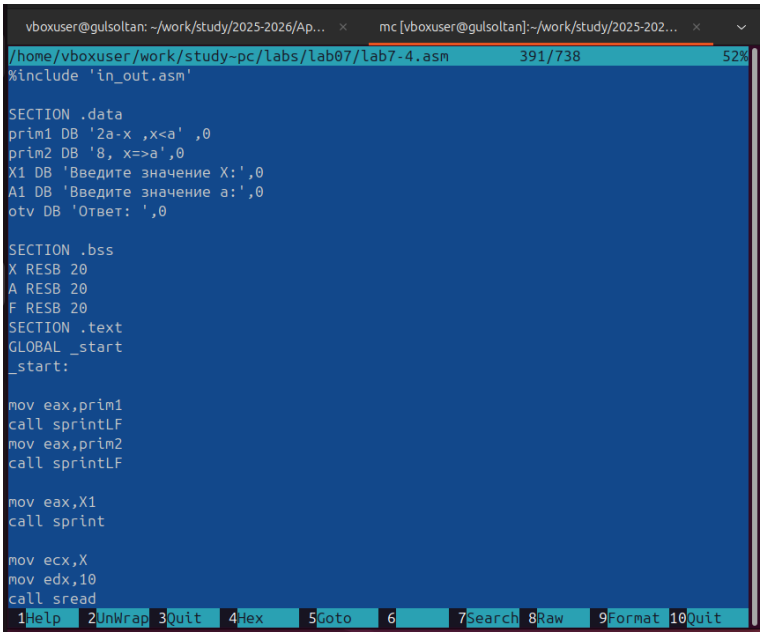



*Рис. 19: Внесения программы в файл lab7-4.asm*

После этого я создала исполняемый файл и запустила его. потом я написала цифры которые были таблице на X и на A

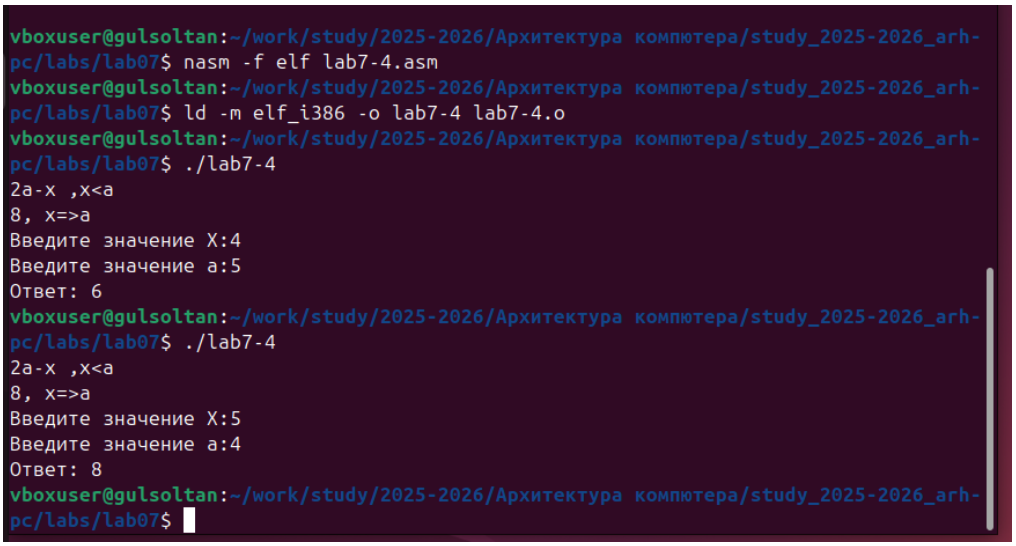



*Рис. 20: Создания исполняемого файла lab7-4.asm*

Все готова!

## 5 Выводы

Я изучила команды условного и безусловного перехода. Приобрела навыки написания программ с переходами.

## Список литературы

(https://esystem.rudn.ru)Архитектура компьютеров, Лабораторная работа №7