Отчёт по лабораторной работе №10

Дисциплина: Архитектура компьютера

Абрамова Ульяна Михайловна

Содержание

# 1 Цель работы

Приобретение навыков написания программ для работы с файлами.

# 2 Задание

1. Написание программы записи в файл сообщения
2. Изменение прав доступа
3. Задание для самостоятельной работы

# 3 Теоретическое введение

## 3.1 Права доступа к файлам

ОС GNU/Linux является многопользовательской операционной системой. И для обеспече- ния защиты данных одного пользователя от действий других пользователей существуют специальные механизмы разграничения доступа к файлам. Кроме ограничения доступа, дан- ный 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. Владельцем файла является его создатель. Для предоставления прав доступа другому пользователю или другой группе командой chown [ключи] <новый\_пользователь>[:новая\_группа] <файл> или chgrp [ключи] < новая\_группа > <файл> Набор прав доступа задается тройками битов и состоит из прав на чтение, запись и ис- полнение 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 х — исполнение), а дефис означает отсутствие права (нулевое значение соответствующего бита). Также права доступа могут быть представлены как вось- меричное число. Так, права доступа rw- (чтение и запись, без исполнения) понимаются как три двоичные цифры 110 или как восьмеричная цифра 6.

Полная строка прав доступа в символьном представлении имеет вид:

Так, например, права rwx r-x –x выглядят как двоичное число 111 101 001, или вось- меричное 751. Свойства (атрибуты) файлов и каталогов можно вывести на терминал с помощью команды ls с ключом -l. Так например, чтобы узнать права доступа к файлу README можно узнать с помощью следующей команды: $ls -l /home/debugger/README -rwxr-xr-- 1 debugger users 0 Feb 14 19:08 /home/debugger/README В первой колонке показаны текущие права доступа, далее указан владелец файла и группа. 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 файл 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: $chmod 640 README # 110 100 000 == 640 == rw-r----- $ls -l README -rw-r 1 debugger users 0 Feb 14 19:08 /home/debugger/README В символьном представлении есть возможность явно указывать какой группе какие права необходимо добавить, отнять или присвоить. Например, чтобы добавить право на исполне- ние файла README группе и всем остальным: $chmod go+x README $ls -l README -rw-r-x--x 1 debugger users 0 Feb 14 19:08 /home/debugger/README Формат символьного режима: chmod

## 3.2 Работа с файлами средствами Nasm

В операционной системе Linux существуют различные методы управления файлами, на- пример, такие как создание и открытие файла, только для чтения или для чтения и записи, добавления в существующий файл, закрытия и удаления файла, предоставление прав досту- па. Обработка файлов в операционной системе Linux осуществляется за счет использования определенных системных вызовов. Для корректной работы и доступа к файлу при его от- крытии 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.

# 4 Выполнение лабораторной работы

## 4.1 Написание программы записи в файл сообщения

Создаю каталог для программ лабораторной работы №10 и, перейдя в него, создаю файлы (рис.1 1)

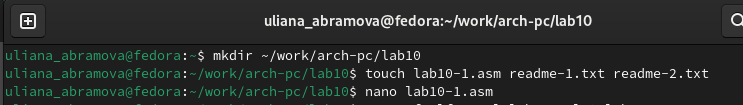


Рис. 1: Создание каталога и файлов

Ввожу в файл lab10-1.asm программу из листинга (рис. 2)

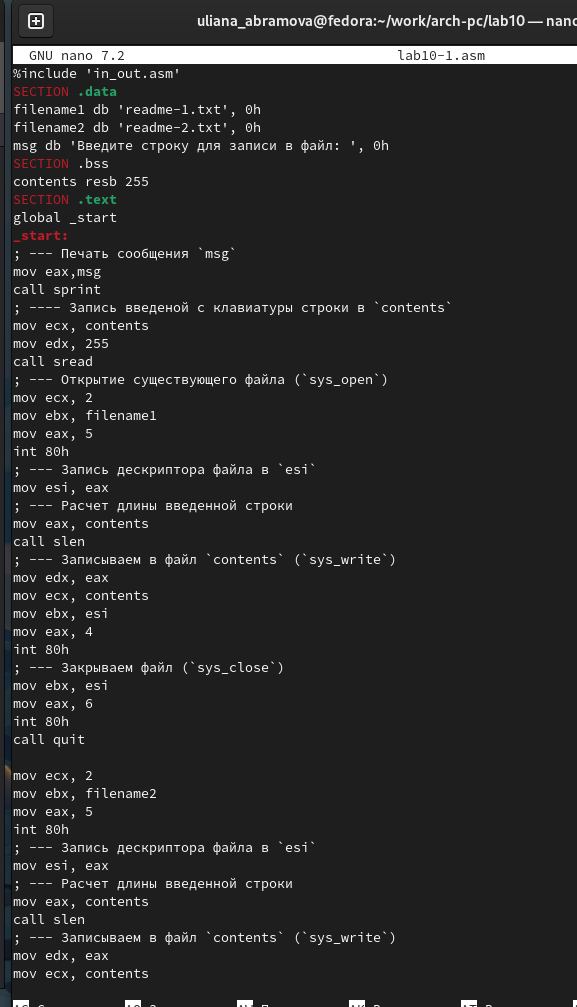


Рис. 2: Написание программы

Создаю исполняемый фал и проверяю корректность его работы (рис.3)

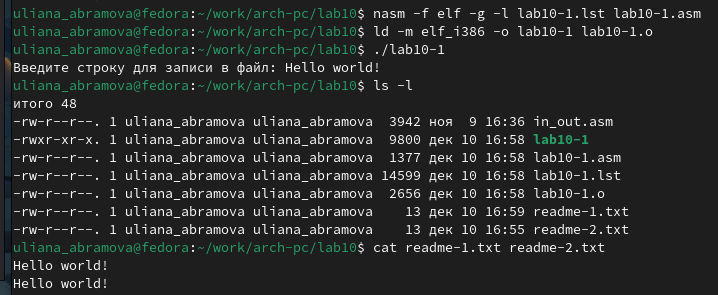


Рис. 3: Запуск программы

## 4.2 Изменение прав доступа

### 4.2.1 к исполняемому файлу

С помощью команды chmod изменяю права доступа к исполняемому файлу lab10-1, запретив его выполнение (рис. 4)

Рис. 4: Изменение прав доступа к исполняемому файлу

Рис. 4: Изменение прав доступа к исполняемому файлу

В результате, выдало отказ в доступе, так как я запретила запускать программу для владельца, то есть для себя

### 4.2.2 к файлу

С помощью команды chmod изменяю права доступа к файлу lab10-1.asm с исходным текстом программы, добавив права на исполнение (рис. 5)

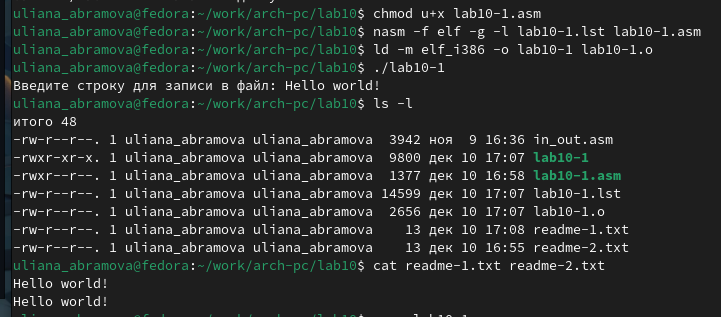


Рис. 5: Изменение прав доступа к файлу

После изменения прав программа заработала, так как по факту она новая в результате перекомпелирования и обладает другими разрешениями

Далее я предоставила в соответсвии со своим вариантом права доступа к файлу readme-1.txt представленные в символьном виде (рис. 6), а для файла readme-2.txt - в двоичном виде (рис. 7)

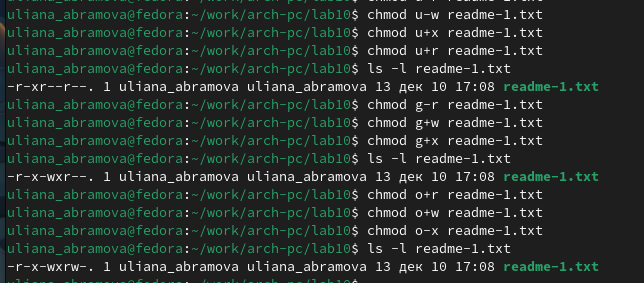


Рис. 6: Предоставление прав доступа, представленных в символьном виде

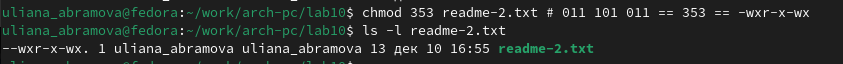


Рис. 7: Предоставление прав доступа, представленных в двоичном виде

## 4.3 Задание для самостоятельной работы

Сначала я написала программу которая работает по определенному 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

(рис. 8)

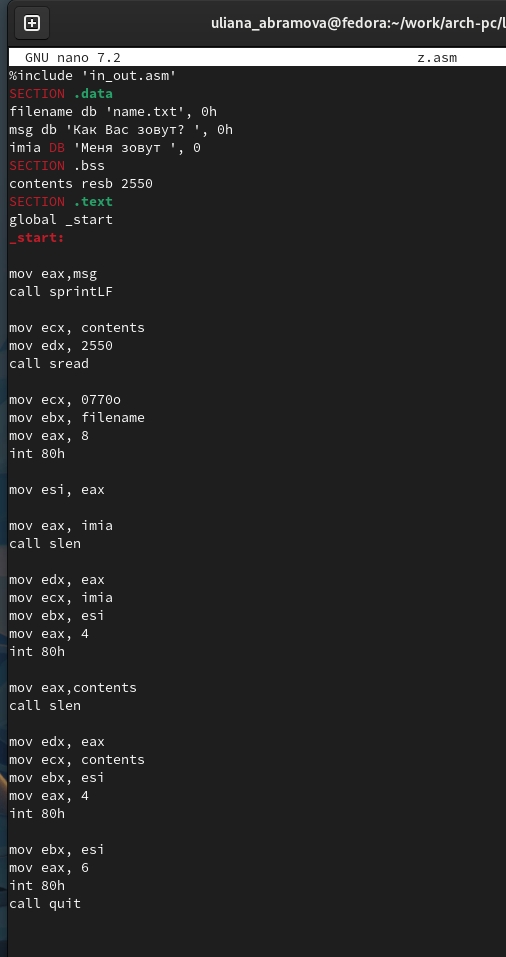


Рис. 8: Написание программмы

Затем создаю исполняемый файл и проверяю его работу (рис. 9)

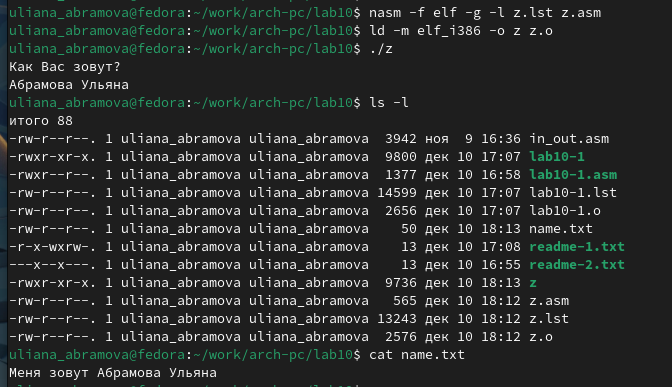


Рис. 9: Запуск программы

Программы отработана исправно

# 5 Выводы

Я приобрела навыки написания программ для работы с файлами.

# 6 Список литературы

1. [Архитектура ЭВМ](https://esystem.rudn.ru/pluginfile.php/2089554/mod_resource/content/0/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%9610.%20%D0%A0%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%D1%81%20%D1%84%D0%B0%D0%B9%D0%BB%D0%B0%D0%BC%D0%B8%20%D1%81%D1%80%D0%B5%D0%B4%D1%81%D1%82%D0%B2%D0%B0%D0%BC%D0%B8%20Nasm.pdf)