

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 8

дисциплина: Архитектура компьютера

Студент: Мартынова Милана Александровна

Группа: НКАбд-04-25

МОСКВА

2025 г.

Содержание

1. Цель работы.....	4
2. Порядок выполнения лабораторной работы.....	4
2.1 Реализация циклов в NASM.....	4
2.2 Обработка аргументов командной строки.....	6
2.3 Задания для самостоятельной работы.....	8
3. Выводы.....	8

Список иллюстраций

1. Создание каталога.....	4
2. Копирование программы из листинга.....	4
3. Запуск программы.....	4
4. Изменение программы.....	5
5. Запуск измененной программы.....	5
6. Добавление push и pop в цикл программы.....	5
7. Запуск измененной программы.....	6
8. Копирование программы из листинга.....	6
9. Запуск второй программы.....	6
10. Копирование программы из третьего листинга.....	7
11. Запуск третьей программы.....	7
12. Изменение третьей программы.....	7
13. Запуск измененной третьей программы.....	8
14. Написание программы для самостоятельной работы.....	8
15. Запуск программы для самостоятельной работы.....	8

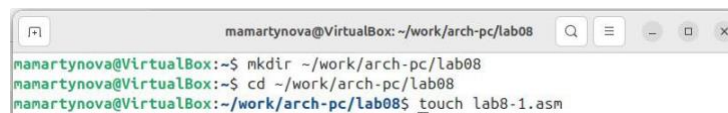
1. Цель работы

Получить практический опыт в написании кода с циклами и обработкой аргументов командной строки.

2. Порядок выполнения лабораторной работы

2.1 Реализация циклов в NASM

Создаю каталог для программ лабораторной работы №8 (рис. 1).



```
mamartynova@VirtualBox: ~/work/arch-pc/lab08
mamartynova@VirtualBox:~$ mkdir ~/work/arch-pc/lab08
mamartynova@VirtualBox:~$ cd ~/work/arch-pc/lab08
mamartynova@VirtualBox:~/work/arch-pc/lab08$ touch lab8-1.asm
```

Рис. 1: Создание каталога

Копирую в созданный файл программу из листинга. (рис. 2).




```
*lab8-1.asm
~/work/arch-pc/lab08

1 %include 'in_out.asm'
2 SECTION .data
3 msg1 db 'Введите N: ', 0h
4 SECTION .bss
5 N: resb 10
6 SECTION .text
7 GLOBAL _start
8 _start:
9
10 mov eax, msg1
11 call sprint
12
13 mov ecx, N
14 mov edx, 10
15 call sread
16
17 mov eax, N
18 call atoi
19 mov [N], eax
20
21 mov ecx, [N]
22
23 label:
24 mov [N], ecx
25 mov eax, [N]
26 call iprintLF
27 loop label
28
29 call quit
```

Рис. 2: Копирование программы из листинга

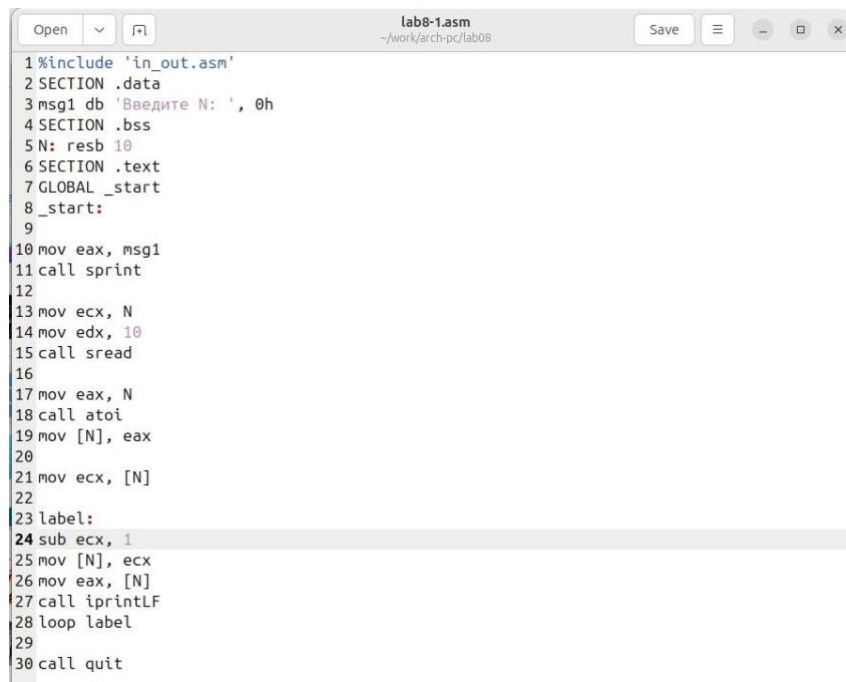
Запускаю программу, она показывает работу циклов в NASM (рис. 3).



```
mamartynova@VirtualBox:~/work/arch-pc/lab08$ gedit lab8-1.asm
mamartynova@VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
mamartynova@VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
mamartynova@VirtualBox:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 10
10
9
8
7
6
5
4
3
2
1
mamartynova@VirtualBox:~/work/arch-pc/lab08$
```

Рис. 3: Запуск программы

Заменяю программу изначальную так, что в теле цикла я изменяю значение регистра ecx (рис. 4).



```
1 %include 'in_out.asm'
2 SECTION .data
3 msg1 db 'Введите N: ', 0h
4 SECTION .bss
5 N: resb 10
6 SECTION .text
7 GLOBAL _start
8 _start:
9
10 mov eax, msg1
11 call sprint
12
13 mov ecx, N
14 mov edx, 10
15 call sread
16
17 mov eax, N
18 call atoi
19 mov [N], eax
20
21 mov ecx, [N]
22
23 label:
24 sub ecx, 1
25 mov [N], ecx
26 mov eax, [N]
27 call iprintLF
28 loop label
29
30 call quit
```

Рис.4: Изменение программы

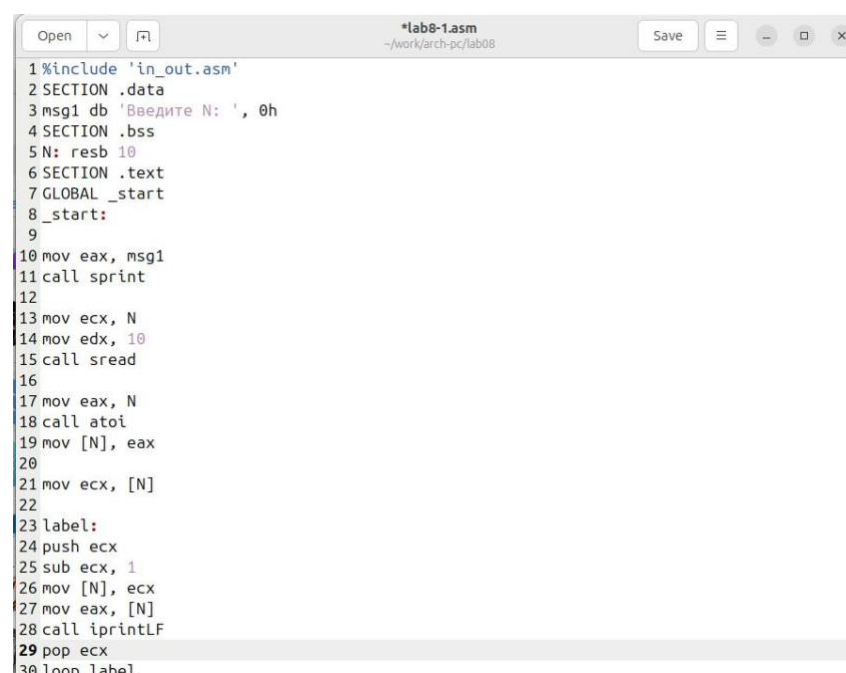
Из-за того, что теперь регистр ecx на каждой итерации уменьшается на 2 значения, количество итераций уменьшается вдвое (рис. 5).



```
mamartynova@VirtualBox:~/work/arch-pc/lab08$ gedit lab8-1.asm
mamartynova@VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
mamartynova@VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
mamartynova@VirtualBox:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 10
9
7
5
3
1
mamartynova@VirtualBox:~/work/arch-pc/lab08$
```

Рис. 5: Запуск измененной программы

Добавляю команды push и pop в программу (рис. 6).



```
1 %include 'in_out.asm'
2 SECTION .data
3 msg1 db 'Введите N: ', 0h
4 SECTION .bss
5 N: resb 10
6 SECTION .text
7 GLOBAL _start
8 _start:
9
10 mov eax, msg1
11 call sprint
12
13 mov ecx, N
14 mov edx, 10
15 call sread
16
17 mov eax, N
18 call atoi
19 mov [N], eax
20
21 mov ecx, [N]
22
23 label:
24 push ecx
25 sub ecx, 1
26 mov [N], ecx
27 mov eax, [N]
28 call iprintLF
29 pop ecx
30 loop label
```

Рис. 6: Добавление push и pop в цикл программы

Теперь количество итераций совпадает введенному N, но произошло смещение выводимых чисел на -1 (рис. 7).

```
mamartynova@VirtualBox:~/work/arch-pc/lab08$ gedit lab8-1.asm
mamartynova@VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
mamartynova@VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
mamartynova@VirtualBox:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 10
9
8
7
6
5
4
3
2
1
0
mamartynova@VirtualBox:~/work/arch-pc/lab08$
```

Рис. 7: Запуск измененной программы

2.2 Обработка аргументов командной строки

Создаю новый файл для программы и копирую в него код из следующего листинга (рис. 8).



```
1 %include 'in_out.asm'
2
3 SECTION .text
4 GLOBAL _start
5
6 _start:
7 pop ecx
8 pop edx
9 sub ecx, 1
10
11 next:
12 cmp ecx, 0
13 jz _end
14 pop eax
15 call sprintLF
16 loop next
17
18 _end:
19 call quit
```

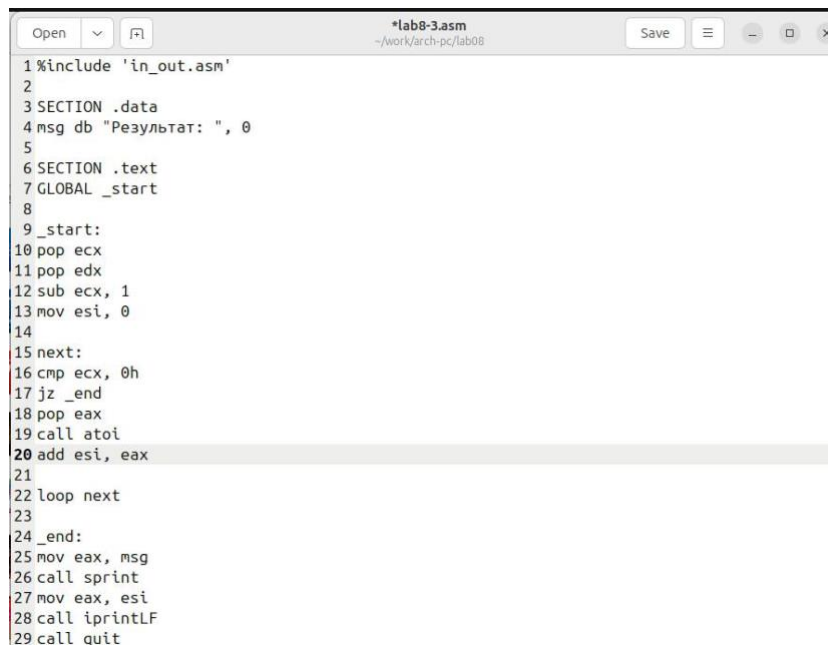
Рис. 8: Копирование программы из листинга

Компилирую программу и запускаю, указав аргументы. Программой было обратно то же количество аргументов, что и было введено (рис. 9).

```
mamartynova@VirtualBox:~/work/arch-pc/lab08$ touch lab8-2.asm
mamartynova@VirtualBox:~/work/arch-pc/lab08$ gedit lab8-2.asm
mamartynova@VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-2.asm
mamartynova@VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-2 lab8-2.o
mamartynova@VirtualBox:~/work/arch-pc/lab08$ ./lab8-2 arg1 arg 2 'arg 3'
arg1
arg
2
arg 3
mamartynova@VirtualBox:~/work/arch-pc/lab08$
```

Рис. 9: Запуск второй программы

Создаю новый файл для программы и копирую в него код из третьего листинга (рис. 10).



```
1 %include 'in_out.asm'
2
3 SECTION .data
4 msg db "Результат: ", 0
5
6 SECTION .text
7 GLOBAL _start
8
9 _start:
10 pop ecx
11 pop edx
12 sub ecx, 1
13 mov esi, 0
14
15 next:
16 cmp ecx, 0h
17 jz _end
18 pop eax
19 call atoi
20 add esi, eax
21
22 loop next
23
24 _end:
25 mov eax, msg
26 call sprint
27 mov eax, esi
28 call iprintLF
29 call quit
```

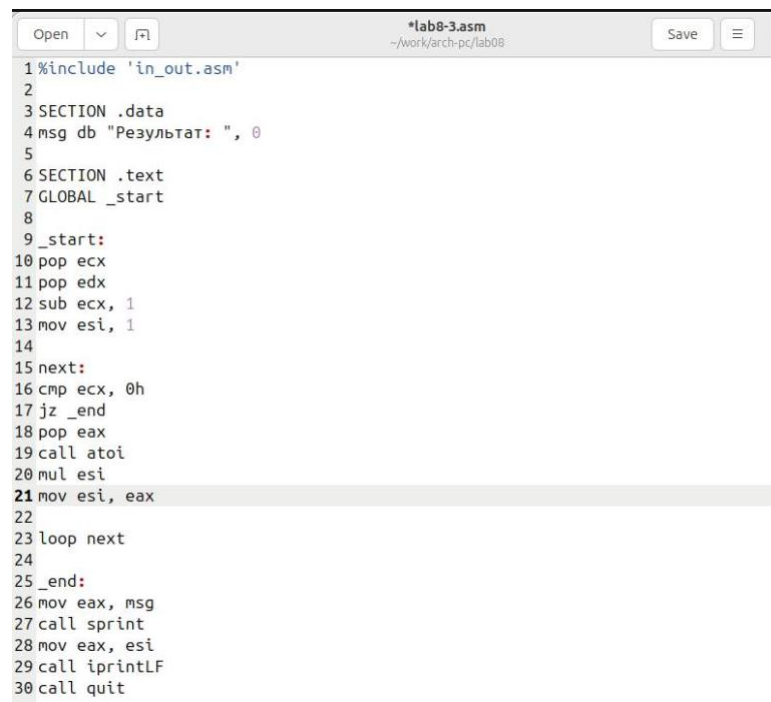
Рис. 10: Копирование программы из третьего листинга

Компилирую программу и запускаю, указав в качестве аргументов некоторые числа, программа их складывает (рис. 11).

```
mamartynova@VirtualBox:~/work/arch-pc/lab08$ touch lab8-3.asm
mamartynova@VirtualBox:~/work/arch-pc/lab08$ gedit lab8-3.asm
mamartynova@VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
mamartynova@VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
mamartynova@VirtualBox:~/work/arch-pc/lab08$ ./lab8-3 12 13 7 10 5
Результат: 47
```

Рис. 11: Запуск третьей программы

Изменяю поведение программы так, чтобы указанные аргументы она умножала, а не складывала (рис. 12).



```
1 %include 'in_out.asm'
2
3 SECTION .data
4 msg db "Результат: ", 0
5
6 SECTION .text
7 GLOBAL _start
8
9 _start:
10 pop ecx
11 pop edx
12 sub ecx, 1
13 mov esi, 1
14
15 next:
16 cmp ecx, 0h
17 jz _end
18 pop eax
19 call atoi
20 mul esi
21 mov esi, eax
22
23 loop next
24
25 _end:
26 mov eax, msg
27 call sprint
28 mov eax, esi
29 call iprintLF
30 call quit
```

Рис. 12: Изменение третьей программы

Программа действительно теперь умножает данные на вход числа (рис. 13).

```

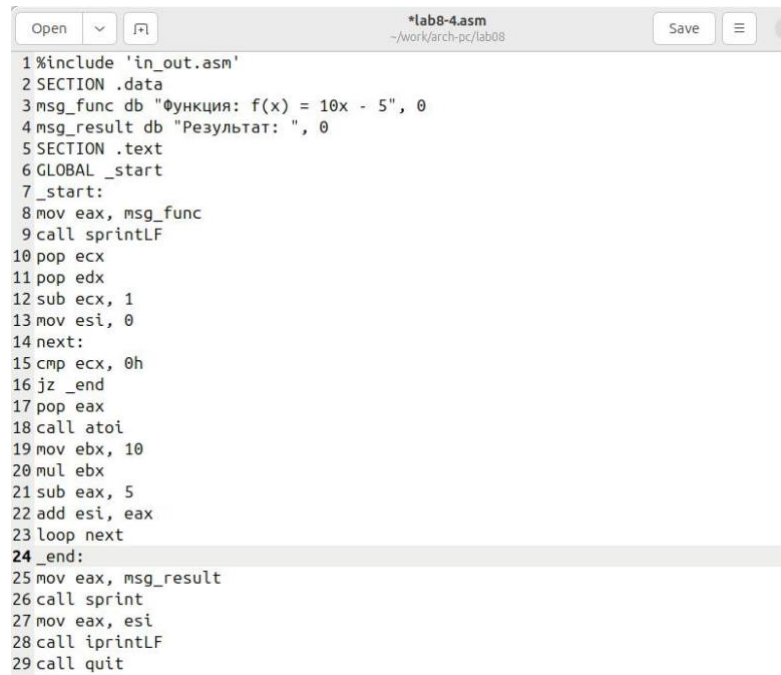
mamartynova@VirtualBox:~/work/arch-pc/lab08$ gedit lab8-3.asm
mamartynova@VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
mamartynova@VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
mamartynova@VirtualBox:~/work/arch-pc/lab08$ ./lab8-3 111 1 6
Результат: 666
mamartynova@VirtualBox:~/work/arch-pc/lab08$

```

Рис. 12: Запуск измененной третьей программы

2.3 Задания для самостоятельной работы

Пишу программу, которая будет находить сумма значений для функции $f(x) = 10x - 5$, которая совпадает с третьим вариантом (рис. 14).



```

1 %include 'in_out.asm'
2 SECTION .data
3 msg_func db "Функция: f(x) = 10x - 5", 0
4 msg_result db "Результат: ", 0
5 SECTION .text
6 GLOBAL _start
7 _start:
8 mov eax, msg_func
9 call sprintf
10 pop ecx
11 pop edx
12 sub ecx, 1
13 mov esi, 0
14 next:
15 cmp ecx, 0h
16 jz _end
17 pop eax
18 call atoi
19 mov ebx, 10
20 mul ebx
21 sub eax, 5
22 add esi, eax
23 loop next
24 _end:
25 mov eax, msg_result
26 call sprintf
27 mov eax, esi
28 call iprintf
29 call quit

```

Рис. 14: Написание программы для самостоятельной работы

Проверяю работу программы, указав в качестве аргумента несколько чисел (рис. 15).

```

mamartynova@VirtualBox:~/work/arch-pc/lab08$ touch lab8-4.asm
mamartynova@VirtualBox:~/work/arch-pc/lab08$ gedit lab8-4.asm
mamartynova@VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-4.asm
mamartynova@VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-4 lab8-4.o
mamartynova@VirtualBox:~/work/arch-pc/lab08$ ./lab8-4 1 2 3 4
Функция: f(x) = 10x - 5
Результат: 80
mamartynova@VirtualBox:~/work/arch-pc/lab08$

```

Рис. 15: Запуск программы для самостоятельной работы

3. Вывод

В ходе выполнения лабораторной работы №8 я получила навыки написания программ с использованием циклов, а также научилась обрабатывать аргументы командной строки.

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).