

Лабораторная работа №7

**Команды безусловного и условного переходов в NASM.
Программирование ветвлений**

Павленко Сергей

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	7
4	Выводы	11
5	Задание для самостоятельной работы	12
6	Выводы по самостоятельной работе	14
	Список литературы	15

Список иллюстраций

3.1	1	7
3.2	2	7
3.3	3	8
3.4	4	8
3.5	5	9
3.6	6	9
3.7	7	10
5.1	8	12
5.2	9	13

Список таблиц

1 Цель работы

Изучение команд условного и безусловного переходов. Приобретение навыков написания

2 Теоретическое введение

Для реализации ветвлений в ассемблере используются так называемые команды передачи

- * условный переход - выполнение или не выполнение перехода в определенную точку п
- * безусловный переход - выполнение передачи управления в определенную точку прогр

3 Выполнение лабораторной работы

1. Создайте файл lab7-1.asm, в соответствующем каталоге touch lab7-1.asm

```
spavlenko@spavlenko:~/Desktop$ cd
spavlenko@spavlenko:~$ cd ~/work/study/2023-2024/Архитектура\ компьютера/study_2
023-2024_arhpc/labs/lab07/report/
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-202
4_arhpc/labs/lab07/report$ touch lab7-1.asm
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-202
4_arhpc/labs/lab07/report$
```

Рис. 3.1: 1

2. Инструкция jmp в NASM используется для реализации безусловных переходов. Рассмотрим пример программы с использованием инструкции jmp. Введите в файл lab7-1.asm текст программы из листинга 7.1. Создайте исполняемый файл и запустите его.

```
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-202
4_arhpc/labs/lab07/report$ nasm -f elf lab7-1.asm
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-202
4_arhpc/labs/lab07/report$ ld -m elf_i386 -o lab7-1 lab7-1.o
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-202
4_arhpc/labs/lab07/report$ ./lab7-1
Сообщение №2
Сообщение №3
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-202
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-202
4_arhpc/labs/lab07/report$
```

Рис. 3.2: 2

Измените текст программы в соответствии с листингом 7.2. Создайте исполняемый файл и проверьте его работу.

```

spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_4_arhpc/labs/lab07/report$ nasm -f elf lab7-1.asm
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_4_arhpc/labs/lab07/report$ ld -m elf_i386 -o lab7-1 lab7-1.o
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_4_arhpc/labs/lab07/report$ ./lab7-1
Сообщение №2
Сообщение №1
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_4_arhpc/labs/lab07/report$

```

Рис. 3.3: 3

Измените текст программы добавив или изменив инструкции `jmp`, чтобы вывод программы был следующим: `user@dk4n31:~$./lab7-1` Сообщение №3
Сообщение №2
Сообщение №1
`user@dk4n31:~$`

```

spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_4_arhpc/labs/lab07/report$ nasm -f elf lab7-1.asm
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_4_arhpc/labs/lab07/report$ ld -m elf_i386 -o lab7-1 lab7-1.o
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_4_arhpc/labs/lab07/report$ ./lab7-1
Сообщение №3
Сообщение №2
Сообщение №1
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_4_arhpc/labs/lab07/report$

```

Рис. 3.4: 4

- Использование инструкции `jmp` приводит к переходу в любом случае. Однако, часто при написании программ необходимо использовать условные переходы, т.е. переход должен происходить если выполнено какое-либо условие. В качестве примера рассмотрим программу, которая определяет и выводит на экран наибольшую из 3 целочисленных переменных: А, В и С. Значения для А и С задаются в программе, значение В вводится с клавиатуры. Создайте файл `lab7-2.asm` в каталоге `~/work/arch-pc/lab07`. Внимательно изучите текст программы из листинга 7.3 и введите в `lab7-2.asm`. Создайте исполняемый файл и проверьте его работу.


```

spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arhpc/labs/lab07/report$ nasm -f elf lab7-2.asm
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arhpc/labs/lab07/report$ ld -m elf_i386 -o lab7-2 lab7-2.o
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arhpc/labs/lab07/report$ ./lab7-2
Введите B2
Наибольшее число50
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arhpc/labs/lab07/report$

```

Рис. 3.5: 5

4. Обычно nasm создаёт в результате ассемблирования только объектный файл. Получить файл листинга можно, указав ключ -l и задав имя файла листинга в командной строке. Создайте файл листинга для программы из файла lab7-2.asm `nasm -f elf -l lab7-2.lst lab7-2.asm` Откройте файл листинга lab7-2.lst с помощью любого текстового редактора, например mcedit: `mcedit lab7-2.lst`

```

spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arhpc/labs/lab07/report$ nasm -f elf -l lab7-2.lst lab7-2.asm
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arhpc/labs/lab07/report$ mcedit lab7-2.lst
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arhpc/labs/lab07/report$

```

Рис. 3.6: 6

Внимательно ознакомиться с его форматом и содержимым. Подробно объяснить содержимое трёх строк файла листинга по выбору. Откройте файл с программой lab7-2.asm и в любой инструкции с двумя операндами удалите один операнд. Выполните трансляцию с получением файла листинга: `nasm -f elf -l lab7-2.lst lab7-2.asm` Какие выходные файлы создаются в этом случае? Что добавляется в листинге?

А `dd '20'` - присваивает переменной А значение '20' `!g check_B` - при `a>c` происходит переход на метку `%include 'in_out.asm'` - подключение внешнего файла, для упрощения хода работы

```
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_ar  
hpc/labs/lab07/report$ nasm -f elf -l lab7-2.lst lab7-2.asm  
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_ar  
hpc/labs/lab07/report$ ld -m elf_i386 -o lab7-2 lab7-2.o  
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_ar  
hpc/labs/lab07/report$ ./lab7-2  
Введите B5  
Наибольшее число20
```

Рис. 3.7: 7

4 Выводы

Таким образом, мы ознакомились с реализацией переходов, изучили структуры файлов листинга, научились находить наибольшее/наименьшее

5 Задание для самостоятельной работы

1. Напишите программу нахождения наименьшей из 3 целочисленных переменных a, b и c. Значения переменных выбрать из табл. 7.5 в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу.

```
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_ar  
hpc/labs/lab07/report$ nasm -f elf -l lab7-2 lab7-2.asm  
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_ar  
hpc/labs/lab07/report$ ld -m elf_i386 -o lab7-2 lab7-2.o  
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_ar  
hpc/labs/lab07/report$ ./lab7-2  
Наименьшее число: 41  
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_ar  
hpc/labs/lab07/report$
```

Рис. 5.1: 8

2. Напишите программу, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции f(x) и выводит результат вычислений. Вид функции f(x) выбрать из таблицы 7.6 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу для значений x и a из 7.6.

```

spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arhpc/labs/lab07/report$ nasm -f elf -l lab7-3 lab7-3.asm
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arhpc/labs/lab07/report$ ld -m elf_i386 -o lab7-3 lab7-3.o
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arhpc/labs/lab07/report$ ./lab7-3
Введите число x: 2
Введите число a: 2
f(x) = 4
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arhpc/labs/lab07/report$ ./lab7-3
Введите число x: 2
Введите число a: 1
f(x) = 10
spavlenko@spavlenko:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arhpc/labs/lab07/report$

```

Рис. 5.2: 9

6 Выводы по самостоятельной работе

В ходе самостоятельной работы, мы усвоили, как сравнивать значения и нашли выражения $f(x)$ при определенных условиях.

Список литературы

1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learning-bash-Shell-Programming-Nutshell/dp/0596009658>.
6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс,
- 11.
12. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
13. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
14. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВ- Петербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
15. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-

- е изд. — М. : МАКС Пресс, 2011. — URL: http://www.stolyarov.info/books/asm_unix.
16. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
17. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).