РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 7

дисциплина: Архитектура компьютера

Студент: Мартынова Милана Александровна

Группа: НКАбд-04-25

МОСКВА

2025 г.

Содержание

Список иллюстраций

1. Цель работы

Освоение применения команд условных и безусловных переходов при программировании. Формирование практических умений разработки программ с использованием механизма переходов. Ознакомление с назначением, форматом и содержанием файла листинга.

2. Порядок выполнения лабораторной работы

2.1 Реализация переходов в NASM

Создаю каталог для программ лабораторной работы №7(рис. 1).

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

Рис. 1: Создание каталога и файла для программы

Копирую код из листинга в файл будущей программы(рис. 2).

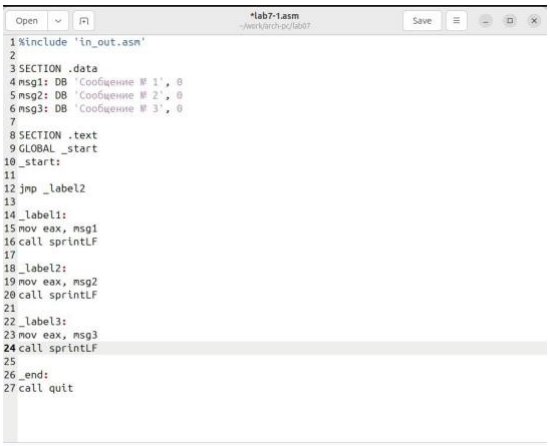

Рис 2: Сохранение программы

При запуске программы убедилась, что неусловный переход действительно изменяет порядок выполнения инструкций(рис. 3).

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

Рис. 3: Запуск программы

Изменяю программу таким образом, чтобы поменялся порядок выполнения функций(рис. 4).

```
Open   *lab7-1.asm   Save
~/work/arch-pc/lab07

1 %include 'in_out.asm'
2
3 SECTION .data
4 msg1: DB 'Сообщение № 1', 0
5 msg2: DB 'Сообщение № 2', 0
6 msg3: DB 'Сообщение № 3', 0
7
8 SECTION .text
9 GLOBAL _start
10 _start:
11
12 jmp _label2
13
14 _label1:
15 mov eax, msg1
16 call sprintLF
17 jmp _end
18
19 _label2:
20 mov eax, msg2
21 call sprintLF
22 jmp _label1
23
24 _label3:
25 mov eax, msg3
26 call sprintLF
27
28 _end:
29 call quit
```

Рис. 4: Изменение программы

Запуская программу и проверяю верность применённых изменений(рис. 5).

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

Рис. 5: Запуск измененной программы

Изменяю текст программы так, чтобы все три сообщения вывелись в обратном порядке(рис. 6).

```
Open   *lab7-1.asm   Save
~/work/arch-pc/lab07

1 %include 'in_out.asm'
2
3 SECTION .data
4 msg1: DB 'Сообщение № 1', 0
5 msg2: DB 'Сообщение № 2', 0
6 msg3: DB 'Сообщение № 3', 0
7
8 SECTION .text
9 GLOBAL _start
10 _start:
11
12 jmp _label3
13
14 _label1:
15 mov eax, msg1
16 call sprintLF
17 jmp _end
18
19 _label2:
20 mov eax, msg2
21 call sprintLF
22 jmp _label1
23
24 _label3:
25 mov eax, msg3
26 call sprintLF
27 jmp _label2
28
29 _end:
30 call quit
```

Рис. 6: Повторное изменение программы

Изменение выполнено корректно, программа выводит сообщения в нужном мне порядке(рис. 7).

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

Рис. 7: Проверка изменений

Создаю новый рабочий файл и вставляю в него код из следующего листинга(рис. 8).

Open | lab7-2.asm ~/work/arch-pc/lab07 | Save

```
%include 'in_out.asm'

SECTION .data
msg1 db 'Введите B: ', 0h
msg2 db 'Наибльшее число: ', 0h
A dd '20'
C dd '50'

SECTION .bss
max resb 10
B resb 10

SECTION .text
GLOBAL _start
_start:

mov eax, msg1
call sprint

mov ecx, B
mov edx, 10
call sread

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

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

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

check_B:
mov eax, max
call atoi
mov [max], eax

mov ecx, [max]
cmp ecx, [B]
jb fin
mov ecx, [B]
mov [max], ecx

fin:
mov eax, msg2
call sprint
mov eax, [max]
call iprintLF
call quit
```

Рис. 8: Сохранение новой программы

Программа выводит значение переменной с максимальным значением, проверяю работу программы с разными входными данными (рис. 9).

```
mamartynova@VirtualBox:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
mamartynova@VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
mamartynova@VirtualBox:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 25
Наибольшее число: 25
mamartynova@VirtualBox:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 60
Наибольшее число: 50
mamartynova@VirtualBox:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 10
Наибольшее число: 10
mamartynova@VirtualBox:~/work/arch-pc/lab07$
```

Рис. 9: Проверка программы листинга

2.2 Изучение структуры файлы листинга

Создаю файл листинга с помощью флага -l команды nasm и открываю его с помощью текстового редактора gedit (рис. 10, 11).

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

Рис. 10: Создание файла листинга

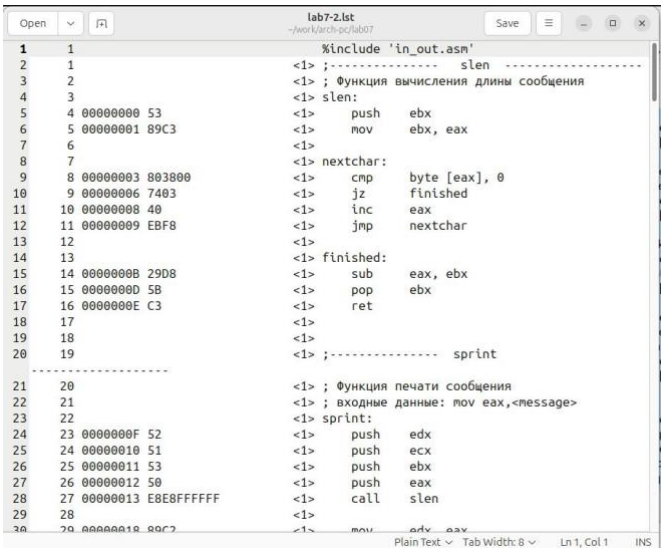

Рис. 11: Проверка файла листинга

Первое значение в файле листинга - номер строки, и он может не совпадать с номером строки изначального файла. Второе вхождение - адрес, смещение машинного кода относительно начала текущего сегмента, затем идет сам машинный код, а заключает строку исходный текст программы с комментариями.

Удаляю один операнд из случайной инструкции, чтобы проверить поведение файла листинга в дальнейшем (рис. 12).

Рис.12: Удаление операнда из программы

В новом файле листинга показывает ошибку, которая возникла при попытке трансляции файла. Никакие выходные файлы при этом помимо файла листинга не создаются. (рис. 13).

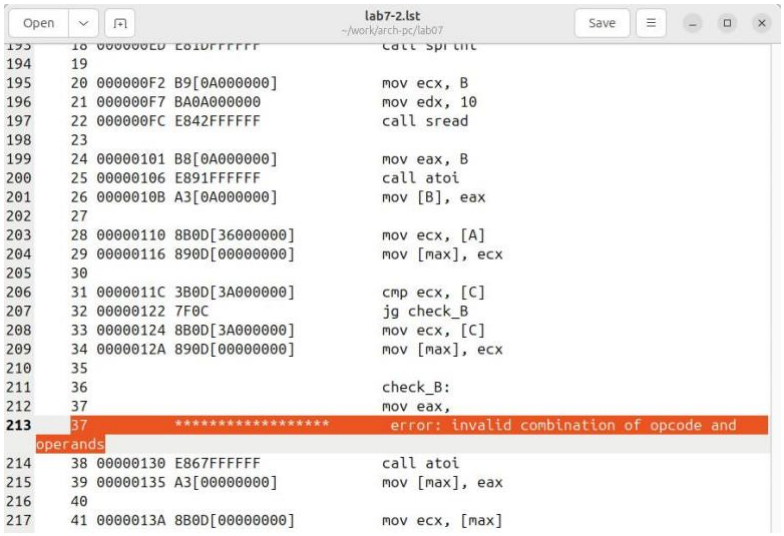

Рис. 13: Просмотр ошибки в файле листинга

2.3 Задания для самостоятельной работы

Возвращаю операнд к функции в программе и изменяю ее так, чтобы она выводила переменную с наименьшим значением (рис. 14).

Рис. 14: Первая программа самостоятельной работы

Проверяю корректность написания первой программы (рис. 15).

Рис. 15: Проверка работы первой программы

Пишу программу, которая будет вычислять значение заданной функции согласно моему варианту для введенных с клавиатурых переменных a и x (рис. 16).

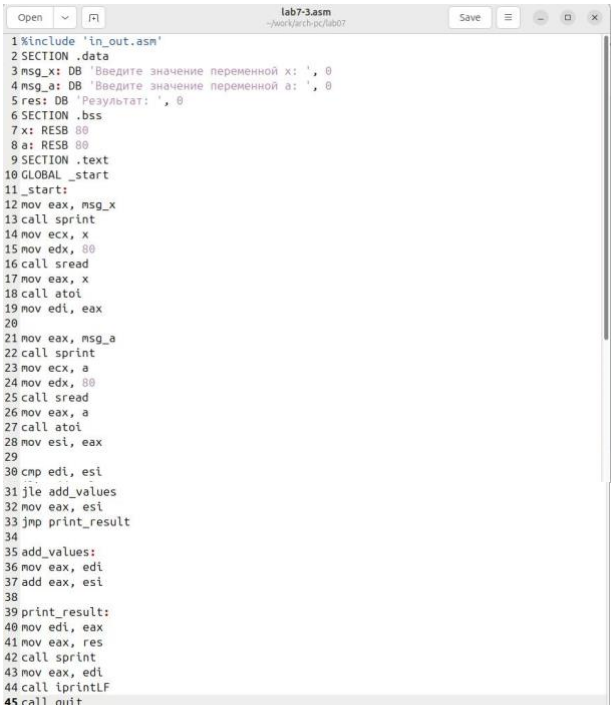

Рис. 16: Вторая программа самостоятельной работы

Транслирую и компоную файл, запускаю и проверяю работу программы для различных значений a и x (рис. 17).

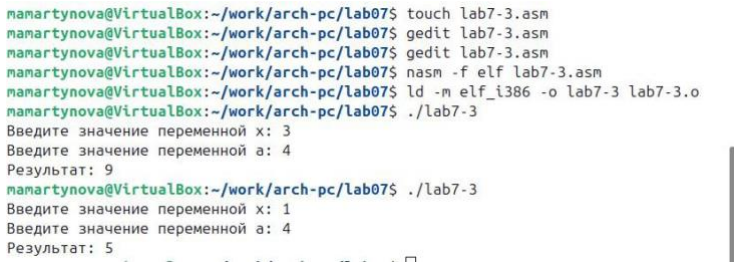

Рис. 17: Проверка работы второй программы

3. Выводы

В рамках лабораторной работы успешно освоено применение команд условных и безусловных переходов для управления логикой выполнения программ. Приобретен практический опыт написания и отладки кода с использованием ветвлений и циклов. Также проведен анализ файла листинга, что позволило понять взаимосвязь между исходным кодом и результирующим машинным кодом, а также использовать листинг для поиска и исправления ошибок.

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, 2017.
11. *Новожилов О. П.* Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM. — 2021. — URL: https://www.opennet.ru/docs/RUS/nasm/.
13. *Робачевский А.*, *Немнюгин С.*, *Стесик О.* Операционная система UNIX. — 2-е изд. — БХВ-Петербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
14. *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.
15. *Таненбаум Э.* Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
16. *Таненбаум Э.*, *Бос Х.* Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).