Отчёт по лабораторной работе №10

дисциплина Архитектура компьютера

Терещенкова Маргарита Владимировна

Содержание

# 1 Цель работы

Приобретение навыков написания программ для работы с файлами.

# 2 Задание

1. Создание файлов в программах
2. Изменение прав на файлы для разных групп пользователей
3. Выполнение самостоятельных заданий по материалам лабораторной работы.

# 3 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.

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.

# 4 Выполнение лабораторной работы

1. Создаю каталог для программам лабораторной работы № 10, перехожу в него и создаю файлы lab10-1.asm, readme-1.txt и readme-2.txt.

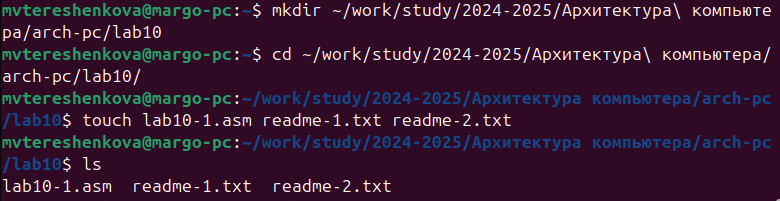


Рис. 1: Создание каталога и файлов

1. Ввожу в файл lab10-1.asm текст программы из листинга 10.1.

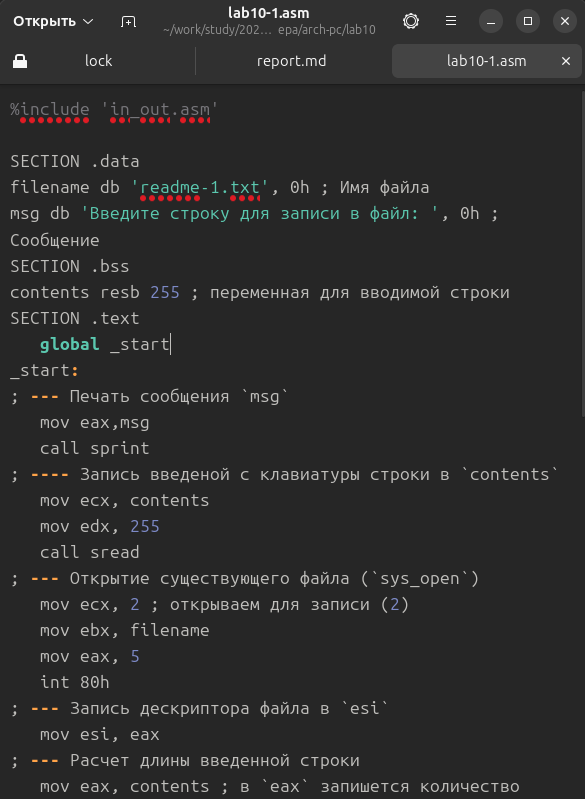


Рис. 2: Редактирование файла

Создаю исполняемый файл.

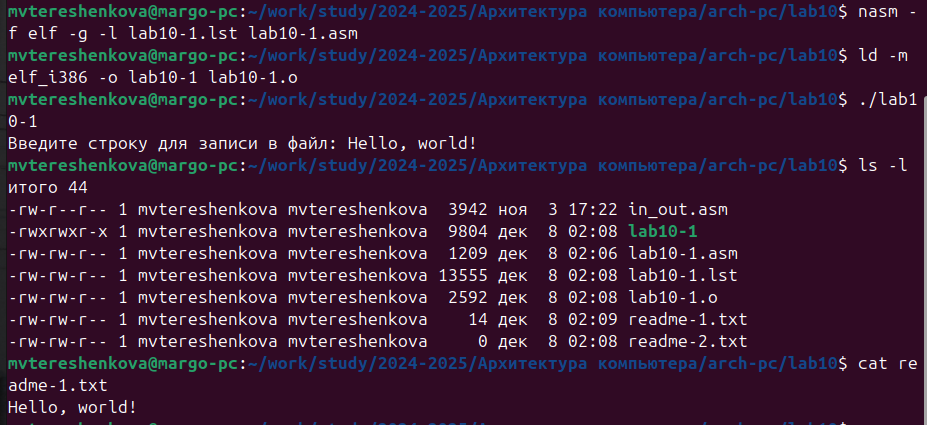


Рис. 3: Запуск исполняемого файла

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

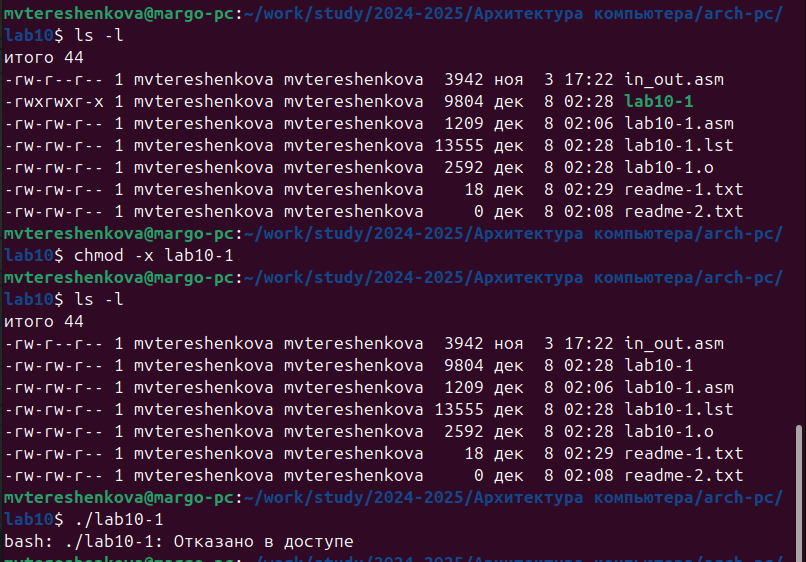


Рис. 4: Запуск исполняемого файла

В результате выполнения отказано в доступе, и при написании команды ls -l, в правом поле (-rw-rw-r–) показывает, что файл больше не имеет права на выполнение (x исчезло). Ошибка возникает, потому что права на выполнение файла были удалены. Когда права на выполнение (execute) отсутствуют, операционная система запрещает запускать файл, даже если это действительный исполняемый файл.

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

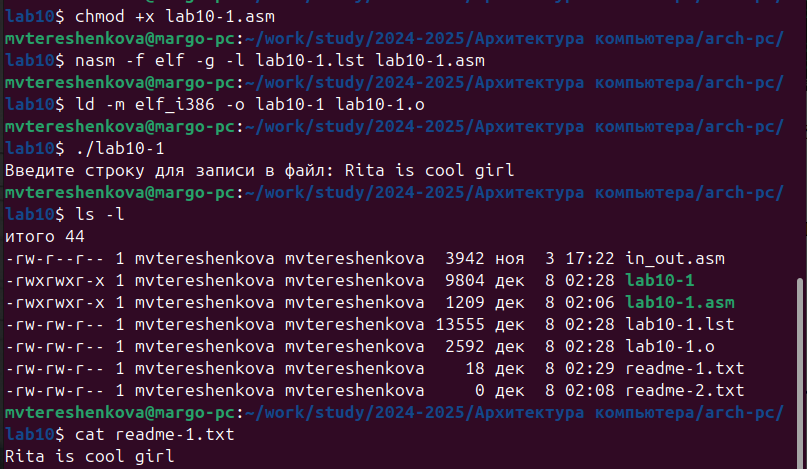


Рис. 5: Запуск исполняемого файла

Файл выполняется. Также можно заметить, что у файлов lab10-1 и lab10-1.asm есть флаг x (execute), указывающий, что файл может быть исполнен.

1. В соответствии с вариантом в таблице 10.4, мой вариант - 4, предоставляю права доступа к файлу readme- 1.txt представленные в символьном виде, а для файла readme-2.txt – в двочном виде. Проверяю правильность выполнения с помощью команды ls -l.

* Для **readme-1.txt**:

• **u=w** — оставляю только запись для владельца

• **g=** — убираю все права для группы

• **o=w** — оставляю только запись для остальных

* Для **readme-2.txt**:

Преобразую в восьмеричный формат:

• **001** (владелец) → 1 (только выполнение)

• **011** (группа) → 3 (запись и выполнение)

• **110** (остальные) → 6 (чтение и запись)

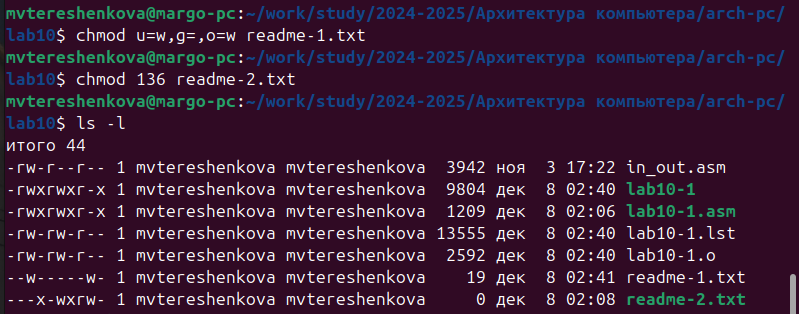


Рис. 6: Запуск исполняемого файла

Права доступа выданы верно.

# 5 Выполнение заданий для самостоятельной работы

Создаю файл lab10-2.asm и пишу программу по заданию.

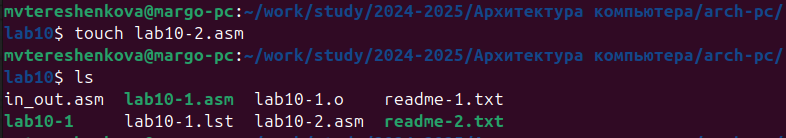


Рис. 7: Создание файла

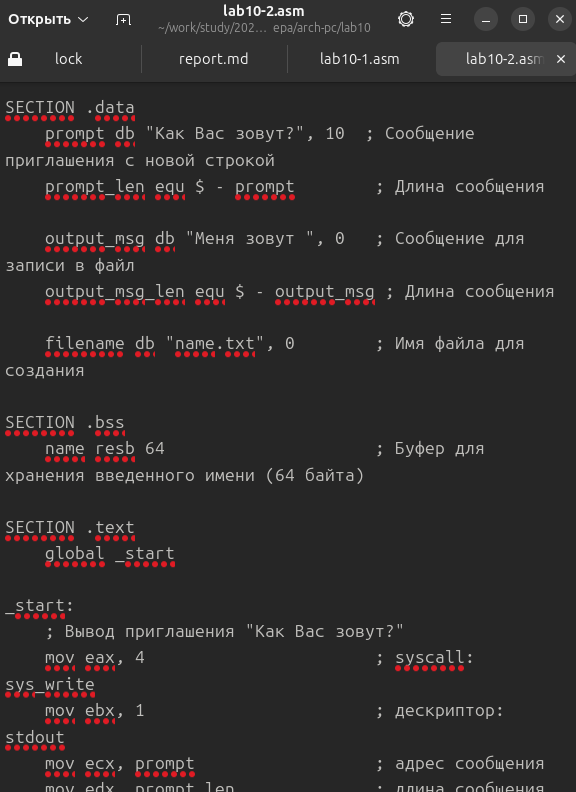


Рис. 8: Написание программы

Cоздаю объектный файл и запускаю программу.

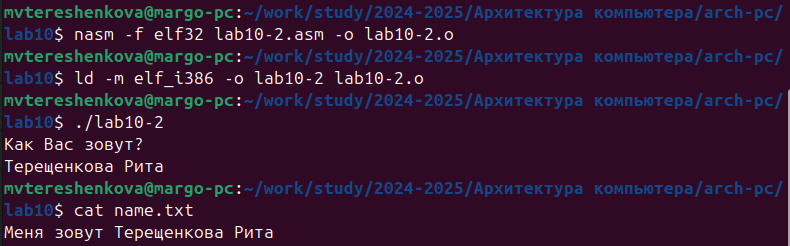


Рис. 9: Запуск исполняемого файла

Программа работает корректно.

**Код программы:**

SECTION .data

prompt db “Как Вас зовут?”, 10

prompt\_len equ $ - prompt

output\_msg db “Меня зовут”, 0

output\_msg\_len equ $ - output\_msg

filename db “name.txt”, 0

SECTION .bss

name resb 64

SECTION .text

global \_start

\_start:

mov eax, 4

mov ebx, 1

mov ecx, prompt

mov edx, prompt\_len

int 0x80

mov eax, 3

mov ebx, 0

mov ecx, name

mov edx, 64

int 0x80

mov eax, 5

mov ebx, filename

mov ecx, 0101b

mov edx, 0644

int 0x80

mov edi, eax

mov eax, 4

mov ebx, edi

mov ecx, output\_msg

mov edx, output\_msg\_len

int 0x80

mov eax, 4

mov ebx, edi

mov ecx, name

mov edx, 64

int 0x80

mov eax, 6

mov ebx, edi

int 0x80

mov eax, 1

xor ebx, ebx

int 0x80

# 6 Выводы

Благодаря данной лабораторной работе приобрела навыки написания программ для работы с файлами.

# Список литературы

1. Архитектура компьютера