

# **Отчёт по лабораторной работе №10**

**Дисциплина: Архитектура компьютеров**

Лазарев Даниил Михайлович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Теоретическое введение</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>4</b>	<b>Выполнение самостоятельной работы</b>	<b>9</b>
<b>5</b>	<b>Выводы</b>	<b>12</b>

## Список иллюстраций

3.1	Создание файла в каталоге . . . . .	6
3.2	Код программы в файле . . . . .	7
3.3	Преобразование в исполняемый файл . . . . .	7
3.4	Изменение прав доступа и результат . . . . .	8
3.5	Изменение прав доступа и результат . . . . .	8
3.6	Изменение прав доступа и проверка . . . . .	8
4.1	Создание файла . . . . .	9
4.2	Код программы . . . . .	10
4.3	Преобразование файла . . . . .	11
4.4	Содержимое файла . . . . .	11

# **1 Цель работы**

Приобретение навыков написания программ для работы с файлами.

## 2 Теоретическое введение

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. 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, приведенных в таблице 10.1. 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`.

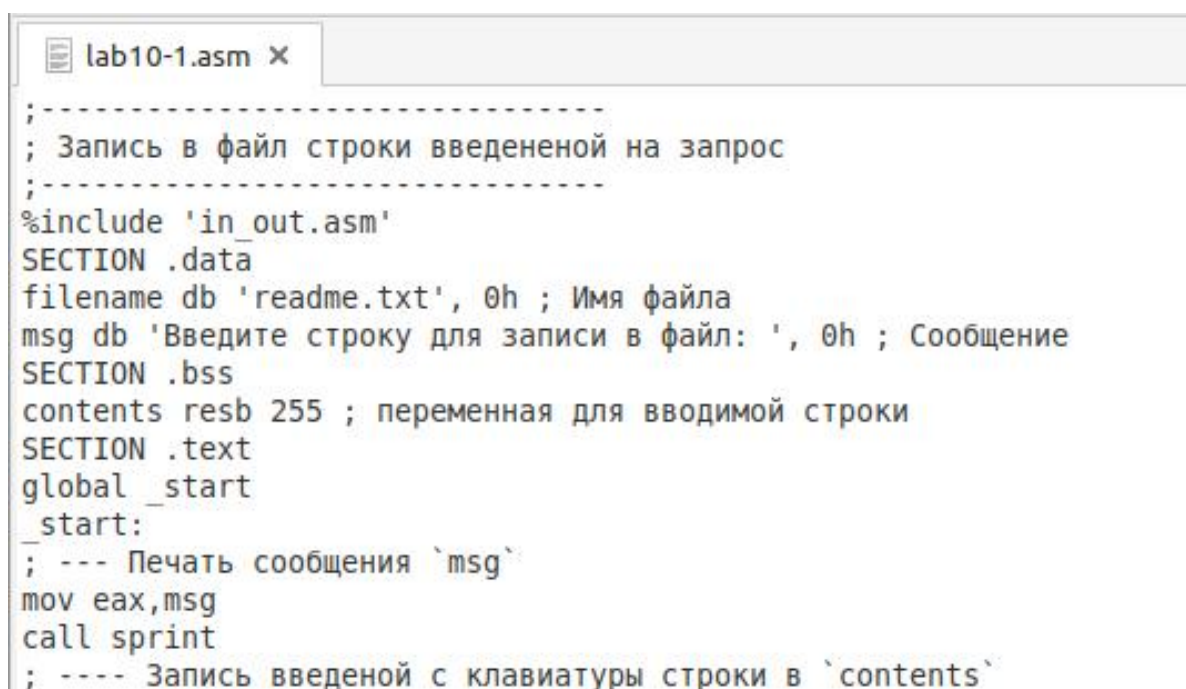
### 3 Выполнение лабораторной работы

Создадим каталог для программ лаб. работы н.10, перейдем в него и создадим файл “lab10-1.asm” (рис. 3.1)

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

Рис. 3.1: Создание файла в каталоге

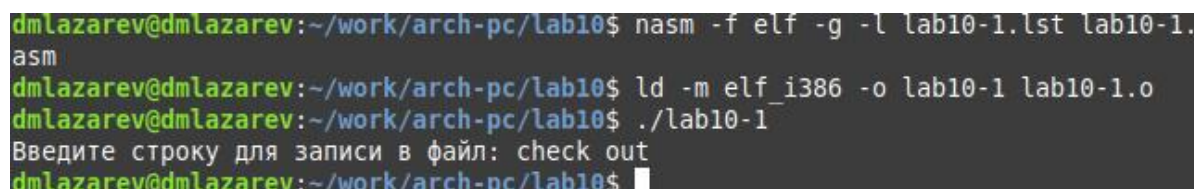
Введем в созданный файл текст программы из предложенного нам листинга 10.1 (рис. 3.2)



```
lab10-1.asm x
; -----
; Запись в файл строки введенной на запрос
; -----
%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`
```

Рис. 3.2: Код программы в файле

Создадим исполняемый файл и запустим его, предварительно скопировав из предыдущей лаб. работы файл “in\_out.asm” для корректной работы (рис. 3.3)



```
dmlazarev@dmlazarev:~/work/arch-pc/lab10$ nasm -f elf -g -l lab10-1.lst lab10-1.asm
dmlazarev@dmlazarev:~/work/arch-pc/lab10$ ld -m elf_i386 -o lab10-1 lab10-1.o
dmlazarev@dmlazarev:~/work/arch-pc/lab10$ ./lab10-1
Введите строку для записи в файл: check out
dmlazarev@dmlazarev:~/work/arch-pc/lab10$
```

Рис. 3.3: Преобразование в исполняемый файл

С помощью команды `chmod` изменим права доступа к созданному нами исполняемому файлу так, чтобы присутствовал запрет на его выполнение. При попытке выполнения получим отказ. Следовательно, команда `chmod` была введена с верными ключами. (рис. 3.5)

```

dmlazarev@dmlazarev:~/work/arch-pc/lab10$ chmod -x lab10-1
dmlazarev@dmlazarev:~/work/arch-pc/lab10$ ./lab10-1
bash: ./lab10-1: Permission denied
dmlazarev@dmlazarev:~/work/arch-pc/lab10$

```

Рис. 3.4: Изменение прав доступа и результат

С помощью команды `chmod` изменим права доступа к изначальному файлу “lab10-1.asm” с текстом листинга так, чтобы присутствовали права на его выполнение. Поскольку файл содержит в себе исходный код на языке ассемблера, то добавление права на исполнение не даст нам ожидаемого результата, а лишь выведет ошибку. Чтобы выполнить данный файл, необходимо предварительно его превратить в исполняемый. (рис. ??)

```

dmlazarev@dmlazarev:~/work/arch-pc/lab10$ chmod +x lab10-1.asm
dmlazarev@dmlazarev:~/work/arch-pc/lab10$ ./lab10-1.asm
./lab10-1.asm: line 1: syntax error near unexpected token `;'
./lab10-1.asm: line 1: `;-----'
dmlazarev@dmlazarev:~/work/arch-pc/lab10$

```

Рис. 3.5: Изменение прав доступа и результат

Далее, из таблицы 10.4 выберем 9 вариант и изменим права для файлов “readme-1.txt” и “readme-2.txt” в соответствии с ней. Проверим правильность выполнения. (рис. 3.6)

```

dmlazarev@dmlazarev:~/work/arch-pc/lab10$ chmod u=x,g=w,o=w readme-1.txt
dmlazarev@dmlazarev:~/work/arch-pc/lab10$ chmod 135 readme-2.txt
dmlazarev@dmlazarev:~/work/arch-pc/lab10$ ls -l
total 40
-rw-rw-r-- 1 dmlazarev dmlazarev 3942 ноя 11 18:32 in_out.asm
-rw-rw-r-- 1 dmlazarev dmlazarev 9700 дек 14 16:44 lab10-1
-rwxrwxr-x 1 dmlazarev dmlazarev 1287 дек 14 16:43 lab10-1.asm
-rw-rw-r-- 1 dmlazarev dmlazarev 13713 дек 14 16:44 lab10-1.lst
-rw-rw-r-- 1 dmlazarev dmlazarev 2512 дек 14 16:44 lab10-1.o
---x-w--w- 1 dmlazarev dmlazarev 0 дек 14 16:40 readme-1.txt
---x-wxr-x 1 dmlazarev dmlazarev 0 дек 14 16:40 readme-2.txt
dmlazarev@dmlazarev:~/work/arch-pc/lab10$

```

Рис. 3.6: Изменение прав доступа и проверка



## 4 Выполнение самостоятельной работы

Создадим файл “lab102.asm” для выполнения самостоятельной работы. (рис. 4.1)



```
dmlazarev@dmlazarev:~/work/arch-pc/lab10$ touch lab102.asm
dmlazarev@dmlazarev:~/work/arch-pc/lab10$
```

Рис. 4.1: Создание файла

Напишем код, работающий по алгоритму, запрошенному в гайде к лабораторной работе. (рис. 4.2)

```

SECTION .data
msg: DB 'Как вас зовут? ',0
file: DB 'name.txt',0
name: DB 'Меня зовут ',0

SECTION .bss
X: RESB 80

SECTION .text
GLOBAL _start
_start:
;Вывод приглашения "Как Вас зовут?"
    mov eax,msg
    call sprint
;ввести с клавиатуры свои фамилию и имя
    mov ecx,X
    mov edx,80
    call sread
;создать файл с именем name.txt
    mov ecx,0777o
    mov ebx,file
    mov eax,8
    int 80h
;записать в файл сообщение "Меня зовут"
    mov ecx, 2
    mov ebx, file
    mov eax, 5
    int 80h
    mov esi, eax
    mov eax, name
    call slen
    mov edx, eax
    mov ecx, name
    mov ebx, esi
    mov eax, 4
    int 80h
;дописать в файл строку введенную с клавиатуры
    mov eax,X
    call slen
    mov edx,eax
    mov ecx,X
    mov ebx,esi
    mov eax,4
    int 80h
;закрывать файл
    mov ebx,esi
    mov eax,6
    int 80h
    call quit

```

Рис. 4.2: Код программы

Преобразуем написанный нами файл в исполняемый и введем требуемую от нас информацию. (рис. 4.3)

```
dmlazarev@dmlazarev:~/work/arch-pc/lab10$ nasm -f elf lab102.asm
dmlazarev@dmlazarev:~/work/arch-pc/lab10$ ld -m elf_i386 -o lab102 lab102.o
dmlazarev@dmlazarev:~/work/arch-pc/lab10$ ./lab102
Как вас зовут? Лазарев Даниил
```

Рис. 4.3: Преобразование файла

Проверим правильность выполнения кода, просмотрев содержимое файла “name.txt” (рис. 4.4)

```
dmlazarev@dmlazarev:~/work/arch-pc/lab10$ ls
in out.asm  lab10-1.asm  lab10-1.o  lab102.asm  name.txt      readme-2.txt
lab10-1     lab10-1.lst  lab102     lab102.o   readme-1.txt
dmlazarev@dmlazarev:~/work/arch-pc/lab10$ cat name.txt
Меня зовут Лазарев Даниил
dmlazarev@dmlazarev:~/work/arch-pc/lab10$
```

Рис. 4.4: Содержимое файла

## **5 Выводы**

В ходе лабораторной работы мы приобрели навыки написания программ для работы с файлами.