

Отчет по лабораторной работе №7

Архитектура компьютера

Исаханян Армен Артурович

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	7
4	Объяснение трех строк	11
5	Выводы	12
	Список литературы	13

Список иллюстраций

3.1	Создание каталога и переход в него	7
3.2	Создание и открытие файла	7
3.3	Ввод программы в файл	8
3.4	Запуск файла	8
3.5	Редактирование программы	9
3.6	Запуск файла	9
3.7	Создание файла	9
3.8	Ввод программы	10
3.9	Создание файла листинга	10
3.10	Открытие файла листинга	10
4.1	Переход значения	11
4.2	Сравнение числа из регистра с 0	11
4.3	Вызов функции	11

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 # Задание

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

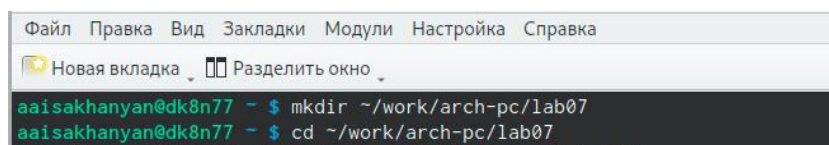
2 Теоретическое введение

Для реализации в ассемблере используются так называемые команды передачи управления или команды перехода. Можно выделить только 2 типа переходов:

- 1) Условный переход - выполнение или не выполнение перехода в определенную точку программы в зависимости от проверки условия.
- 2) Безусловный переход - выполнение передачи управления в определенную точку программы без каких-либо условий.

3 Выполнение лабораторной работы

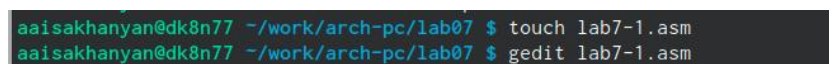
Создаю каталог и перехожу в него. (рис. 3.1).



```
Файл  Правка  Вид  Закладки  Модули  Настройка  Справка
Новая вкладка  Разделить окно
aaisakhanyan@dk8n77 ~ $ mkdir ~/work/arch-pc/lab07
aaisakhanyan@dk8n77 ~ $ cd ~/work/arch-pc/lab07
```

Рис. 3.1: Создание каталога и переход в него

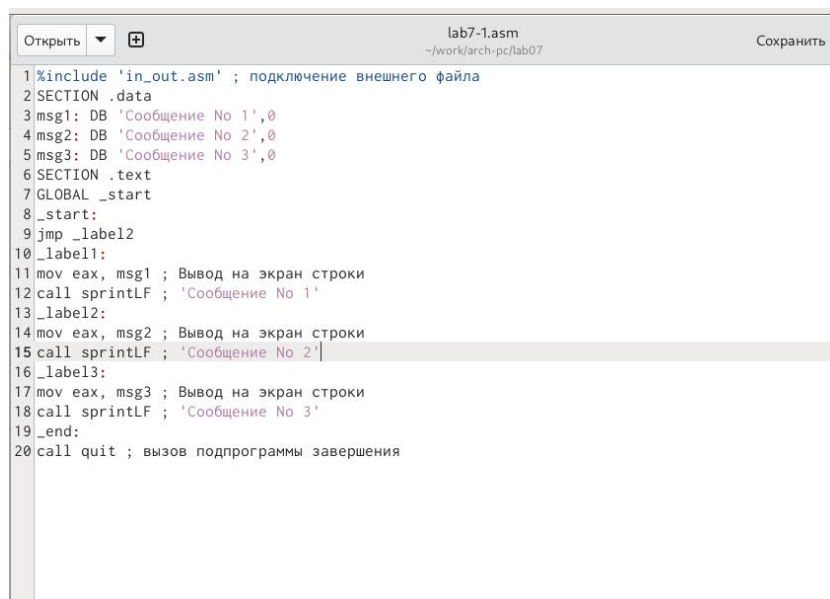
Создал файл lab7-1.asm и открываю его. (рис. 3.2).



```
aaisakhanyan@dk8n77 ~/work/arch-pc/lab07 $ touch lab7-1.asm
aaisakhanyan@dk8n77 ~/work/arch-pc/lab07 $ gedit lab7-1.asm
```

Рис. 3.2: Создание и открытие файла

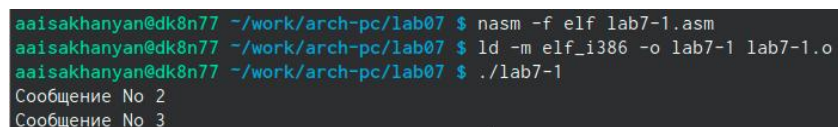
Ввел в файл программу (рис. 3.3).



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

Рис. 3.3: Ввод программы в файл

Создал исполняемый файл и запустил его (рис. 3.4).



```
aaisakhanyan@dk8n77 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
aaisakhanyan@dk8n77 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
aaisakhanyan@dk8n77 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение No 2
Сообщение No 3
```

Рис. 3.4: Запуск файла

Редактировал программу в файле lab7-1.asm (рис. 3.5).

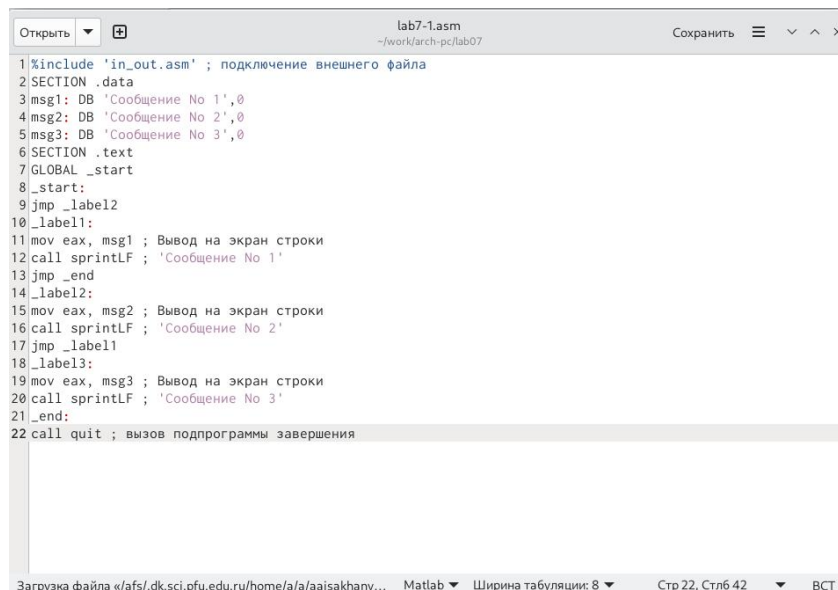


Рис. 3.5: Редактирование программы

Создаю исполняемый файл с отредактированной программой и запускаю его (рис. 3.6).

```

aaisakhanyan@dk8n77 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
aaisakhanyan@dk8n77 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
aaisakhanyan@dk8n77 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение No 2
Сообщение No 1
aaisakhanyan@dk8n77 ~/work/arch-pc/lab07 $

```

Рис. 3.6: Запуск файла

Создал файл lab7-2.asm в том же каталоге (рис. 3.7).

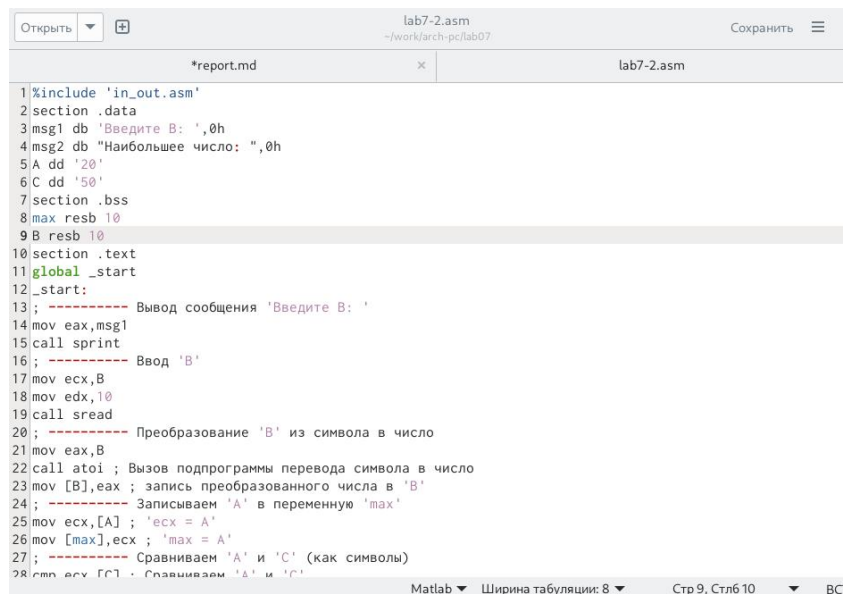
```

Файл Правка Вид Закладки Модули Настройка Справка
Новая вкладка Разделить окно
aaisakhanyan@dk8n77 ~ $ cd ~/work/arch-pc/lab07
aaisakhanyan@dk8n77 ~/work/arch-pc/lab07 $ touch lab7-2.asm

```

Рис. 3.7: Создание файла

Ввел программу в этот файл (рис. 3.8).



```
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'
```

Рис. 3.8: Ввод программы


Создал файл листинга для программы из файла lab7-2.asm указав ключ -l и задав имя файла листинга в командной строке (рис. 3.9).



```
aaisakhanyan@dk8n77 ~/work/arch-pc/lab07 $ nasm -f elf -l lab7-2.lst lab7-2.asm
```

Рис. 3.9: Создание файла листинга

Открыл файл листинга (рис. 3.10).



```
aaisakhanyan@dk8n77 ~/work/arch-pc/lab07 $ mcedit lab7-2.lst
```

Рис. 3.10: Открытие файла листинга

4 Объяснение трех строк

Из регистра `eax` перетаскиваю значение в регистр `ebx` (рис. 4.1).

```
> mov ebx, eax
```

Рис. 4.1: Переход значения

Сравниваю число из регистра `eax` с нулем (рис. 4.2).

```
> cmp byte [eax], 0
```

Рис. 4.2: Сравнение числа из регистра с 0

Вызываю функцию вычисления длины сообщения (рис. 4.3).

```
<1> call slen
```

Рис. 4.3: Вызов функции

5 Выводы

Изучил команды условного и безусловного переходов. Приобрел навыки написания программ с использованием переходов. Познакомился с назначением и структурой файла листинга.

Список литературы