

Отчёт по лабораторной работе №7

Архитектура компьютеров и операционные системы.

Брыляков Никита Евгеньевич

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выполнение лабораторной работы	7
4.1	Реализация переходов в NASM	7
4.2	Изучение структуры файлы листинга	11
4.3	Выполнение заданий для самостоятельной работы	13
5	Вывод	15
6	Список литературы	16

Список иллюстраций

4.1	Создание каталога и файла внутри	7
4.2	Ввод программы	7
4.3	Создание и запуск	8
4.4	Изменение программы	8
4.5	Создание и запуск	9
4.6	Изменение программы	9
4.7	Создание и запуск	10
4.8	Создание файла	10
4.9	Ввод программы	10
4.10	Создание и запуск	11
4.11	Создание файла листинга	11
4.12	Открытие файла	12
4.13	Строки кода	12
4.14	Удаление операнда	12
4.15	Получение файла листинга	12
4.16	Написание программы	13
4.17	Создание и проверка	14

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. Выполнение заданий для самостоятельной 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. Безусловный переход выполняется инструкцией `jmp`. Адрес перехода может быть либо меткой, либо адресом области памяти, в которую предварительно помещен указатель перехода. Кроме того, в качестве операнда можно использовать имя регистра, в таком случае переход будет осуществляться по адресу, хранящемуся в этом регистре. Как отмечалось выше, для условного перехода необходима проверка какого-либо условия. В ассемблере команды условного перехода вычисляют условие перехода анализируя флаги из регистра флагов. Листинг (в рамках понятийного аппарата NASM) — это один из выходных файлов, создаваемых транслятором. Он имеет текстовый вид и нужен при отладке программы, так как кроме строк самой программы он содержит дополнительную информацию.

4 Выполнение лабораторной работы

4.1 Реализация переходов в NASM

Создаю каталог для программ лабораторной работы № 7, перехожу в него и создаю файл lab7-1.asm (рис. [4.1]).

```
(nebrihlyakov@nebrihlyakov)-[~]  
$ mkdir ~/work/arch-pc/lab07  
  
(nebrihlyakov@nebrihlyakov)-[~]  
$ cd ~/work/arch-pc/lab07  
  
(nebrihlyakov@nebrihlyakov)-[~/work/arch-pc/lab07]  
$ touch lab7-1.asm
```

Рис. 4.1: Создание каталога и файла внутри

Ввожу в файл lab7-1.asm текст программы из листинга 7.1. (рис. [4.2]).

```
Л07_Брыляков_отчет.md lab7-1.asm x  
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 _label2  
10 _label1:  
11 mov eax, msg1 ; Вывод на экран строки  
12 call sprintf ; 'Сообщение № 1'  
13 _label2:  
14 mov eax, msg2 ; Вывод на экран строки  
15 call sprintf ; 'Сообщение № 2'  
16 _label3:  
17 mov eax, msg3 ; Вывод на экран строки  
18 call sprintf ; 'Сообщение № 3'  
19 _end:  
20 call quit ; вызов подпрограммы завершения  
21
```

Рис. 4.2: Ввод программы

Создаю исполняемый файл и запускаю его. (рис. [4.3]).

```
(nebrihlyakov@nebrihlyakov)~[~/work/arch-pc/lab07]
$ nasm -f elf lab7-1.asm

(nebrihlyakov@nebrihlyakov)~[~/work/arch-pc/lab07]
$ ld -m elf_i386 -o lab7-1 lab7-1.o

(nebrihlyakov@nebrihlyakov)~[~/work/arch-pc/lab07]
$ ./lab7-1
Сообщение № 2
Сообщение № 3
```

Рис. 4.3: Создание и запуск

Изменяю текст программы в соответствии с листингом 7.2. (рис. [4.4]).

```
Л07_Брыляков_отчет.md  lab7-1.asm x
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 _label2
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 _end:
22 call quit ; вызов подпрограммы завершения
23 |
```

Рис. 4.4: Изменение программы

Создаю исполняемый файл и запускаю его. (рис. [4.5]).


```

(nebrihlyakov@nebrihlyakov)~[~/work/arch-pc/lab07]
$ nasm -f elf lab7-1.asm

(nebrihlyakov@nebrihlyakov)~[~/work/arch-pc/lab07]
$ ld -m elf_i386 -o lab7-1 lab7-1.o

(nebrihlyakov@nebrihlyakov)~[~/work/arch-pc/lab07]
$ ./lab7-1
Сообщение № 2
Сообщение № 1

```

Рис. 4.5: Создание и запуск

Редактирую программу так, чтобы вывод был как по заданию (рис. [4.6]).

```

Л07_Брыляков_отчет.md
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 ; вызов подпрограммы завершения
24 |

```

Рис. 4.6: Изменение программы

Создаю исполняемый файл и запускаю его. (рис. [4.7]).

```
(nebrihlyakov@nebrihlyakov)-[~/work/arch-pc/lab07]
$ nasm -f elf lab7-1.asm

(nebrihlyakov@nebrihlyakov)-[~/work/arch-pc/lab07]
$ ld -m elf_i386 -o lab7-1 lab7-1.o

(nebrihlyakov@nebrihlyakov)-[~/work/arch-pc/lab07]
$ ./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1
```

Рис. 4.7: Создание и запуск

Создаю файл lab7-2.asm в каталоге ~/work/arch-pc/lab07. (рис. [4.8]).

```
(nebrihlyakov@nebrihlyakov)-[~/work/arch-pc/lab07]
$ touch lab7-2.asm
```

Рис. 4.8: Создание файла

Изучаю текст программы из листинга 7.3 и ввожу в lab7-2.asm. (рис. [4.9]).

```
1 #include "lab7-1.asm"
2 section .data
3     msg db "Введите B: ",0h
4     msg2 db "Наибольшее число: ",0h
5     A db 0
6
7 section .bss
8     resb 10
9     resb 10
10 section .text
11 global _start
12 _start:
13     ; Вывод сообщения "Введите B: "
14     mov eax,msg
15     call printf
16     ;
17     mov ecx,B
18     mov ebx,0
19     call read
20
21     ; Преобразование 'B' из символа в число
22     mov ecx,B
23     call atoi ; Выход подпрограммы перевода символа в число
24     mov [B],eax ; Запись преобразованного числа в B
25     ; Инициализация 'A' в переменную 'max'
26     mov ecx,[A] ; 'max' = A
27     mov [max],ecx ; 'max' = A
28
29     ; Сравнение 'A' с 'C' (как символы)
30     mov ecx,[C] ; Сравнение 'A' с 'C'
31     cmp check_B,ecx ; Если 'A' < 'C', то переход на метку 'check_B'
32     mov ecx,[C] ; иначе 'max' = C
33     mov [max],ecx ; 'max' = C
34
35     ; Преобразование 'max(A,C)' из символа в число
36     mov ecx,[max]
37     call atoi ; Выход подпрограммы перевода символа в число
38     mov [max],eax ; Запись преобразованного числа в 'max'
39
40     ; Сравнение 'max(A,C)' с 'B' (как числа)
41     mov ecx,[max]
42     cmp ecx,[B] ; Сравнение 'max(A,C)' с 'B'
43     jg B_gt ; Если 'max(A,C)' > 'B', то переход на 'B_gt'
44     mov [B],ecx ; иначе 'max' = B
45     mov [max],ecx
46
47     ; Вывод результата
48     call B_gt
```

Рис. 4.9: Ввод программы

Создаю исполняемый файл и проверяю его работу для некоторых значений B. Программа работает корректно. (рис. [4.10]).

```
(nebrihlyakov@nebrihlyakov)-[~/work/arch-pc/lab07]
$ nasm -f elf lab7-2.asm

(nebrihlyakov@nebrihlyakov)-[~/work/arch-pc/lab07]
$ ld -m elf_i386 -o lab7-2 lab7-2.o

(nebrihlyakov@nebrihlyakov)-[~/work/arch-pc/lab07]
$ ./lab7-2
Введите В: 25
Наибольшее число: 50

(nebrihlyakov@nebrihlyakov)-[~/work/arch-pc/lab07]
$ ./lab7-2
Введите В: 36
Наибольшее число: 50
```

Рис. 4.10: Создание и запуск

4.2 Изучение структуры файлы листинга

Создаю файл листинга для программы из файла lab7-2.asm (рис. [4.11]).

```
(nebrihlyakov@nebrihlyakov)-[~/work/arch-pc/lab07]
$ nasm -f elf -l lab7-2.lst lab7-2.asm
```

Рис. 4.11: Создание файла листинга

Открываю файл листинга lab7-2.lst с помощью текстового редактора mousepad и ознакамливаюсь с его содержимым. (рис. [4.12]).

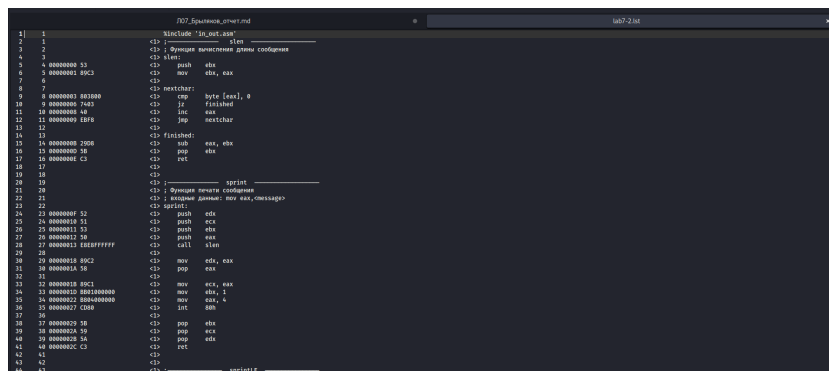


Рис. 4.12: Открытие файла

Объясню эти три строчки: “60” - номер строки кода, “; входные данные: mov eax, mov ebx,” - комментарий, “61” - номер строки кода, “sread:” - считывающая строку функция, “62” - номер строки кода, “00000043” - адрес строки, “push ebx” - текст самой программы (помещает значение регистра ‘EBX’ в стек. (рис. [4.13]).

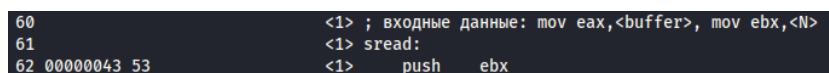


Рис. 4.13: Строки кода

Открываю файл с программой lab7-2.asm и в инструкции с двумя операндами удаляю операнд B (рис. [4.14]).

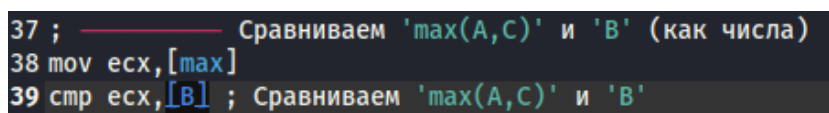


Рис. 4.14: Удаление операнда

Выполняю трансляцию с получением файла листинга (рис. [4.15]).

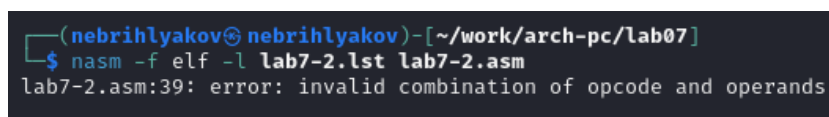
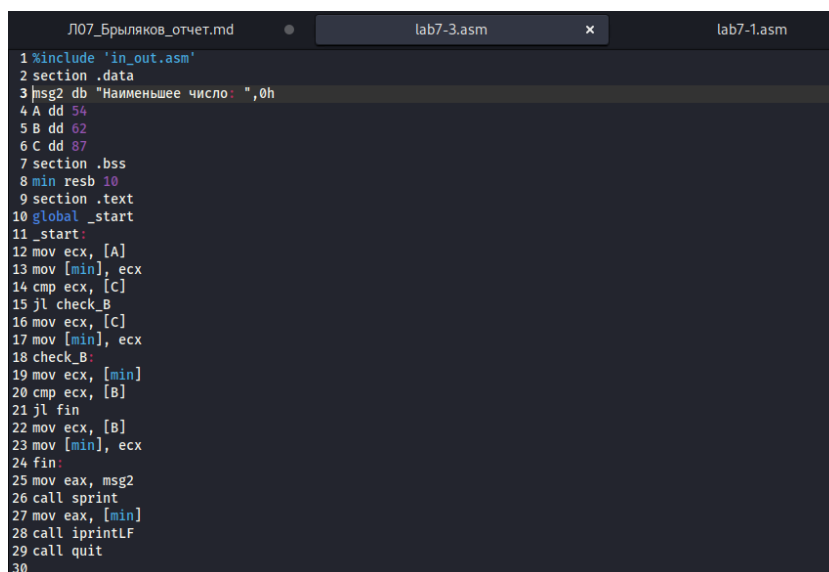


Рис. 4.15: Получение файла листинга

На выходе я не получаю ничего, так как инструкция не срабатывает из-за наличия только одного операнда.

4.3 Выполнение заданий для самостоятельной работы

Пишу программу в файле lab7-3.asm нахождения наименьшей из 3 целочисленных переменных a, b и c. Значения переменных выбираю из табл. 7.5 в соответствии с вариантом, полученным при выполнении предыдущей лабораторной работы. Мой вариант был 5 так что мои числа это 54, 62, 87. (рис. [4.16]).



```
1 %include 'in_out.asm'
2 section .data
3 msg2 db "Наименьшее число: ",0h
4 A dd 54
5 B dd 62
6 C dd 87
7 section .bss
8 min resb 10
9 section .text
10 global _start
11 _start:
12 mov ecx, [A]
13 mov [min], ecx
14 cmp ecx, [C]
15 jlt check_B
16 mov ecx, [C]
17 mov [min], ecx
18 check_B:
19 mov ecx, [min]
20 cmp ecx, [B]
21 jlt fin
22 mov ecx, [B]
23 mov [min], ecx
24 fin:
25 mov eax, msg2
26 call sprint
27 mov eax, [min]
28 call iprintf
29 call quit
30
```

Рис. 4.16: Написание программы

Создаю исполняемый файл и проверяю его работу. Всё верно. (рис. [4.17]).

```
(nebrihlyakov@nebrihlyakov)-[~/work/arch-pc/lab07]
$ nasm -f elf lab7-3.asm

(nebrihlyakov@nebrihlyakov)-[~/work/arch-pc/lab07]
$ ld -m elf_i386 -o lab7-3 lab7-3.o

(nebrihlyakov@nebrihlyakov)-[~/work/arch-pc/lab07]
$ ./lab7-3
Наименьшее число: 54
```

Рис. 4.17: Создание и проверка

5 Вывод

При выполнении данной лабораторной работы я изучил команды условного и безусловного переходов. Приобрёл навыки написания программ с использованием переходов. Познакомился с назначением и структурой файла листинга.

6 Список литературы

- [illegible]