

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 10**

дисциплина: Архитектура компьютера

Студент: Нурмухаметова Каролина Марселеевна

Группа: НКАбд-05-25

**МОСКВА**

2025 г.

# *Оглавление*

1 Цель работы.....	3
2 Порядок выполнения лабораторной работы.....	3
3 Задания для самостоятельной работы.....	4
4 Выводы.....	6
5 Список литературы.....	6

## 1 Цель работы

Приобретение навыков написания программ для работы с файлами.

## 2 Выполнение лабораторной работы

Создала каталог для программ лабораторной работы № 10, перешла в него и создала файлы lab10-1.asm, readme-1.txt и readme-2.txt:

```
kmnurmukhametov@dk3n55 ~ $ mkdir ~/work/arch-pc/lab10
kmnurmukhametov@dk3n55 ~ $ cd ~/work/arch-pc/lab10
kmnurmukhametov@dk3n55 ~/work/arch-pc/lab10 $ touch lab10-1.asm readme-1.txt readme-2.txt
kmnurmukhametov@dk3n55 ~/work/arch-pc/lab10 $
```

Рис. 2.1 — Создание каталога

Ввела в файл lab10-1.asm текст программы из листинга 10.1. Создала исполняемый файл и проверила его работу.

```
%include 'in_out.asm'
SECTION .data
filename db 'readme.txt', 0h ; Имя файла
msg db 'Введите строку для записи в файл: ', 0h ; Сообщение
SECTION .bss
contents resb 255 ; переменная для вводимой строки
SECTION .text
global _start
_start:
; --- Печать сообщения `msg`
mov eax, msg
call sprint
; ---- Запись введенной с клавиатуры строки в `contents`
mov ecx, contents
mov edx, 255
call sread
; --- Открытие существующего файла (`sys_open`)
mov ecx, 2 ; открываем для записи (2)
mov ebx, filename
mov eax, 5
int 80h
; --- Запись дескриптора файла в `esi`
mov esi, eax
; --- Расчет длины введенной строки
mov eax, contents ; в `eax` запишется количество
call slen ; введенных байтов
; --- Записываем в файл `contents` (`sys_write`)
mov edx, eax
mov ecx, contents
mov ebx, esi
mov eax, 4
int 80h
; --- Закрываем файл (`sys_close`)
mov ebx, esi
mov eax, 6
int 80h
call quit
```

Рис. 2.2 — Текст программы lab10-1.asm

```
~/work/arch-pc/lab10 $ nasm -f elf lab10-1.asm
~/work/arch-pc/lab10 $ ld -m elf_i386 -o lab10-1 lab10-1.o
~/work/arch-pc/lab10 $ ./lab10-1
```

Введите строку для записи в файл: Hello RUDN!

```
~/work/arch-pc/lab10 $
```

Рис. 2.3 — Вывод программы lab10-1.asm

С помощью команды chmod изменила права доступа к исполняемому файлу lab10-1, запретив его выполнение. Попыталась выполнить файл.

```
kmnurmukhametov@dk3n55 ~/work/arch-pc/lab10 $ ls -l lab10-1
-rwxr-xr-x 1 kmnurmukhametov studsci 9768 дек 12 18:31 lab10-1
kmnurmukhametov@dk3n55 ~/work/arch-pc/lab10 $ chmod a-x lab10-1
kmnurmukhametov@dk3n55 ~/work/arch-pc/lab10 $ ls -l lab10-1
-rw-r--r-- 1 kmnurmukhametov studsci 9768 дек 12 18:31 lab10-1
kmnurmukhametov@dk3n55 ~/work/arch-pc/lab10 $ ./lab10-1
bash: ./lab10-1: Отказано в доступе
kmnurmukhametov@dk3n55 ~/work/arch-pc/lab10 $
```

Рис. 2.4 — Изменение прав программы lab10-1.asm

При попытке выполнить файл после запрета доступа система выдаёт ошибку "Отказано в доступе". Это происходит потому, что ядро операционной системы проверяет права доступа перед запуском любой программы. Поскольку я убрала бит выполнения (x), система рассматривает файл как обычные данные, а не как программу.

С помощью команды chmod изменила права доступа к файлу lab10-1.asm с исходным текстом программы, добавив права на исполнение. Попыталась выполнить его.

```
kmnurmukhametov@dk3n55 ~/work/arch-pc/lab10 $ chmod a+x lab10-1
kmnurmukhametov@dk3n55 ~/work/arch-pc/lab10 $ ./lab10-1
Введите строку для записи в файл: Hello world!
kmnurmukhametov@dk3n55 ~/work/arch-pc/lab10 $
```

Рис. 2.5 — Изменение прав программы lab10-1.asm

Команда chmod a+x установила права на выполнение для всех пользователей, что разрешило операционной системе запустить программу. После запуска программа начала работать в интерактивном режиме, запросив ввод строки для записи в файл, как и было задумано в её алгоритме.

В соответствии с вариантом в таблице 10.4 предоставила права доступа к файлу readme-1.txt представленные в символьном виде, а для файла readme-2.txt – в двоичном виде. После проверила правильность выполнения с помощью команды ls -l.

```
kmnurmukhametov@dk3n55 ~/work/arch-pc/lab10 $ chmod u=x,g=rwx,o=w readme-1.txt
kmnurmukhametov@dk3n55 ~/work/arch-pc/lab10 $ chmod 125 readme-2.txt
kmnurmukhametov@dk3n55 ~/work/arch-pc/lab10 $ ls -l readme-1.txt readme-2.txt
---rwx-w- 1 kmnurmukhametov studsci 0 дек 12 18:23 readme-1.txt
---x-w-r-x 1 kmnurmukhametov studsci 0 дек 12 18:23 readme-2.txt
kmnurmukhametov@dk3n55 ~/work/arch-pc/lab10 $
```

Рис. 2.6 — Выполнение программы

### 3 Задание для самостоятельной работы

Написала программу работающую по следующему алгоритму:

- Вывод приглашения “Как Вас зовут?”
- ввести с клавиатуры свои фамилию и имя
- создать файл с именем name.txt
- записать в файл сообщение “Меня зовут”
- дописать в файл строку введенную с клавиатуры
- закрыть файл

Создала исполняемый файл и проверила его работу.

```

1 %include 'in_out.asm'
2
3 SECTION .data
4   prompt db 'Как Вас зовут? ', 0h
5   message db 'Меня зовут ', 0h
6   filename db 'name.txt', 0h
7
8 SECTION .bss
9   name resb 100
10
11 SECTION .text
12   global _start
13
14 _start:
15   mov eax, prompt
16   call sprint
17
18   mov ecx, name
19   mov edx, 100
20   call sread
21
22   mov ecx, 0x241
23   mov ebx, filename
24   mov eax, 5
25   int 80h
26
27   cmp eax, -1
28   jle error
29
30   mov esi, eax
31
32   mov eax, message
33   call slen
34
35   mov edx, eax
36   mov ecx, message
37   mov ebx, esi
38   mov eax, 4
39   int 80h
40
41   mov eax, name
42   call slen
43
44   mov edx, eax
45   mov ecx, name
46   mov ebx, esi
47   mov eax, 4
48   int 80h
49
50   mov ebx, esi
51   mov eax, 6
52   int 80h
53
54   call quit
55
56 error:
57   mov eax, 'O'
58   call sprint
59   mov eax, 'ш'
60   call sprint
61   mov eax, 'и'
62   call sprint
63   mov eax, '6'
64   call sprint
65   mov eax, 'к'
66   call sprint
67   mov eax, 'а'
68   call sprint
69   mov eax, ''
70   call sprint
71   mov eax, 'с'
72   call sprint
73   mov eax, 'о'
74   call sprint
75   mov eax, 'з'
76   call sprint
77   mov eax, 'д'
78   call sprint
79   mov eax, 'а'
80   call sprint
81   mov eax, 'н'
82   call sprint
83   mov eax, 'и'
84   call sprint
85   mov eax, 'я'
86   call sprint
87   mov eax, ''
88   call sprint
89   mov eax, 'ф'
90   call sprint
91   mov eax, 'а'
92   call sprint
93   mov eax, 'й'
94   call sprint
95   mov eax, 'л'
96   call sprint
97   mov eax, 'а'
98   call sprint

```

Рис. 3.1 — Текст самостоятельного задания

```

kmnurmukhametov@dk3n55 ~/work/arch-pc/lab10 $ touch sam10.asm
kmnurmukhametov@dk3n55 ~/work/arch-pc/lab10 $ nasm -f elf sam10.asm
kmnurmukhametov@dk3n55 ~/work/arch-pc/lab10 $ ld -m elf_i386 -o sam10 sam10.o
kmnurmukhametov@dk3n55 ~/work/arch-pc/lab10 $ ./sam10
Как Вас зовут? Nurmukhametova Karolina
kmnurmukhametov@dk3n55 ~/work/arch-pc/lab10 $ █

```

Рис. 3.2 — Результат выполнения самостоятельного задания

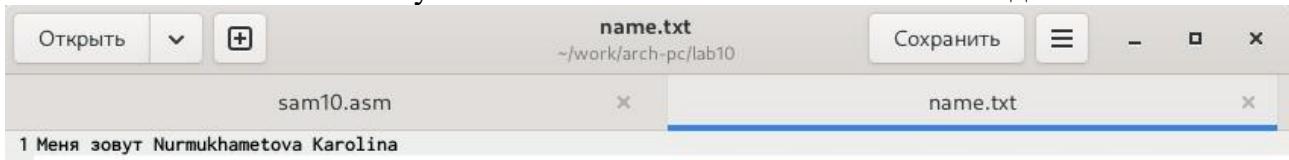


Рис. 3.3 — Проверка

## 4 Выводы

В результате выполнения данной лабораторной работы я приобрела навыки написания программ для работы с файлами.

## 5 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 c. — (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 c. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 c. — ISBN 9781784396879.
9. Калдаев 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 c. — ISBN 978-5-94157-538-1.
14. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : MAKС Пресс, 2011. — URL: [http://www.stolyarov.info/books/asm\\_unix](http://www.stolyarov.info/books/asm_unix).
15. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 c. — (Классика Computer Science).
16. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 c. — (Классика Computer Science).