

# 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: Бессонов Андрей Максимович

Группа: НКАбд - 01 - 25

МОСКВА

2025 г.

## **Содержание**

1. Цель работы	4
2. Теоретическое введение	4
3. Выполнение лабораторной работы	5
4. Выполнение самостоятельной работы	8
5. Выводы	10

## Список иллюстраций

Рис 3.1: 31	5
Рис 3.2: 32	6
Рис 3.3: 33	7
Рис 3.4: 34	7
Рис 3.5.1: 35.1	7
Рис 3.5.2: 35.2	8
Рис 3.6: 36	8
Рис 4.1: 41	9
Рис 4.2: 42	10

## 1. Цель работы

Приобретение навыков написания программ для работы с файлами.

## 2. Теоретическое введение

### 2.1. 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. Для предоставления прав доступа другому пользователю или другой группе командой:

```
chown [ключи] <новый_пользователь>[:новая_группа]<файл>
```

Набор прав доступа задается тройками битов и состоит из прав на чтение, запись и исполнение файла. В символьном представлении он имеет вид строк rwx, 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.

Тип файла определяется первой позицией, это может быть: каталог — d, обычный файл — дефис (-) или символьная ссылка на другой файл — l. Следующие 3 набора по 3 символа определяют конкретные права для конкретных групп: r — разрешено чтение файла, w — разрешена запись в файл; x — разрешено исполнение файла и дефис (-) — право не дано. Для изменения прав доступа служит команда `chmod`, которая понимает как символьное, так и числовое указание прав. Для того чтобы назначить файлу `/home/debugger/README` права rw-r, то есть разрешить владельцу чтение и запись, группе только чтение, остальным пользователям — ничего.

### 2.2. 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. Выполнение лабораторной работы

1. Создайте каталог для программ лабораторной работы № 10, перейдите в него и создайте файлы lab10-1.asm, readme-1.txt и readme-2.txt:

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

Рис 3.1: 31

2. Введите в файл lab10-1.asm текст программы из листинга 10.1 (Программа записи в файл сообщения). Создайте исполняемый файл и проверьте его работу.

```
;-----  
; 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`  
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 3.2: 32

```
ambessonov@fedora:~/work/arch-pc/lab10$ nasm -f elf lab10-1.asm
ambessonov@fedora:~/work/arch-pc/lab10$ ld -m elf_i386 -o lab10-1 lab10-1.o
ambessonov@fedora:~/work/arch-pc/lab10$ ./lab10-1
Введите строку для записи в файл: Bessonov Andrey NKAbd-01-25
```

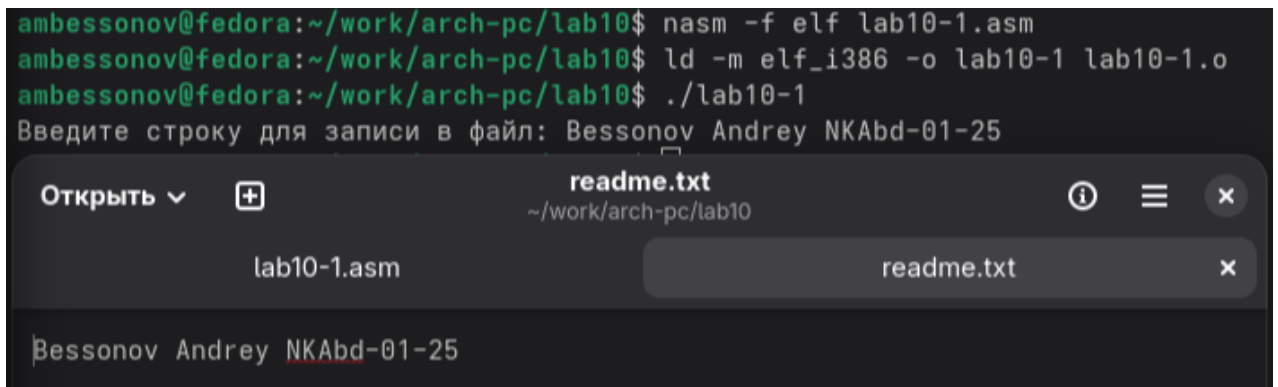


Рис 3.3: 33

3. С помощью команды `chmod` измените права доступа к исполняемому файлу `lab10-1`, запретив его выполнение. Попробуйте выполнить файл. Объясните результат:

```
ambessonov@fedora:~/work/arch-pc/lab10$ chmod a-x lab10-1
ambessonov@fedora:~/work/arch-pc/lab10$ ./lab10-1
bash: ./lab10-1: Отказано в доступе
```

Рис 3.4: 34

Объяснение: после выполнения этой команды бит исполнения (x) в правах доступа к файлу будет снят. Согласно теории из раздела 10.2.1, операционная система определяет, является ли файл исполняемым, именно по наличию этого права. Если право на исполнение отсутствует, ОС не позволит запустить файл как программу. При попытке выполнить файл (командой `./lab10-1`) система вернёт ошибку, сообщающую об отказе в доступе.

4. С помощью команды `chmod` измените права доступа к файлу `lab10-1.asm` с исходным текстом программы, добавив права на исполнение. Попробуйте выполнить его и объясните результат.

```
ambessonov@fedora:~/work/arch-pc/lab10$ chmod a+x lab10-1.asm
ambessonov@fedora:~/work/arch-pc/lab10$ ./lab10-1.asm
./lab10-1.asm: строка 1: синтаксическая ошибка рядом с неожиданным маркером «;»
./lab10-1.asm: строка 1: `;-----'
ambessonov@fedora:~/work/arch-pc/lab10$
```

Рис 3.5.1: 35.1

Объяснение: после выполнения этой команды в правах доступа к файлу `lab10-1.asm` для всех категорий пользователей будет добавлен бит исполнения (x). Однако, как указано в разделе 10.2.1, тип файла определяется первой позицией в выводе команды `ls -l`. Для обычного текстового файла (к которым относится файл с исходным кодом) эта позиция содержит дефис (-). Добавление права на исполнение (x) не изменяет тип файла и не превращает текстовый файл в исполняемую

программу. Поэтому, даже с установленным правом на исполнение, попытка запустить файл напрямую (например, командой `./lab10-1.asm`) завершится ошибкой. Операционная система попытается интерпретировать его содержимое как исполняемый код или скрипт, но первая же строка не будет распознана шелл-интерпретатором или загрузчиком, что приведёт к ошибке.

```
ambessonov@fedora:~/work/arch-pc/lab10$ chmod a+x lab10-1.asm
ambessonov@fedora:~/work/arch-pc/lab10$ nasm -f elf lab10-1.asm
ambessonov@fedora:~/work/arch-pc/lab10$ ld -m elf_i386 -o lab10-1 lab10-1.o
ambessonov@fedora:~/work/arch-pc/lab10$ ./lab10-1
Введите строку для записи в файл: Architektura komputera
```

Рис 3.5.1: 35.1

Если выполнять действие с исполняемым файлом, то ошибок не будет.

5. В соответствии с вариантом в таблице 10.4 предоставить права доступа к файлу `readme1.txt` представленные в символьном виде, а для файла `readme-2.txt` – в двоичном виде. Проверить правильность выполнения с помощью команды `ls -l`.

```
ambessonov@fedora:~/work/arch-pc/lab10$ chmod u=wx,g=x,o=rwx readme-1.txt
ambessonov@fedora:~/work/arch-pc/lab10$ chmod 252 readme-2.txt
ambessonov@fedora:~/work/arch-pc/lab10$ ls -l readme-1.txt readme-2.txt
--wx--xrwx. 1 ambessonov ambessonov 0 дек  7 19:43 readme-1.txt
--w-r-x-w-. 1 ambessonov ambessonov 0 дек  7 19:43 readme-2.txt
ambessonov@fedora:~/work/arch-pc/lab10$
```

Рис 3.6: 36

Проверили и убедились в правильности.

Вывод: выполнили задания лабораторной работы.

## 4. Выполнение самостоятельной работы

1. Напишите 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

Создать исполняемый файл и проверить его работу. Проверить наличие файла и его содержимое с помощью команд `ls` и `cat`.



Решение:

```
_start:
; --- Шаг 1: Вывод приглашения "Как Вас зовут?" ---
mov eax, prompt      ; Загружаем адрес приглашения
call sprint           ; Выводим строку на экран

; --- Шаг 2: Ввод с клавиатуры фамилии и имени ---
mov ecx, name         ; Адрес буфера для ввода
mov edx, 255          ; Максимальная длина ввода
call sread            ; Читаем строку с клавиатуры

; --- Шаг 3: Создание файла с именем name.txt ---
mov ecx, 0644o        ; Права доступа: rw-r--r--
mov ebx, filename     ; Имя файла: "name.txt"
mov eax, 8            ; Номер системного вызова sys_creat
int 80h              ; Вызов ядра
; В eax теперь дескриптор созданного файла

mov esi, eax          ; Сохраняем дескриптор файла

; --- Шаг 4: Запись в файл сообщения "Меня зовут" ---
mov eax, msg          ; Адрес сообщения
call slen             ; Вычисляем длину строки
mov edx, eax          ; Длина строки для записи
mov ecx, msg          ; Адрес строки "Меня зовут "
mov ebx, esi          ; Дескриптор файла
mov eax, 4            ; sys_write
int 80h

; --- Шаг 5: Дописать в файл строку, введенную с клавиатуры ---
; Вычисляем длину введенной строки и убираем символ новой строки
mov eax, name         ; Адрес введенной строки
call slen             ; Вычисляем длину

; Уменьшаем длину на 1, чтобы убрать символ новой строки
dec eax

; Записываем введенную строку в файл (без символа новой строки)
mov edx, eax          ; Длина строки для записи
mov ecx, name         ; Адрес введенной строки
mov ebx, esi          ; Дескриптор файла
mov eax, 4            ; sys_write
int 80h

; --- Добавляем символ новой строки в конец файла ---
mov edx, 1            ; Длина = 1 символ
mov ecx, newline      ; Адрес символа новой строки
mov ebx, esi          ; Дескриптор файла
mov eax, 4            ; sys_write
int 80h

; --- Шаг 6: Закрыть файл ---
mov ebx, esi          ; Дескриптор файла
mov eax, 6            ; sys_close
int 80h

; --- Завершение программы ---
call quit             ; Корректный выход
```

Рис 4.1: 41

Создадим исполняемый файл и проверим его работу. Проверим наличие файла и его содержимое с помощью команд `ls` и `cat`.

```
ambessonov@fedora:~/work/arch-pc/lab10$ touch work.asm
ambessonov@fedora:~/work/arch-pc/lab10$ nasm -f elf work.asm
ambessonov@fedora:~/work/arch-pc/lab10$ ld -m elf_i386 -o work work.o
ambessonov@fedora:~/work/arch-pc/lab10$ ./work
Как Вас зовут? AndreyBessonov
ambessonov@fedora:~/work/arch-pc/lab10$ ls -l name.txt
-rw-r--r--. 1 ambessonov ambessonov 35 дек  7 21:33 name.txt
ambessonov@fedora:~/work/arch-pc/lab10$ cat name.txt
Меня зовут AndreyBessonov
ambessonov@fedora:~/work/arch-pc/lab10$ █
```

Рис 4.2: 42

Вывод: выполнили задания самостоятельной работы, отработали навыки, полученные в ходе лабораторной работы.

## 5. Выводы

Приобрели навыки написания программ для работы с файлами.