

# **Отчет по лабораторной работе 11**

**Работа с файлами средствами Nasm**

Гисматуллин Артём Вадимович НПИбд-01-22

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
3.1	Права доступа к файлам . . . . .	7
3.2	Работа с файлами средствами Nasm . . . . .	8
3.2.1	Открытие и создание файла . . . . .	8
3.2.2	Запись в файл . . . . .	9
3.2.3	Чтение файла . . . . .	9
3.2.4	Закрытие файла . . . . .	9
3.2.5	Изменение содержимого файла . . . . .	10
3.2.6	Удаление файла . . . . .	10
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>11</b>
<b>5</b>	<b>Выполнение задания для самостоятельной работы</b>	<b>17</b>
<b>6</b>	<b>Выводы</b>	<b>21</b>
	<b>Список литературы</b>	<b>22</b>

## Список иллюстраций

4.1	Командная строка. каталог ~/work/arch-pc/lab11 . . . . .	12
4.2	Командная строка. Выполнение программы lab11-1.asm . . . . .	13
4.3	Командная строка. Изменение прав доступа lab11-1 . . . . .	14
4.4	Командная строка. Изменение прав доступа lab11-1.asm . . . . .	15
4.5	Командная строка. Изменение прав доступа readme.txt . . . . .	16
5.1	Командная строка. Выполнение программы lab11-2.asm . . . . .	18
5.2	Командная строка. Содержимое программы lab11-2.asm . . . . .	19
5.3	Командная строка. Содержимое программы lab11-2.asm 2 . . . . .	20

## **Список таблиц**

# **1 Цель работы**

Приобретение навыков написания программ для работы с файлами.

## 2 Задание

1. Ввести в файл `lab11-1.asm` текст программы из листинга 11.1. Проверить его работу.
2. С помощью команды `chmod` изменить права доступа к `lab11-1`, запретив его исполнение. Попробовать выполнить файл.
3. С помощью команды `chmod` изменить права доступа, добавив права к исполнению, к `lab11-1.asm`. Попробовать выполнить файл и объяснить результаты.
4. Изменить права доступа к файлу `readme.txt` в соответствии с вариантом 11.
5. Написать программу, работающую по определенному алгоритму.

## 3 Теоретическое введение

### 3.1 Права доступа к файлам

ОС GNU/Linux является многопользовательской операционной системой. И для обеспечения защиты данных одного пользователя от действий других пользователей существуют специальные механизмы разграничения доступа к файлам. Кроме ограничения доступа, данный механизм позволяет разрешить другим пользователям доступ данным для совместной работы.

Права доступа определяют набор действий (чтение, запись, выполнение), разрешённых для выполнения пользователям системы над файлами. Для каждого файла пользователь может входить в одну из трех групп: владелец, член группы владельца, все остальные. Для каждой из этих групп может быть установлен свой набор прав доступа. Владелцем файла является его создатель.

Набор прав доступа задается тройками битов и состоит из прав на чтение, запись и исполнение файла. В символьном представлении он имеет вид строк `gwx`, где вместо любого символа может стоять дефис. Всего возможно 8 комбинаций. Буква означает наличие права (установлен в единицу второй бит триады `r` — чтение, первый бит `w` — запись, нулевой бит `x` — исполнение), а дефис означает отсутствие права (нулевое значение соответствующего бита). Также права доступа могут быть представлены как восьмеричное число. Так, права доступа `rw` (чтение и запись, без исполнения) понимаются как три двоичные цифры `110` или как восьмеричная цифра `6`.

Для изменения прав доступа служит команда `chmod`, которая понимает как

символьное, так и числовое указание прав.

## 3.2 Работа с файлами средствами Nasm

В операционной системе Linux существуют различные методы управления файлами, например, такие как создание и открытие файла, только для чтения или для чтения и записи, добавления в существующий файл, закрытия и удаления файла, предоставление прав доступа.

Обработка файлов в операционной системе Linux осуществляется за счет использования определенных системных вызовов. Для корректной работы и доступа к файлу при его открытии или создании, файлу присваивается уникальный номер (16-битное целое число) – дескриптор файла.

Общий алгоритм работы с системными вызовами в Nasm можно представить в следующем виде:

1. Поместить номер системного вызова в регистр EAX;
2. Поместить аргументы системного вызова в регистрах EBX, ECX и EDX;
3. Вызов прерывания (int 80h);
4. Результат обычно возвращается в регистр EAX.

### 3.2.1 Открытие и создание файла

Для создания и открытия файла служит системный вызов `sys_creat`, который использует следующие аргументы: права доступа к файлу в регистре ECX, имя файла в EBX и номер системного вызова `sys_creat` (8) в EAX.

Для открытия существующего файла служит системный вызов `sys_open`, который использует следующие аргументы: права доступа к файлу в регистре EDX, режим доступа к файлу в регистр ECX, имя файла в EBX и номер системного вызова `sys_open` (5) в EAX.



### **3.2.2 Запись в файл**

Для записи в файл служит системный вызов `sys_write`, который использует следующие аргументы: количество байтов для записи в регистре `EDX`, строку содержимого для записи `ECX`, файловый дескриптор в `EBX` и номер системного вызова `sys_write` (4) в `EAX`.

Системный вызов возвращает фактическое количество записанных байтов в регистр `EAX`. В случае ошибки, код ошибки также будет находиться в регистре `EAX`.

Прежде чем записывать в файл, его необходимо создать или открыть, что позволит получить дескриптор файла.

### **3.2.3 Чтение файла**

Для чтения данных из файла служит системный вызов `sys_read`, который использует следующие аргументы: количество байтов для чтения в регистре `EDX`, адрес в памяти для записи прочитанных данных в `ECX`, файловый дескриптор в `EBX` и номер системного вызова `sys_read` (3) в `EAX`. Как и для записи, прежде чем читать из файла, его необходимо открыть, что позволит получить дескриптор файла.

### **3.2.4 Закрытие файла**

Для правильного закрытия файла служит системный вызов `sys_close`, который использует один аргумент – дескриптор файла в регистре `EBX`. После вызова ядра происходит удаление дескриптора файла, а в случае ошибки, системный вызов возвращает код ошибки в регистр `EAX`.

### **3.2.5 Изменение содержимого файла**

Для изменения содержимого файла служит системный вызов `sys_lseek`, который использует следующие аргументы: исходная позиция для смещения `EDX`, значение смещения в байтах в `ECX`, файловый дескриптор в `EBX` и номер системного вызова `sys_lseek` (19) в `EAX`.

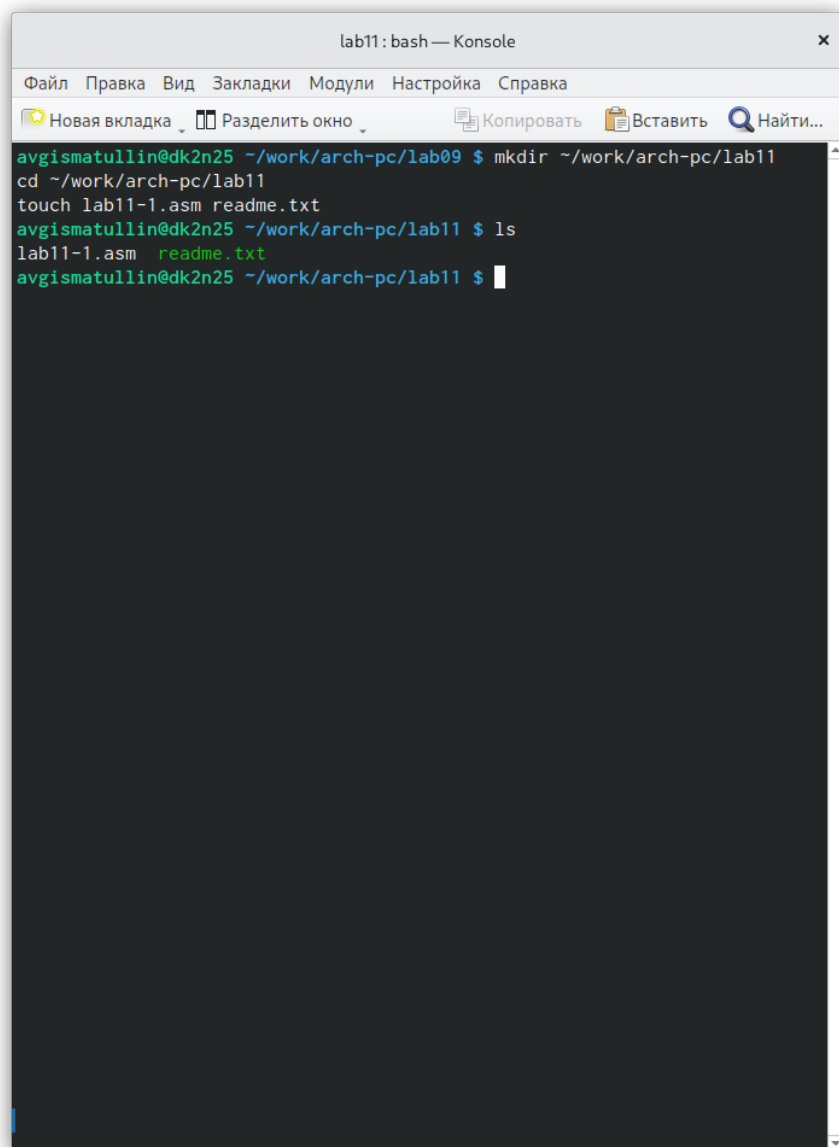
Значение смещения можно задавать в байтах.

### **3.2.6 Удаление файла**

Удаление файла осуществляется системным вызовом `sys_unlink`, который использует один аргумент – имя файла в регистре `EBX`.

## **4 Выполнение лабораторной работы**

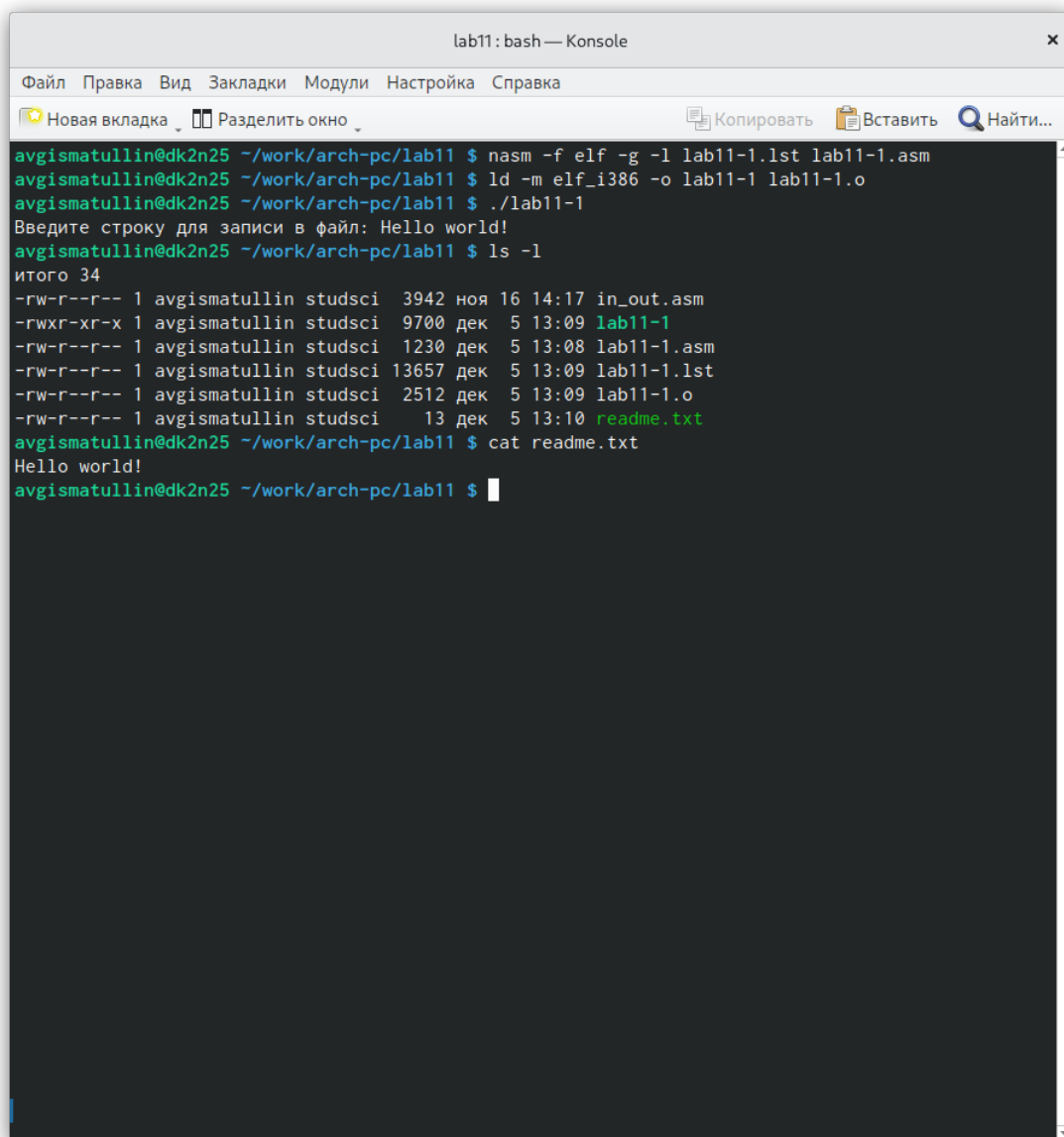
1. Создаем каталог для программ лабораторной работы No 11, переходим в него и создаем файл lab11-1.asm. (рис. 4.1)



```
lab11: bash — Konsole
Файл  Правка  Вид  Закладки  Модули  Настройка  Справка
Новая вкладка  Разделить окно  Копировать  Вставить  Найти...
avgismatullin@dk2n25 ~/work/arch-pc/lab09 $ mkdir ~/work/arch-pc/lab11
cd ~/work/arch-pc/lab11
touch lab11-1.asm readme.txt
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $ ls
lab11-1.asm  readme.txt
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $
```

Рис. 4.1: Командная строка. каталог ~/work/arch-pc/lab11

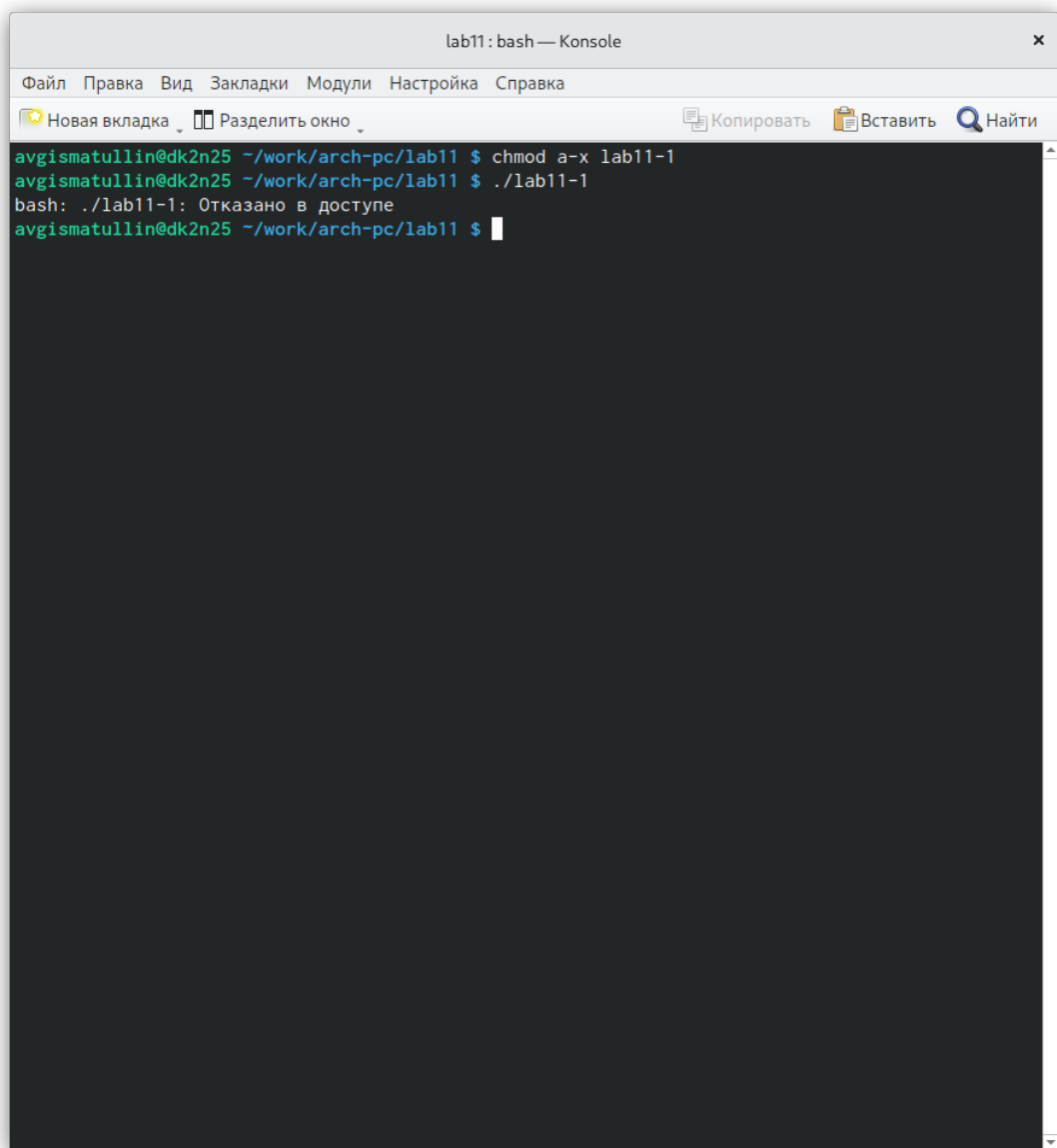
2. Ввели в файл lab11-1.asm текст программы листинга, создали исполняемый файл и посмотрели результат программы. (рис. 4.2)



```
lab11: bash — Konsole
Файл  Правка  Вид  Закладки  Модули  Настройка  Справка
Новая вкладка  Разделить окно  Копировать  Вставить  Найти...
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $ nasm -f elf -g -l lab11-1.lst lab11-1.asm
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $ ld -m elf_i386 -o lab11-1 lab11-1.o
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $ ./lab11-1
Введите строку для записи в файл: Hello world!
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $ ls -l
итого 34
-rw-r--r-- 1 avgismatullin studsci 3942 ноя 16 14:17 in_out.asm
-rwxr-xr-x 1 avgismatullin studsci 9700 дек 5 13:09 lab11-1
-rw-r--r-- 1 avgismatullin studsci 1230 дек 5 13:08 lab11-1.asm
-rw-r--r-- 1 avgismatullin studsci 13657 дек 5 13:09 lab11-1.lst
-rw-r--r-- 1 avgismatullin studsci 2512 дек 5 13:09 lab11-1.o
-rw-r--r-- 1 avgismatullin studsci 13 дек 5 13:10 readme.txt
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $ cat readme.txt
Hello world!
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $
```

Рис. 4.2: Командная строка. Выполнение программы lab11-1.asm

3. С помощью команды `chmod` изменили права доступа к исполняемому файлу lab11-1. (рис. 4.3)

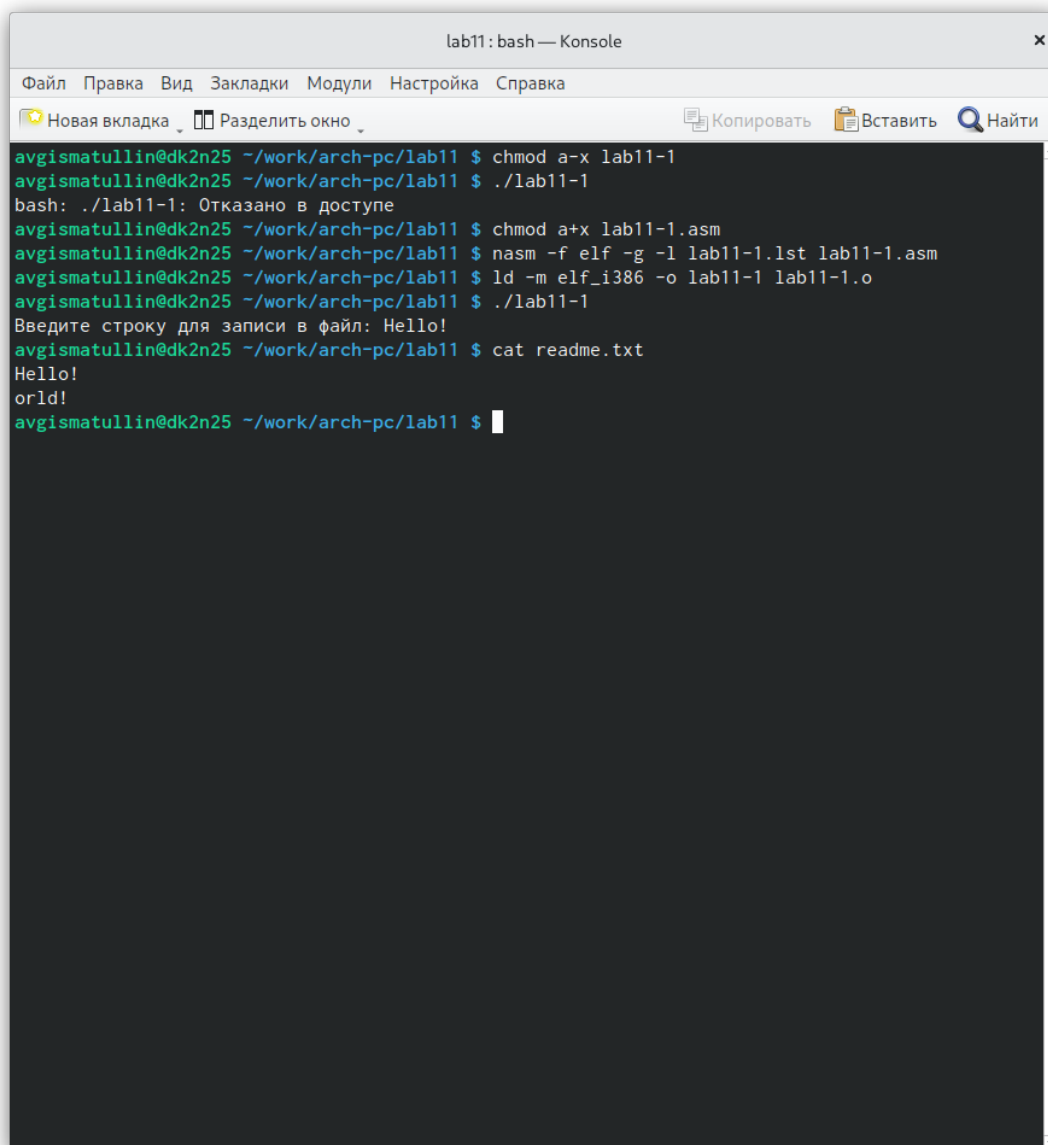


```
lab11: bash — Konsole
Файл Правка Вид Закладки Модули Настройка Справка
Новая вкладка Разделить окно Копировать Вставить Найти
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $ chmod a-x lab11-1
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $ ./lab11-1
bash: ./lab11-1: Отказано в доступе
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $
```

Рис. 4.3: Командная строка. Изменение прав доступа lab11-1

Так как мы запретили исполнение файла, при попытке это осуществить в ответ получаем сообщение о том, что нам отказано в доступе.

4. С помощью команды `chmod` изменяем права доступа к файлу `lab11-1.asm`, но уже наделяем его правами на исполнение. (рис. 4.4)

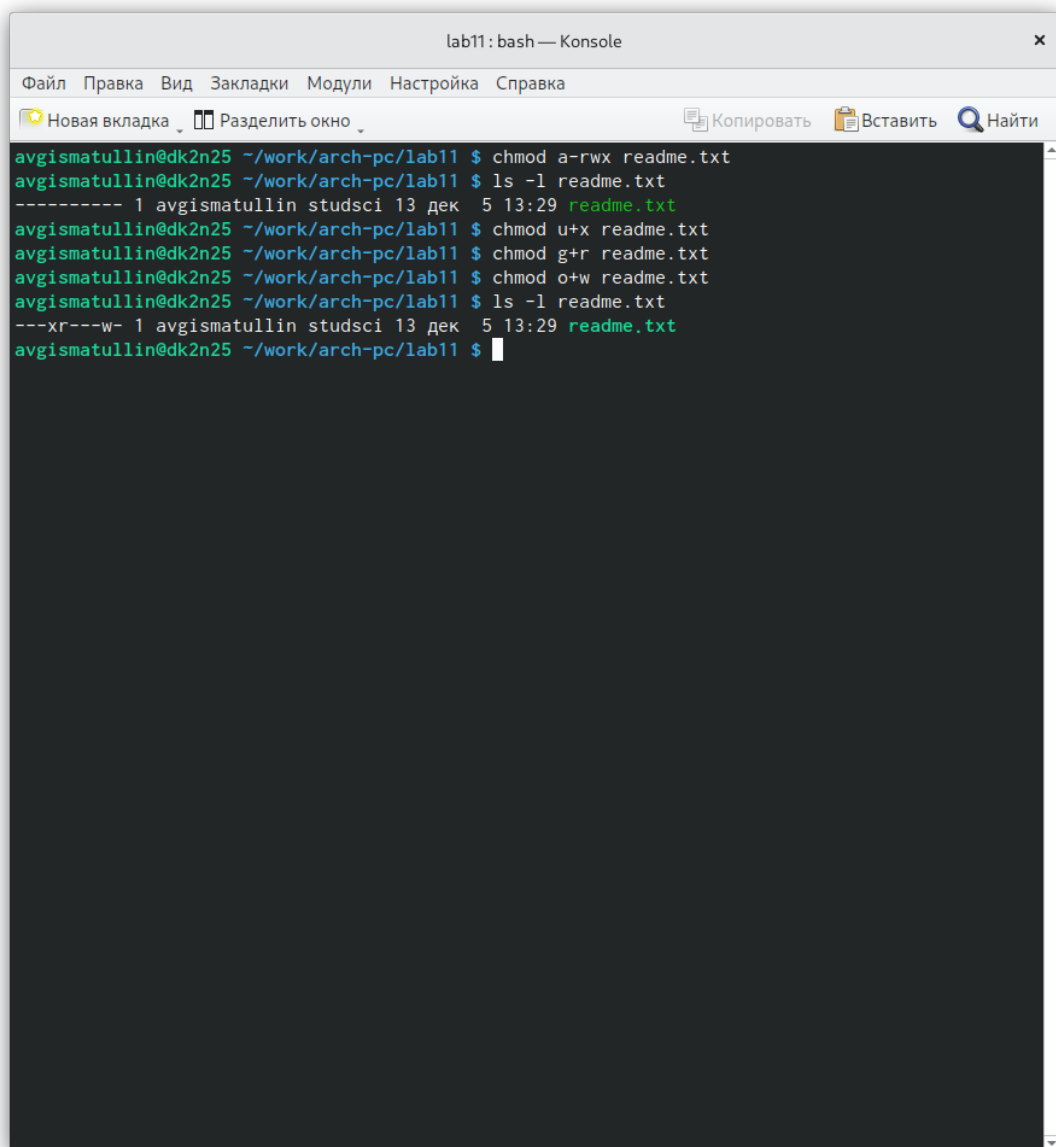


```
lab11: bash — Konsole
Файл Правка Вид Закладки Модули Настройка Справка
Новая вкладка Разделить окно Копировать Вставить Найти
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $ chmod a-x lab11-1
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $ ./lab11-1
bash: ./lab11-1: Отказано в доступе
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $ chmod a+x lab11-1.asm
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $ nasm -f elf -g -l lab11-1.lst lab11-1.asm
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $ ld -m elf_i386 -o lab11-1 lab11-1.o
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $ ./lab11-1
Введите строку для записи в файл: Hello!
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $ cat readme.txt
Hello!
orld!
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $
```

Рис. 4.4: Командная строка. Изменение прав доступа lab11-1.asm

В результате после создания исполняемого файла мы смогли успешно его запустить, так как файл с программой изначально был наделен правами на исполнение.

5. Далее мы предоставляем права доступа к файлу readme.txt в соответствии с вариантом 11. Проверяем корректность командой `ls -l readme.txt`. (рис. 4.5)



```
lab11: bash — Konsole
Файл Правка Вид Закладки Модули Настройка Справка
Новая вкладка Разделить окно Копировать Вставить Найти
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $ chmod a-rwx README.txt
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $ ls -l README.txt
----- 1 avgismatullin studsci 13 дек  5 13:29 README.txt
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $ chmod u+x README.txt
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $ chmod g+r README.txt
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $ chmod o+w README.txt
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $ ls -l README.txt
---xr---w- 1 avgismatullin studsci 13 дек  5 13:29 README.txt
avgismatullin@dk2n25 ~/work/arch-pc/lab11 $
```

Рис. 4.5: Командная строка. Изменение прав доступа README.txt

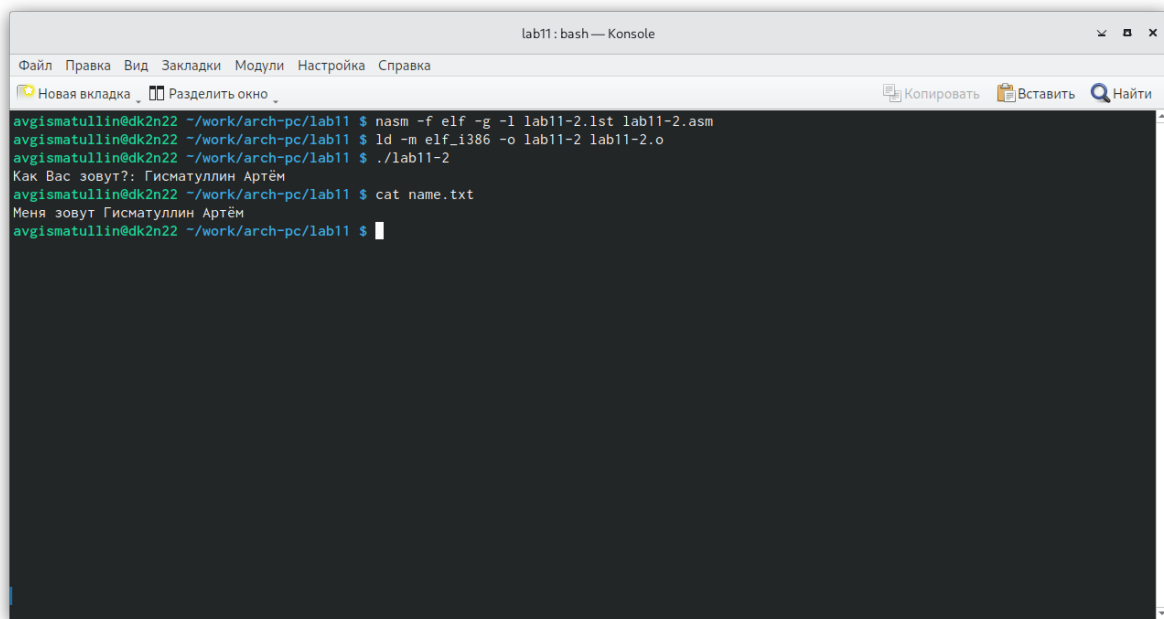


## 5 Выполнение задания для самостоятельной работы

Суть задания для самостоятельной работы заключается в том, чтобы написать 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

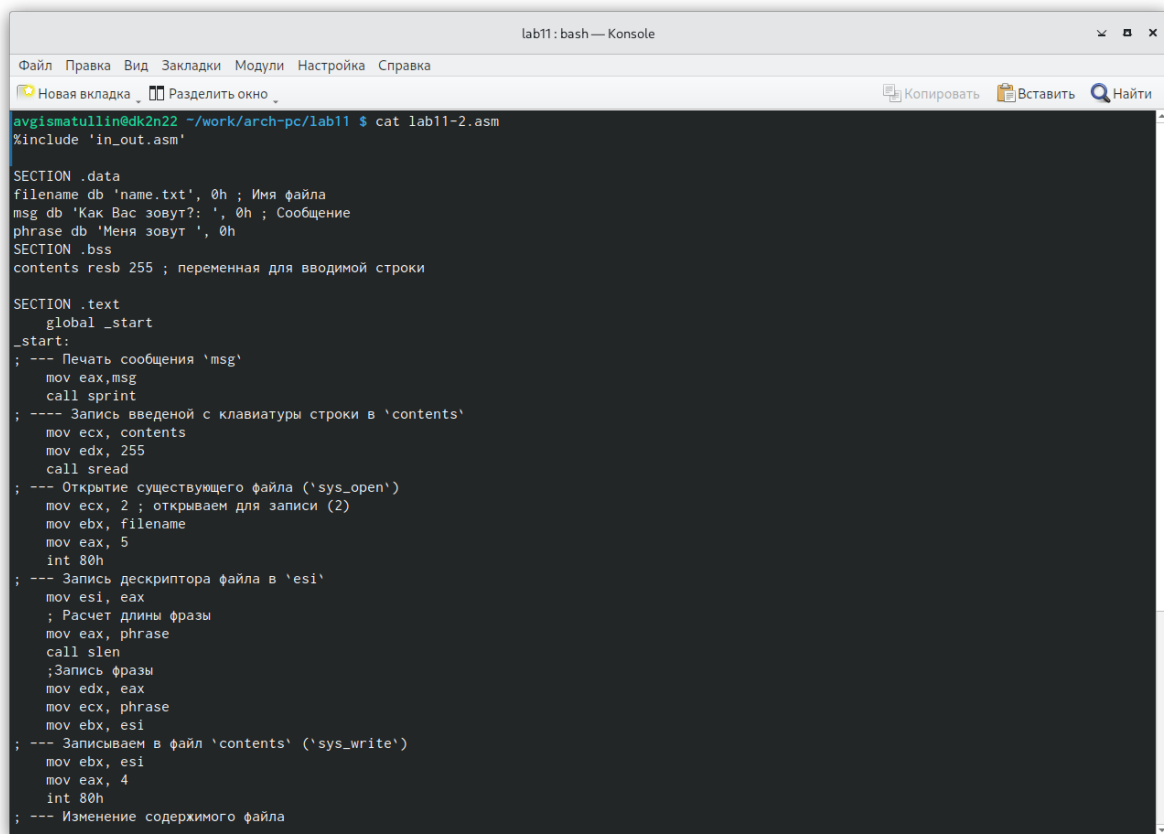
Напишем эту программу в файле lab11-2.asm. Результат получился следующий:  
(рис. 5.1)



```
lab11: bash — Konsole
Файл  Правка  Вид  Закладки  Модули  Настройка  Справка
Новая вкладка  Разделить окно  Копировать  Вставить  Найти
avgismatullin@dk2n22 ~/work/arch-pc/lab11 $ nasm -f elf -g -l lab11-2.lst lab11-2.asm
avgismatullin@dk2n22 ~/work/arch-pc/lab11 $ ld -m elf_i386 -o lab11-2 lab11-2.o
avgismatullin@dk2n22 ~/work/arch-pc/lab11 $ ./lab11-2
Как Вас зовут?: Гисматуллин Артём
avgismatullin@dk2n22 ~/work/arch-pc/lab11 $ cat name.txt
Меня зовут Гисматуллин Артём
avgismatullin@dk2n22 ~/work/arch-pc/lab11 $
```

Рис. 5.1: Командная строка. Выполнение программы lab11-2.asm

Идея заключалась в том, что я сперва записал фразу “Меня зовут”, а затем открыл файл для изменения содержимого файла. В результате удалось вывести сообщение целиком. Реализация следующая: (рис. 5.2), (рис. 5.3)

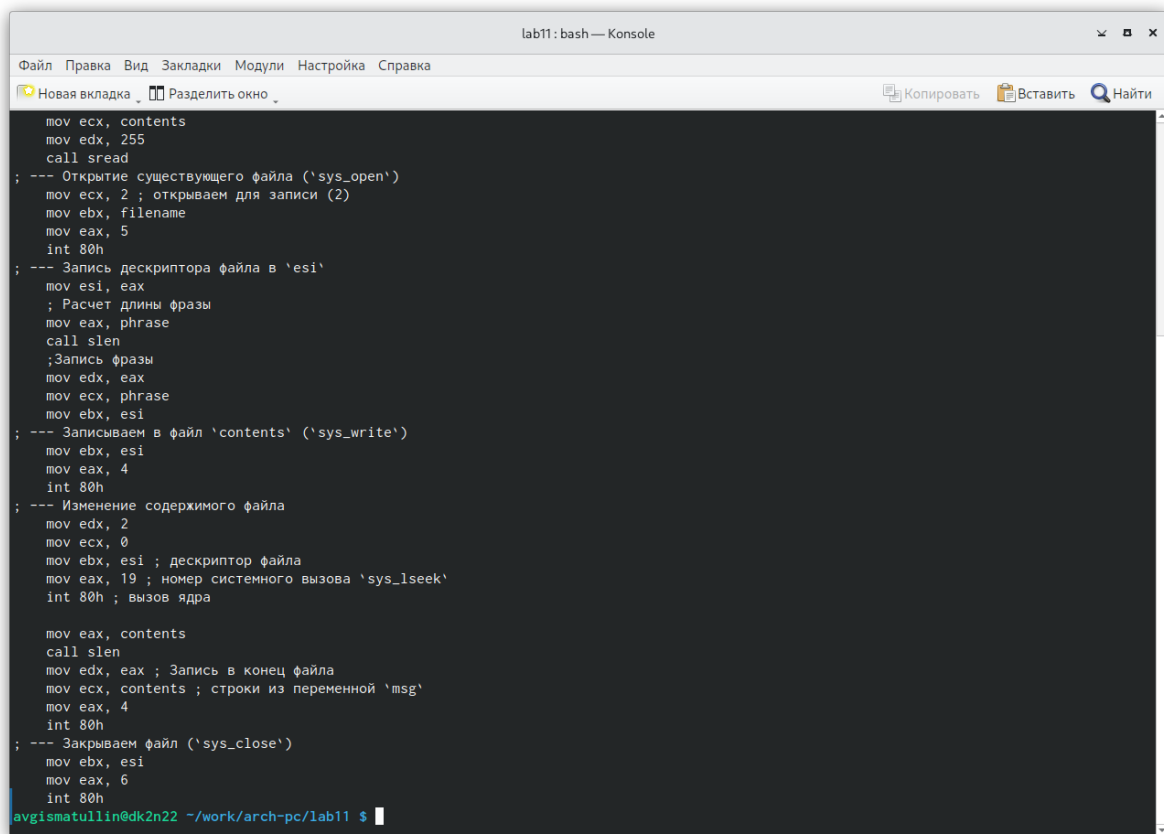


```
lab11: bash — Konsole
Файл  Правка  Вид  Закладки  Модули  Настройка  Справка
Новая вкладка  Разделить окно  Копировать  Вставить  Найти
avgismatullin@dk2n22 ~/work/arch-pc/lab11 $ cat lab11-2.asm
#include 'in_out.asm'

SECTION .data
filename db 'name.txt', 0h ; Имя файла
msg db 'Как Вас зовут?: ', 0h ; Сообщение
phrase db 'Меня зовут ', 0h
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 фразы
    mov eax, phrase
    call slen
    ;Запись фразы
    mov edx, eax
    mov ecx, phrase
    mov ebx, esi
; --- Записываем в файл 'contents' ('sys_write')
    mov ebx, esi
    mov eax, 4
    int 80h
; --- Изменение содержимого файла
```

Рис. 5.2: Командная строка. Содержимое программы lab11-2.asm



```
lab11: bash — Konsole
Файл  Правка  Вид  Закладки  Модули  Настройка  Справка
Новая вкладка  Разделить окно  Копировать  Вставить  Найти

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 фразы
mov eax, phrase
call slen
; Запись фразы
mov edx, eax
mov ecx, phrase
mov ebx, esi
; --- Записываем в файл 'contents' ('sys_write')
mov ebx, esi
mov eax, 4
int 80h
; --- Изменение содержимого файла
mov edx, 2
mov ecx, 0
mov ebx, esi ; дескриптор файла
mov eax, 19 ; номер системного вызова 'sys_lseek'
int 80h ; вызов ядра

mov eax, contents
call slen
mov edx, eax ; Запись в конец файла
mov ecx, contents ; строки из переменной 'msg'
mov eax, 4
int 80h
; --- Закрываем файл ('sys_close')
mov ebx, esi
mov eax, 6
int 80h
avgismatullin@dk2n22 ~/work/arch-pc/lab11 $
```

Рис. 5.3: Командная строка. Содержимое программы lab11-2.asm 2

## **6 Выводы**

В ходе работы я приобрел навыки написания программ для работы с файлами.

## **Список литературы**