

Отчет по лабораторной работе №7

Архитектура Компьютера

Михальский Кирилл Алексеевич

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Задание для самостоятельной работы	13
6	Выводы	15
	Список литературы	16

Список иллюстраций

4.1	Создание	8
4.2	Ввод	9
4.3	Создание	9
4.4	Изменение	10
4.5	Создание	10
4.6	Создание	11
4.7	Создание	11
4.8	Создание	12
4.9	Удаление	12
5.1	Текст	13
5.2	Запуск	14
5.3	Запуск	14

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 Задание

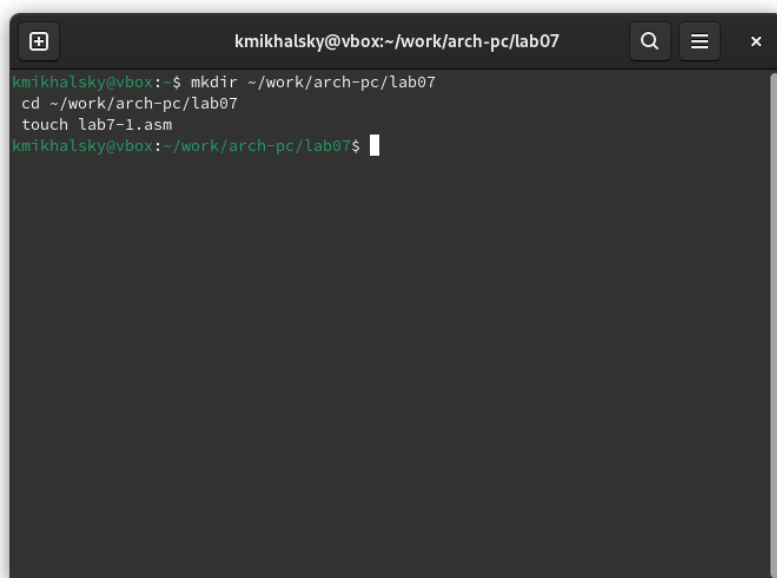
1. Реализация переходов в NASM
2. Изучение структуры файлов листинга
3. Самостоятельное написание программ по материалам лабораторной работы

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 лабораторной работы

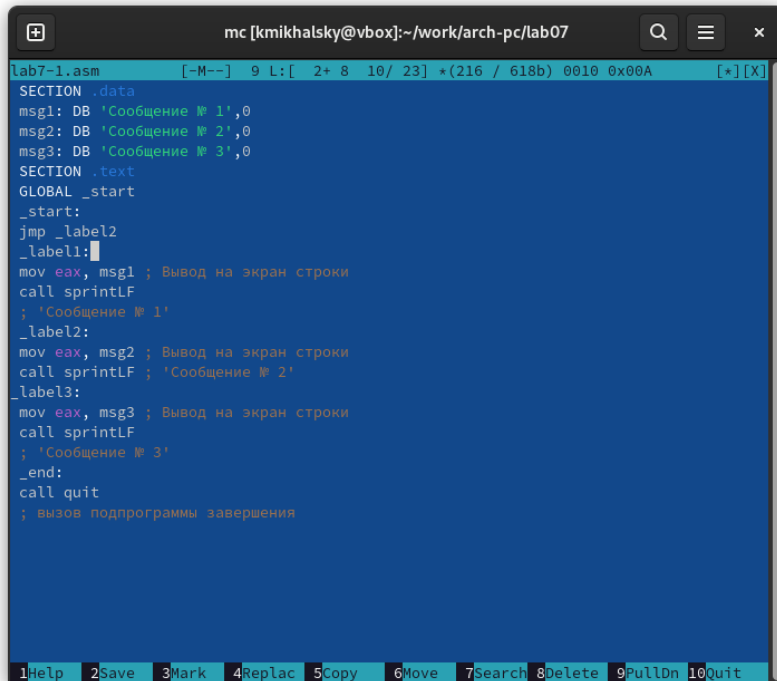
1. Создал каталог и новый файл. (рис. 4.1).



```
kmikhalsky@vbox:~/work/arch-pc/lab07
kmikhalsky@vbox:~$ mkdir ~/work/arch-pc/lab07
cd ~/work/arch-pc/lab07
touch lab7-1.asm
kmikhalsky@vbox:~/work/arch-pc/lab07$
```

Рис. 4.1: Создание

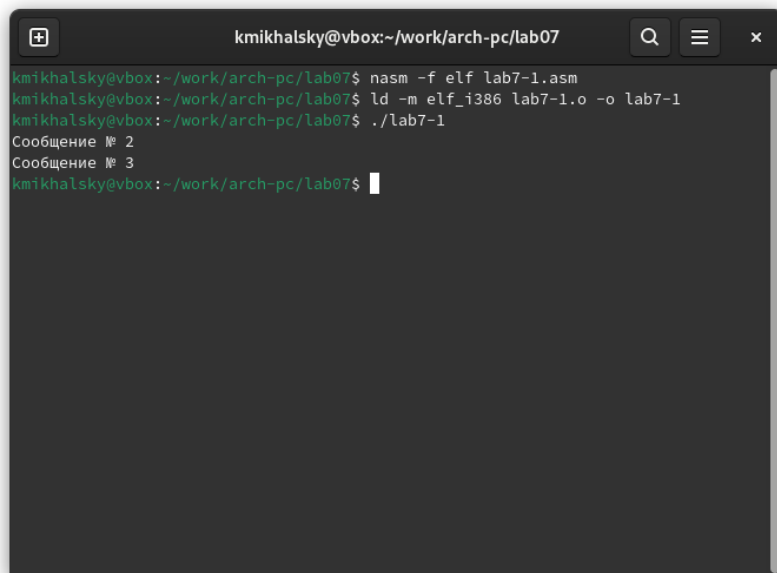
2. Ввел текст программы из листинга 7.1



```
lab7-1.asm [-M--] 9 L: [ 2+ 8 10/ 23] *(216 / 618b) 0010 0x00A [*][X]
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 sprintf
; 'Сообщение № 1'
_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintf ; 'Сообщение № 2'
_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintf
; 'Сообщение № 3'
_end:
call quit
; вызов подпрограммы завершения
```

Рис. 4.2: Ввод

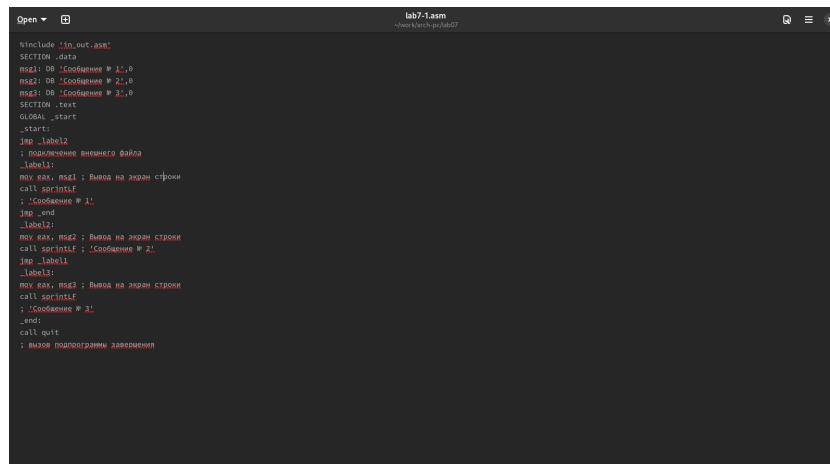
Создал исполняемый файл и запустил:



```
kmikhalsky@vbox:~/work/arch-pc/lab07
kmikhalsky@vbox:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
kmikhalsky@vbox:~/work/arch-pc/lab07$ ld -m elf_i386 lab7-1.o -o lab7-1
kmikhalsky@vbox:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 3
kmikhalsky@vbox:~/work/arch-pc/lab07$
```

Рис. 4.3: Создание

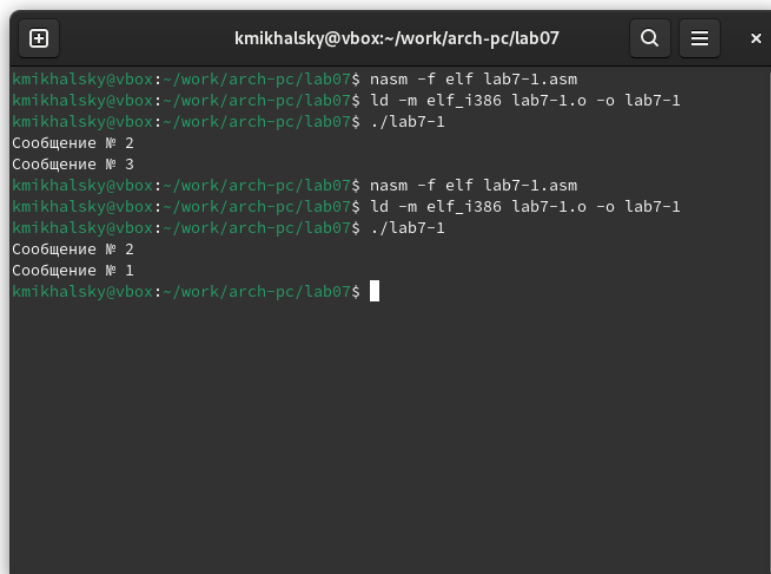
Изменил текст программы в соответствии с листингом 7.2



```
Open 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 _label2
; получение имени файла
_label1:
mov eax, msg1 ; Вывод на экран строки
call _printf
; Сообщение № 1
jmp _end
_label2:
mov eax, msg2 ; Вывод на экран строки
call _printf ; Сообщение № 2
jmp _label1
_label3:
mov eax, msg3 ; Вывод на экран строки
call _printf ; Сообщение № 3
_end:
call _quit
; Выход программы завершения
```

Рис. 4.4: Изменение

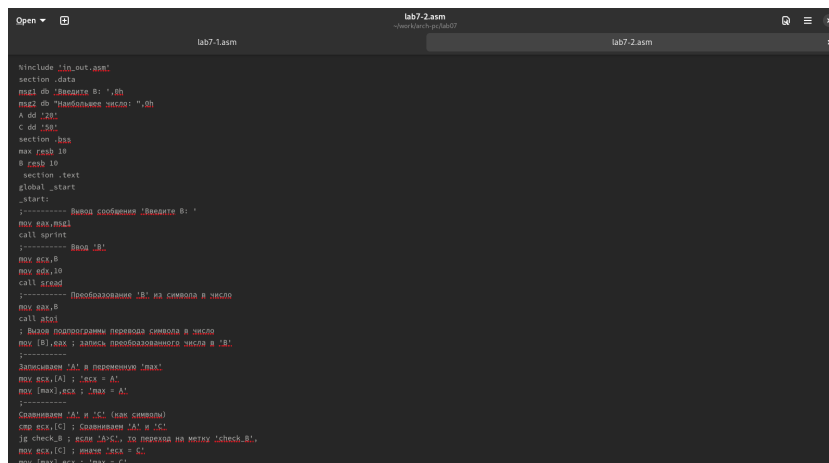
Создал исполняемый файл и проверил работу файла:



```
kmikhalsky@vbox:~/work/arch-pc/lab07
kmikhalsky@vbox:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
kmikhalsky@vbox:~/work/arch-pc/lab07$ ld -m elf_i386 lab7-1.o -o lab7-1
kmikhalsky@vbox:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 3
kmikhalsky@vbox:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
kmikhalsky@vbox:~/work/arch-pc/lab07$ ld -m elf_i386 lab7-1.o -o lab7-1
kmikhalsky@vbox:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 1
kmikhalsky@vbox:~/work/arch-pc/lab07$
```

Рис. 4.5: Создание

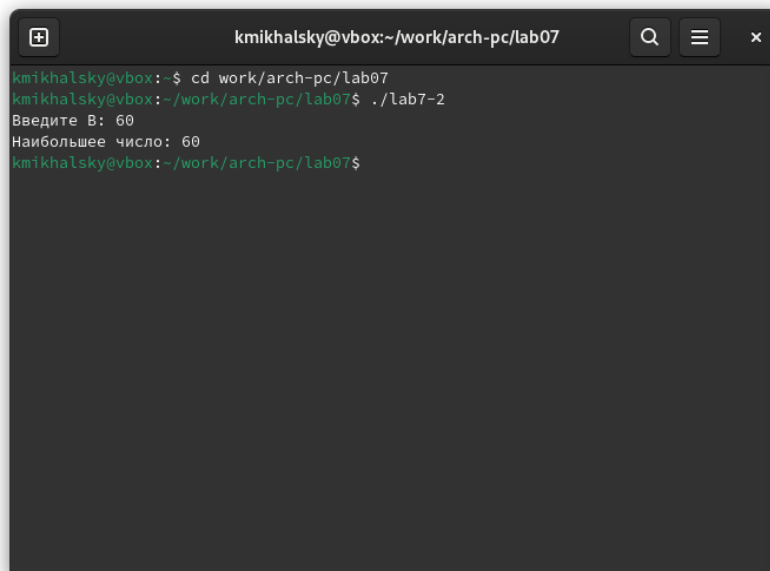
3. Создал файл lab7-2.asm и ввел текст из листинга 7.3



```
Open  lab7.1.asm  lab7.2.asm  lab7.2.asm
#include <in_out.asm>
section .data
msg1 db "Введите B: ",0h
msg2 db "Наибольшее число: ",0h
A dd 120
C dd 0
section .bss
max resb 10
B resb 10
section .text
global _start
_start:
;----- Вывод сообщения "Введите B: "
mov eax,msg1
call printf
;----- Ввод "B"
mov ecx,B
mov ebx,10
call read
;----- Преобразование "B" из символа в число
mov ecx,B
call atoi
; Вывод 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"
;-----
;Сравниваем "A" и "C" (как символы)
cmp ecx,C ; Сравниваем "A" и "C"
jg check_B ; если "A">"C", то переход на метку "check_B"
mov ecx,C ; иначе "ecx = C"
mov [max],ecx ; "max = C"
```

Рис. 4.6: Создание

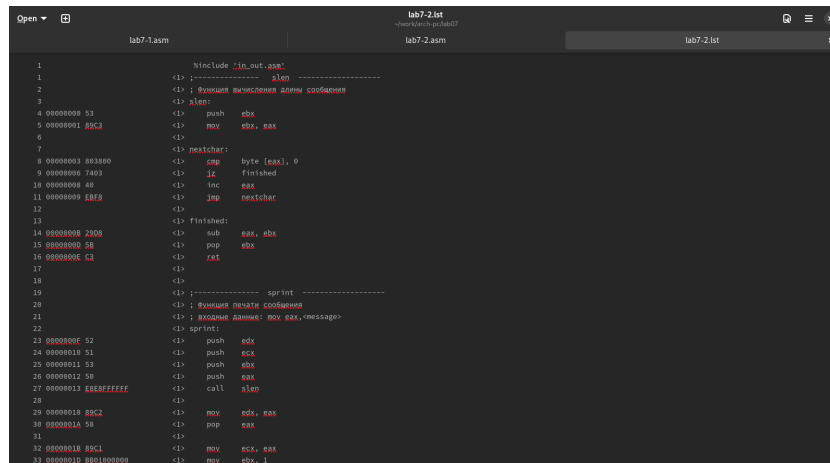
Создал исполняемый файл и запустил.



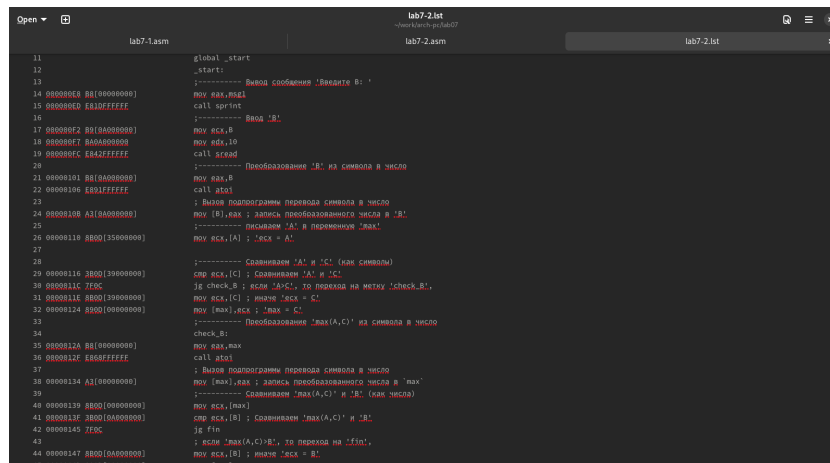
```
kmikhalsky@vbox:~/work/arch-pc/lab07
kmikhalsky@vbox:~$ cd work/arch-pc/lab07
kmikhalsky@vbox:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 60
Наибольшее число: 60
kmikhalsky@vbox:~/work/arch-pc/lab07$
```

Рис. 4.7: Создание

4. Создал файл листинга и открыл его:

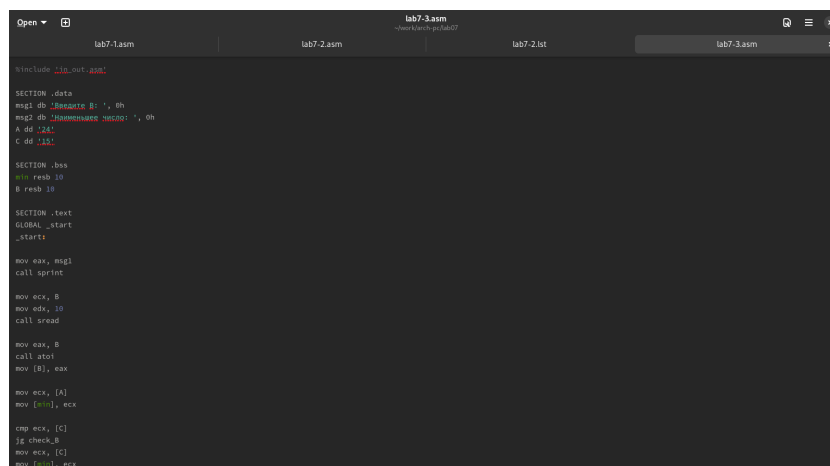


Удалил один операнд и проверил новый файл листинга.



5 Задание для самостоятельной работы

1. Текст программы из задания 1:



```
Open ▾
lab7-1.asm lab7-2.asm lab7-2.lst lab7-3.asm
lab7-3.asm
asm8086 -o lab7.o lab7.asm

#include <io.h>

SECTION .data
msg1 db "Language B: ", 0h
msg2 db "Filename: ", 0h
A db "A"
C db "C"

SECTION .bss
B resb 10

SECTION .text
GLOBAL _start
_start:

mov eax, msg1
call sprintf

mov ecx, B
mov edx, 10
call read

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

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

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

Рис. 5.1: Текст

Работа программы:

```
kmikhalsky@vbox:~/work/arch-pc/lab07
kmikhalsky@vbox:~/work/arch-pc/lab07$ cd work/arch-pc/lab07
kmikhalsky@vbox:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 60
Наибольшее число: 60
kmikhalsky@vbox:~/work/arch-pc/lab07$ nasm -f elf-l lab7-2.lst lab7-2.asm
bash: nasm-f: command not found...
kmikhalsky@vbox:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
kmikhalsky@vbox:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
kmikhalsky@vbox:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
kmikhalsky@vbox:~/work/arch-pc/lab07$ touch lab7-3.asm
kmikhalsky@vbox:~/work/arch-pc/lab07$ nasm -f elf lab7-3.asm
kmikhalsky@vbox:~/work/arch-pc/lab07$ ld -m elf_i386 lab7-3.o -o lab7-3
kmikhalsky@vbox:~/work/arch-pc/lab07$ ./lab7-3
Введите B: 98
Наименьшее число: 15
kmikhalsky@vbox:~/work/arch-pc/lab07$
```

Рис. 5.2: Запуск

2. Текст прогааммы из задания 2. Работа программы:

```
kmikhalsky@vbox:~/work/arch-pc/lab07$ nasm -f elf lab7-4.asm
kmikhalsky@vbox:~/work/arch-pc/lab07$ ld -m elf_i386 lab7-4.o -o lab7-4
kmikhalsky@vbox:~/work/arch-pc/lab07$ ./lab7-4
Введите значение переменной x: 5
Введите значение переменной a: 7
Результат: 12
kmikhalsky@vbox:~/work/arch-pc/lab07$ ./lab7-4
Введите значение переменной x: 6
Введите значение переменной a: 4
Результат: 4
kmikhalsky@vbox:~/work/arch-pc/lab07$
```

Рис. 5.3: Запуск

6 Выводы

При выполнении лабораторной работы я изучил команды условных и безусловных переходов, а также приобрел навыки написания программ с использованием переходов, познакомился с назначением и структурой файлов листинга.

Список литературы