

Шаблон отчёта по лабораторной работе №7

Дисциплина: архитектура компьютера

Чашемова Гульназик

Содержание

1	Цель работы	1
2	Задание	1
3	Теоретическое введение.....	1
4	Выполнение лабораторной работы.....	2
4.1	Изучение структуры файлы листинга.....	2
4.2	Изучение структуры файлы листинга.....	5
4.3	Самостоятельная работа.....	7
4.3.1	Напишите программу нахождения наименьшей из 3 целочисленных переменных a,b и с	7
4.3.2	Напишите программу, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции $f(x)$ и выводит результат вычислений.....	8
5	Выводы.....	9
	Список литературы	9

1 Цель работы

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

2 Задание

1. Реализация переходов в NASM
2. Изучение структуры файлы листинга

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 лабораторной работы

4.1 Изучение структуры файлы листинга

Для начала я создала каталог для программ Лабораторной работы. потом перешла в него и создала файл lab07-1.asm (рис. 1).

```
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a  
rch-pc$ cd labs  
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a  
rch-pc/labs$ cd lab07  
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a  
rch-pc/labs/lab07$ touch lab7-1.asm  
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a  
rch-pc/labs/lab07$ ls  
lab7-1.asm presentation report  
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a  
rch-pc/labs/lab07$
```

Рис. 1: Создания каталога и файла

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

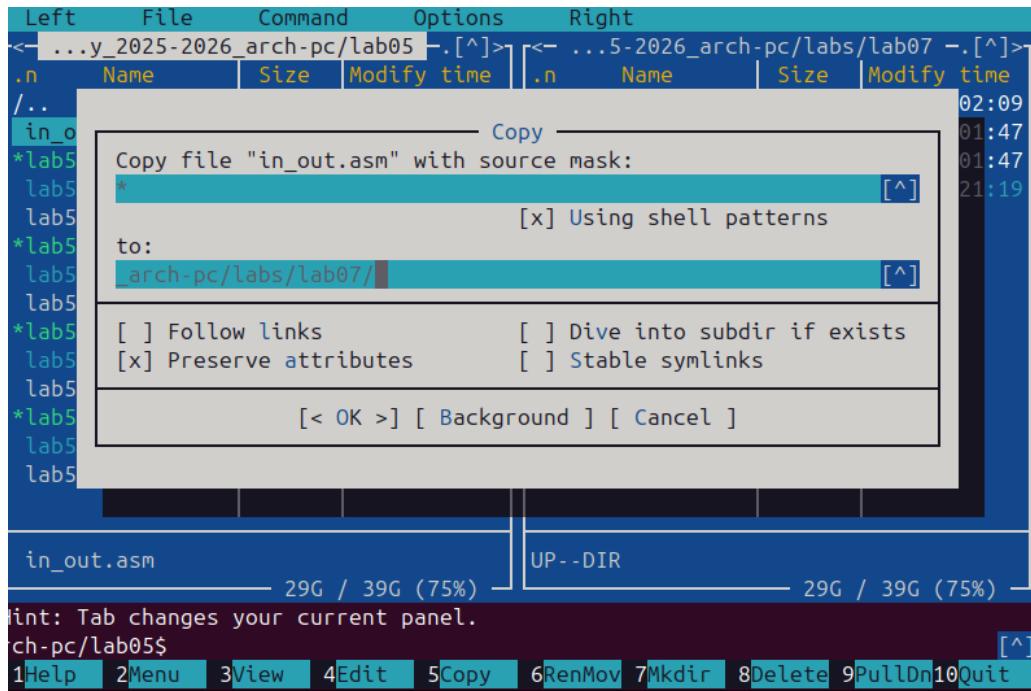


Рис. 2: Скопирования файла in_out.asm в нужный каталог

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

```
...6/Архитектура компьютера/study_2025-2026_arch-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 ; 'Сообщение № 1'
_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 2'
_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 3'
_end:
call quit ; вызов подпрограммы завершения

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

Рис. 3: Программа с использованием инструкции jmp

Потом я создала исполняемый файл и запустила его

```
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a
rch-pc/labs/lab07$ nasm-f elf lab7-1.asm
nasm-f: command not found
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a
rch-pc/labs/lab07$ nasm -f elf lab7-1.asm
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a
rch-pc/labs/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a
rch-pc/labs/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 3
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a
rch-pc/labs/lab07$
```

Рис. 4: Создания исполняемого файла

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

```
...6/Архитектура компьютера/study_2025-2026_arch-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 ^I Replace ^U Paste ^J Justify ^/ Go To Line

Рис. 5: Изменения текста файла

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

```
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a
rch-pc/labs/lab07$ mc

gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a
rch-pc/labs/lab07$ nasm -f elf lab7-1.asm
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a
rch-pc/labs/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a
rch-pc/labs/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 1
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a
rch-pc/labs/lab07$
```

Рис. 6: Создания(изменённого) исполняемого файла

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

```
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a
rch-pc/labs/lab07$ touch lab7-2.asm
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a
rch-pc/labs/lab07$
```

Рис. 7: Создания файла lab7-2.asm

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

```

...6/Архитектура компьютера/study_2025-2026_arch-pc/labs/lab07/lab7-2.asm *
%include 'in_out.asm'
section .data
msg1 db 'Введите В: ',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 В: '
mov eax,msg1
call sprint
; ----- Ввод 'В'
mov ecx,B
mov edx,10
call sread
; ----- Преобразование 'В' из символа в число

```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^I 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 запустила его. И ещё я проверила его работу

```

gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a
rch-pc/labs/lab07$ nasm -f elf lab7-2.asm
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a
rch-pc/labs/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a
rch-pc/labs/lab07$ ./lab7-2
Введите В: 10
Наибольшее число: 50
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a
rch-pc/labs/lab07$ ./lab7-2
Введите В: 80
Наибольшее число: 80
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a
rch-pc/labs/lab07$ 

```

Рис. 9: Создания исполняемого файла lab7-2.asm

4.2 Изучение структуры файлы листинга

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

```

gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a
rch-pc/labs/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a
rch-pc/labs/lab07$ 

```

Рис. 10: Создания листинга

Потом открыла файл листинга с помощью mcedit и изучила содержимое

```
gulnazik@gulnazik: ~/work/study/2025-2026/A... x gulnazik@gulnazik: ~/work/study/2025-2026/A... x v
/home/gu-7-2.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>
```

1Help 2Save 3Mark 4Replace 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 a “call iprint” — обозначает вызов функции 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.

```
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a  
rch-pc/labs/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm  
lab7-2.asm:34: error: invalid combination of opcode and operands  
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a  
rch-pc/labs/lab07$
```

Рис. 15: Ошибка в программе

В файле листинга показывает где ошибка и с чем оно связана

```
34      *****  
d operands      mov eax,  
35 00000130 E867FFFFF error: invalid combination of opcode an  
call atoi ; Вызов подпрограммы перевода
```

Рис. 16: Осмотр листинга

4.3 Самостоятельная работа.

4.3.1 Напишите программу нахождения наименьшей из 3 целочисленных переменных а,б и с.

Для начала я создала файл и в него я написала программу нахождения наименьшей из 3 целочисленных переменных а,б и с.

```
/home/gulnazik/work/stu-/labs/lab07/lab7-3.asm      539/1748      30%  
%include 'in_out.asm'  
SECTION .data  
msg1 DB 'Введите В: ',0h  
msg2 DB "Наименьшее число: ",0h  
A dd '8'  
C dd '68'  
SECTION .bss  
min resb 10  
B resb 10  
SECTION .text  
GLOBAL _start  
_start:  
; ----- Вывод сообщения 'Введите В: '  
mov eax,msg1  
call sprint  
; ----- Ввод 'В'  
mov ecx,B  
mov edx,10  
call sread  
; ----- Преобразование 'В' из символа в число  
mov eax,B  
call atoi ; Вызов подпрограммы перевода символа в число
```

Рис. 17: Внесения программы в файл

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

```

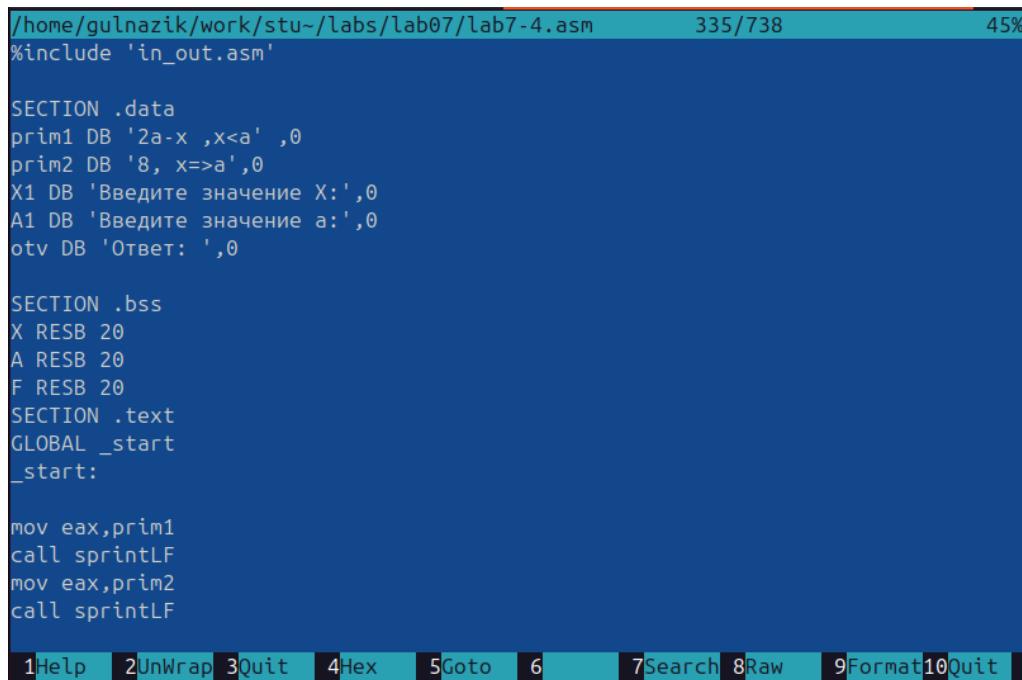
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a
rch-pc/Labs/Lab07$ nasm -f elf lab7-3.asm
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a
rch-pc/Labs/Lab07$ ld -m elf_i386 -o lab7-3 lab7-3.o
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a
rch-pc/Labs/Lab07$ ./lab7-3
Введите В: 8
Наименьшее число: 8
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a
rch-pc/Labs/Lab07$ ./lab7-3
Введите В: 5
Наименьшее число: 5
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a
rch-pc/Labs/Lab07$ 

```

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.



```

/home/gulnazik/work/stu~/labs/lab07/lab7-4.asm      335/738      45%
%include 'in_out.asm'

SECTION .data
prim1 DB '2a-x ,x<a' ,0
prim2 DB '8, x=>a',0
X1 DB 'Введите значение X:',0
A1 DB 'Введите значение a:',0
otv DB 'Ответ: ',0

SECTION .bss
X RESB 20
A RESB 20
F RESB 20
SECTION .text
GLOBAL _start
_start:

    mov eax,prim1
    call sprintLF
    mov eax,prim2
    call sprintLF

```

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

```
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a  
rch-pc/labs/lab07$ nasm -f elf lab7-4.asm  
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a  
rch-pc/labs/lab07$ ld -m elf_i386 -o lab7-4 lab7-4.o  
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a  
rch-pc/labs/lab07$ ./lab7-4  
2a-x ,x<a  
8, x=>a  
Введите значение X:6  
Введите значение a:9  
Ответ: 12  
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a  
rch-pc/labs/lab07$ ./lab7-4  
2a-x ,x<a  
8, x=>a  
Введите значение X:9  
Введите значение a:6  
Ответ: 8  
gulnazik@gulnazik:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a  
rch-pc/labs/lab07$
```

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