

Лабораторная работа №7

Дисциплина: Архитектура компьютера

Жибицкая Евгения Дмитриевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Самостоятельная работа	10
4	Выводы	13

Список иллюстраций

2.1	Создание файла и каталога	6
2.2	Запуск файла lab7-1	6
2.3	Запуск измененной программы lab7-1	7
2.4	Изменение текста программы lab7-1	7
2.5	Запуск самостоятельно измененной программы lab7-1	7
2.6	Запуск и проверка работы программы lab7-2	8
2.7	Создание листинга	8
2.8	Файл листинга	9
2.9	Создание листинга 2	9
2.10	Просмотр листинга 2	9
3.1	Код для task7-1, часть 1	10
3.2	Код для task7-1, часть 2	10
3.3	Запуск task7-1	11
3.4	Код для task7-2, часть 1	11
3.5	Код для task7-2, часть 2	12
3.6	Запуск task7-2	12

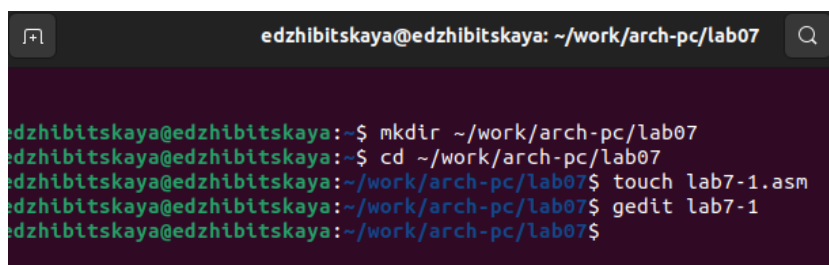
Список таблиц

1 Цель работы

Продолжение изучения языка ассемблера, знакомство с командами условного и безусловного переходов и с назначением и структурой файла листинга.

2 Выполнение лабораторной работы

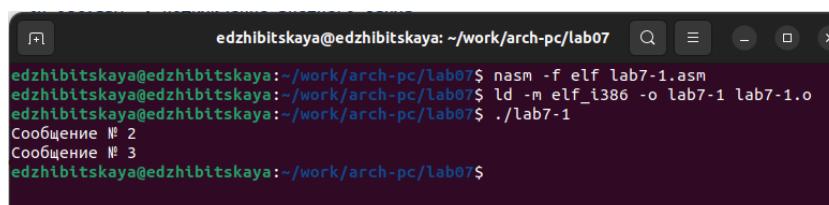
Для начала создадим каталог для 7 лабораторной работы, перейдем туда и создадим файл lab7-1.asm (рис. 2.1). Также не забудем скопировать файл in_out.asm.



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

Рис. 2.1: Создание файла и каталога

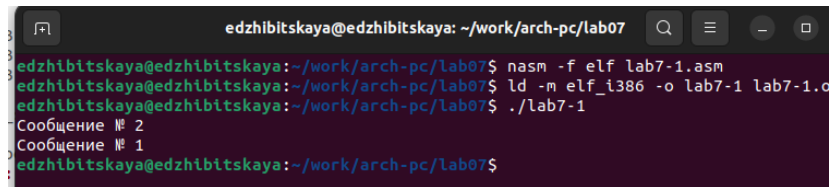
Заполним файл текстом Листинга 7.1, создадим объектный файл и запустим его(рис. 2.2).



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

Рис. 2.2: Запуск файла lab7-1

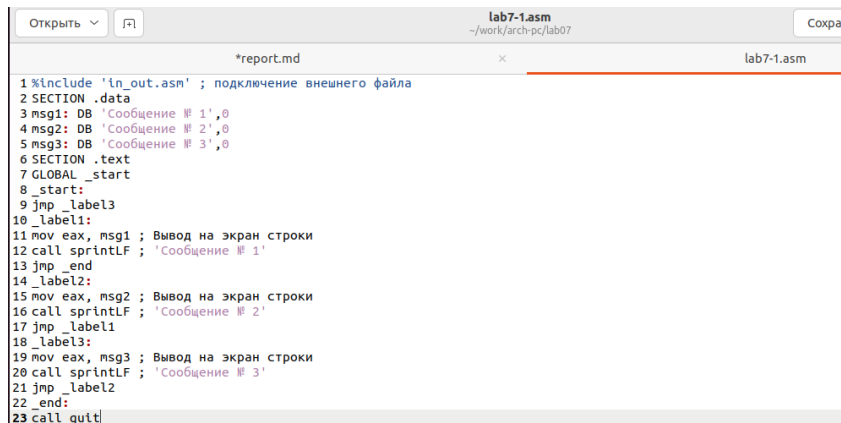
Заменяем содержимое файла на текст из Листинга 7-2, создадим объектный файл и запустим программу (рис. 2.3).



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

Рис. 2.3: Запуск измененной программы lab7-1

Изменим код так, чтобы выводилось 3-2-1 сообщения, запустим и проверим(рис. 2.4),(рис. 2.5).

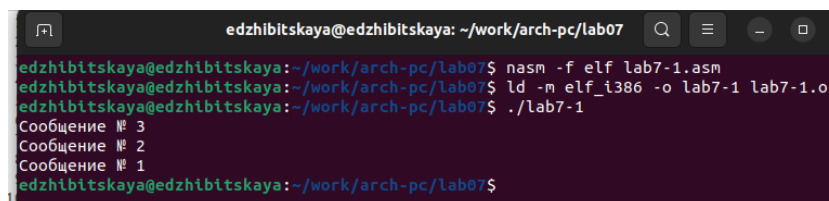


```
lab7-1.asm
~/work/arch-pc/lab07
Сохранить

*report.md
lab7-1.asm

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 _label3
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 jmp _label2
22 _end:
23 call quit
```

Рис. 2.4: Изменение текста программы lab7-1



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

Рис. 2.5: Запуск самостоятельно измененной программы lab7-1

Затем создадим файл lab7-2.asm, введем туда текст Листинга 7-3, создадим объектный файл и проверим корректность работы на нескольких числах(рис. 2.6).

```
edzhbitskaya@edzhbitskaya: ~/work/arch-pc/lab07
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab07$ touch lab7-2.asm
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab07$ gedit lab7-2.asm
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 1
Наибольшее число: 50
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 60
Наибольшее число: 60
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab07$ 43
43: команда не найдена
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 43
Наибольшее число: 50
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab07$
```

Рис. 2.6: Запуск и проверка работы программы lab7-2

Командой `nasm` создадим объектный файл и, указав ключ `-l` и задав имя файла, файл листинга (рис. 2.7).

```
edzhbitskaya@edzhbitskaya: ~/work/arch-pc/lab07
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab07$ mcedit lab7-2.lst
```

Рис. 2.7: Создание листинга

Командой `mcedit` откроем файл листинга, изучим его содержимое(рис. 2.8).

Опишем из чего состоят 8-10 строки. 8 строка состоит только из номера строки и исходного текста программы. 9 и 10 строки содержат номер, затем адрес(8цифр), машинный код и исходный текст программы.


```

edzhibitskaya@edzhibitskaya: ~/work/arch-pc/lab07
/home/edzhibitskaya/lab7-2.lst [----] 0 L: [1+ 0 1/225] *(0 /14445b) 0032 0x020 [*][X]
1      %include 'in_out.asm'
2      <1> ;----- slen -----
3      <1> ; Функция вычисления длины сообщения
4      <1> slen:-----
5      00000000 53      <1> push    ebx
6      00000001 89C3    <1> mov     ebx, eax
7      <1> .....
8      <1> nextchar:-----
9      00000003 803800  <1> cmp     byte [eax], 0
10     00000006 7403    <1> jz      finished
11     00000008 40      <1> inc     eax
12     00000009 EBF8    <1> jmp     nextchar
13     <1> .....
14     <1> finished:-----
15     0000000B 29D8    <1> sub     eax, ebx
16     0000000D 5B      <1> pop     ebx
17     0000000E C3      <1> ret
18     <1> .....
19     <1> .....
20     <1> ;----- sprint -----
21     <1> ; Функция печати сообщения
22     <1> ; входные данные: mov eax, <message>
терминал
1Помощь 2Сох-ить 3Блок 4Замена 5Копия 6Пер-ить 7Поиск 8Удалить 9ИенюМС 10Выход

```

Рис. 2.8: Файл листинга

В конце в инструкции с двумя операндами удалим один операнд, оттранслируем его и получим листинг(рис. 2.9),(рис. 2.10).

```

edzhibitskaya@edzhibitskaya: ~/work/arch-pc/lab07
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
lab7-2.asm:38: error: invalid combination of opcode and operands
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab07$ mcedit lab7-2.lst

```

Рис. 2.9: Создание листинга 2

```

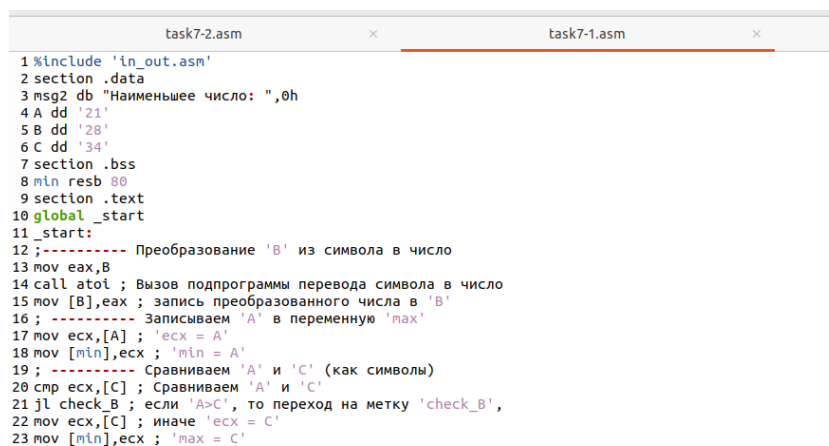
edzhibitskaya@edzhibitskaya: ~/work/arch-pc/lab07
/home/edzhibitskaya/lab7-2.lst [B---] 90 L: [205+ 9 214/226] *(13715/14530b) 0010 0x00A [*][X]
30     00000124 8B0D[39000000] mov ecx,[C] ; иначе 'ecx = C'
31     0000012A 890D[00000000] mov [max],ecx ; 'max = C'
32     ; ----- Преобразование 'max(A,C)' из символа в ч
33     check_B:
34     00000130 B8[00000000] mov eax,max
35     00000135 E862FFFFFF call atoi ; Вызов подпрограммы перевода символа в чис
36     0000013A A3[00000000] mov [max],eax ; запись преобразованного числа в 'max'
37     ; ----- Сравниваем 'max(A,C)' и 'B' (как числа)
38     mov ecx
39     *****
40     0000013F 3B0D[0A000000] cmp ecx,[B] ; Сравниваем 'max(A,C)' и 'B'
41     00000145 7F0C    jg fin ; если 'max(A,C)>B', то переход на 'fin',
42     00000147 8B0D[0A000000] mov ecx,[B] ; иначе 'ecx = B'
43     0000014D 890D[00000000] mov [max],ecx
44     ; ----- Вывод результата
45     fin:
46     00000153 B8[13000000] mov eax, msg2
47     00000158 E8B2FFFFFF call sprint ; вывод сообщения 'Наибольшее число: '

```

Рис. 2.10: Просмотр листинга 2

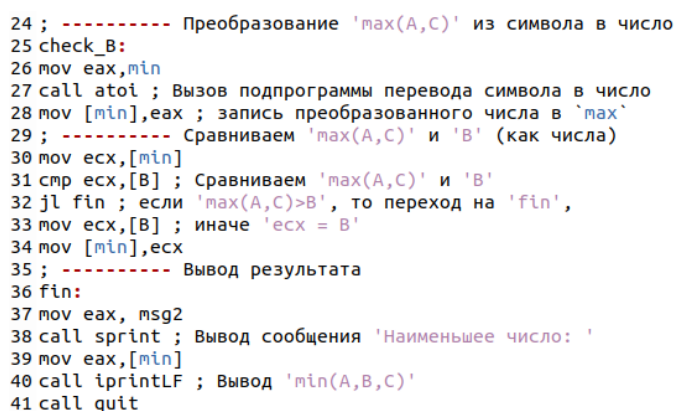
3 Самостоятельная работа

Для выполнения первого задания создадим файл task7-1, напомним код для вычисления наименьшего значения(рис. 3.1),(рис. 3.2) и проверим его работу(рис. 3.3).



```
1 %include 'in_out.asm'
2 section .data
3 msg2 db "Наименьшее число: ",0h
4 A dd '21'
5 B dd '28'
6 C dd '34'
7 section .bss
8 min resb 80
9 section .text
10 global _start
11 _start:
12 ;----- Преобразование 'B' из символа в число
13 mov eax,B
14 call atoi ; Вызов подпрограммы перевода символа в число
15 mov [B],eax ; запись преобразованного числа в 'B'
16 ; ----- Записываем 'A' в переменную 'max'
17 mov ecx,[A] ; 'ecx = A'
18 mov [min],ecx ; 'min = A'
19 ; ----- Сравниваем 'A' и 'C' (как символы)
20 cmp ecx,[C] ; Сравниваем 'A' и 'C'
21 jl check_B ; если 'A>C', то переход на метку 'check_B',
22 mov ecx,[C] ; иначе 'ecx = C'
23 mov [min],ecx ; 'max = C'
```

Рис. 3.1: Код для task7-1, часть 1



```
24 ; ----- Преобразование 'max(A,C)' из символа в число
25 check_B:
26 mov eax,min
27 call atoi ; Вызов подпрограммы перевода символа в число
28 mov [min],eax ; запись преобразованного числа в 'max'
29 ; ----- Сравниваем 'max(A,C)' и 'B' (как числа)
30 mov ecx,[min]
31 cmp ecx,[B] ; Сравниваем 'max(A,C)' и 'B'
32 jl fin ; если 'max(A,C)>B', то переход на 'fin',
33 mov ecx,[B] ; иначе 'ecx = B'
34 mov [min],ecx
35 ; ----- Вывод результата
36 fin:
37 mov eax, msg2
38 call sprint ; Вывод сообщения 'Наименьшее число: '
39 mov eax,[min]
40 call iprintLF ; Вывод 'min(A,B,C)'
41 call quit
```

Рис. 3.2: Код для task7-1, часть 2

```
1 %include 'in_out.asm'
2 section .data
3 msg2 db "Наименьшее число: ",0h
4 A dd '21'
5 B dd '28'
6 C dd '34'
7 section .bss

edzhbitskaya@edzhbitskaya: ~/work/arch-pc/lab07
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab07$ nasm -f elf task7-1.asm
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab07$ ld -m elf_i386 -o task7-1 task7-1.o
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab07$ ./task7-1
Наименьшее число: 21
edzhbitskaya@edzhbitskaya:~/work/arch-pc/lab07$
```

Рис. 3.3: Запуск task7-1

Проделаем аналогичные действия для задания 2(рис. 3.4),(рис. 3.5) (рис. 3.6).

```
task7-2.asm

1 %include 'in_out.asm'
2 SECTION .data
3 msg: DB 'Введите x: ',0
4 msg1: DB 'Введите a: ',0
5 msg2: DB 'Результат: ',0
6 SECTION .bss
7 x: RESB 80
8 a: RESB 80
9 otv: RESB 80
10 SECTION .text
11 GLOBAL _start
12 _start:
13 mov eax, msg
14 call sprintf
15 mov ecx, x
16 mov edx, 80
17 call sread
18 mov eax,x
19 call atoi
20 mov [x], eax
21 mov eax, msg1
22 call sprintf
23 mov ecx, a
24 mov edx, 80
```

Рис. 3.4: Код для task7-2, часть 1

```

25 call sread
26 mov eax,a
27 call atoi
28 mov [a], eax
29 ; ---- Вычисление выражения
30 mov eax, [a]
31 mov ebx,4 ;ebx =4
32 mul ebx ;EAX=EAX*EBX
33 mov ebx, 0
34 cmp [x],ebx
35 je ANS
36 add eax,[x]
37
38 ANS:
39 ; ---- Вывод результата на экран
40 mov [otv], eax
41 mov eax, msg2 ; вызов подпрограммы печати
42 call sprintLF ; сообщения 'Результат: '
43 mov eax,[otv] ; вызов подпрограммы печати значения
44 call iprintLF ;
45 call quit ; вызов подпрограммы завершения

```

Рис. 3.5: Код для task7-2, часть 2

```

edzhibitskaya@edzhibitskaya: ~/work/arch-pc/lab07
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab07$ nasm -f elf task7-2.asm
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab07$ ld -m elf_i386 -o task7-2 task7-2.o
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab07$ ./task7-2
Введите x:
0
Введите a:
3
Результат:
12
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab07$ nasm -f elf task7-2.asm
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab07$ ld -m elf_i386 -o task7-2 task7-2.o
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab07$ ./task7-2
Введите x:
1
Введите a:
2
Результат:
9
edzhibitskaya@edzhibitskaya:~/work/arch-pc/lab07$

```

Рис. 3.6: Запуск task7-2

4 Выводы

В ходе работы было произведено знакомство с командами переходов, структурой файла листинга. Также были написаны команды с использованием переходов.