Отчёта по лабораторной работе No11

ОНВУДИВЕ ВИКТОР ЧИБУИКЕ!

Содержание

# 1 Цель работы

Приобрестни навыки работы с файлами в языке NASM и научиться управлять правами доступа к файлам.

# 2 Задание

1. Изучите примеры реализации подпрограмм
2. Использование программы записи сообщений в файл сообщений
3. Выполните самостоятеьное задание
4. Загрузите файлы на GitHub.

# 3 Теоретическое введение

ОС GNU/Linux является многопользовательской операционной системой. И для обеспечения защиты данных одного пользователя от действий других поль- зователей существуют специальные механизмы разграничения доступа к фай- лам. Кроме ограничения доступа, данный механизм позволяет разрешить дру- гим пользователям доступ данным для совместной работы. Права доступа определяют набор действий (чтение, запись, выполнение), раз- решённых 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 предостав- ления прав доступа другому пользователю или другой группе командой chown [ключи] [:новая\_группа] или chgrp [ключи] < новая\_группа > 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 комбина- ций, приведенных в таблице 11.1. 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 соответству- ющего 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.

# 4 Выполнение лабораторной работы

Создадим рабочую дерикторию и файл, запишем туда код программы из листинга. (рис. 1)

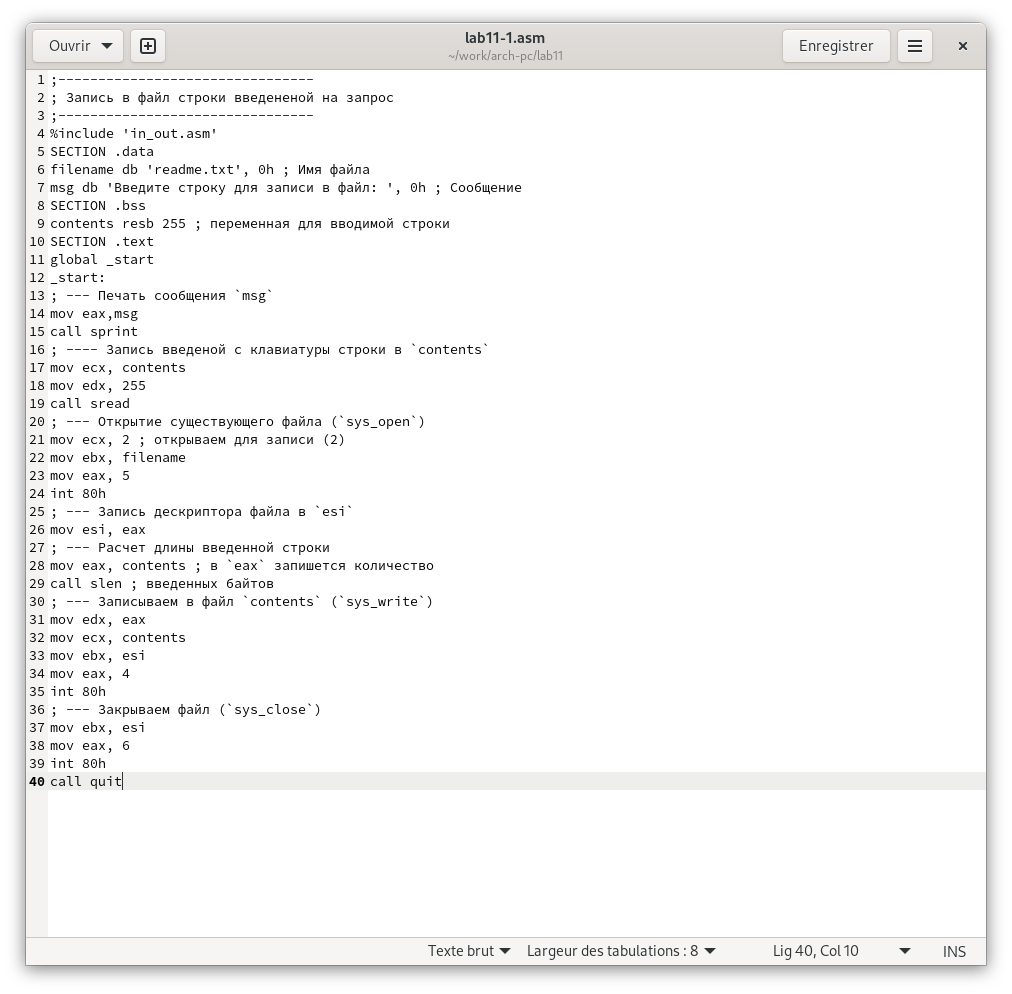


Рис. 1: Текст программы

Проассемблируем программу и проверим ее работу(рис. 2)

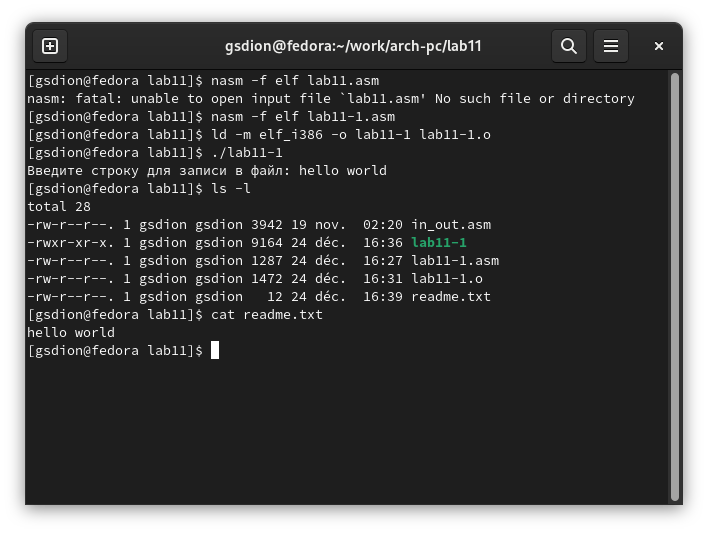


Рис. 2: работа программы

Запретим исполнение для файла lab11-1.(рис. 3)

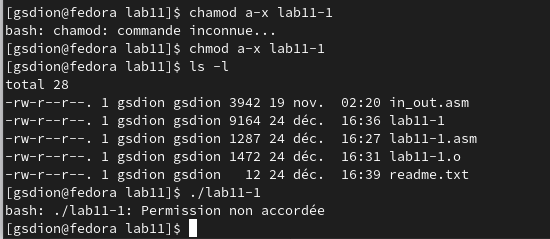


Рис. 3: как и ожидалось, мы не смогли исполнить этот файл

если запретить исполнение файла, то исполнить его станет невозможно.

Когда мы разрешим исполнение файла с расширением .asm и собственно исполним его, то мы увидем множество ошибок, ведь этот файл не предназначен для такого использования.(рис. 4)

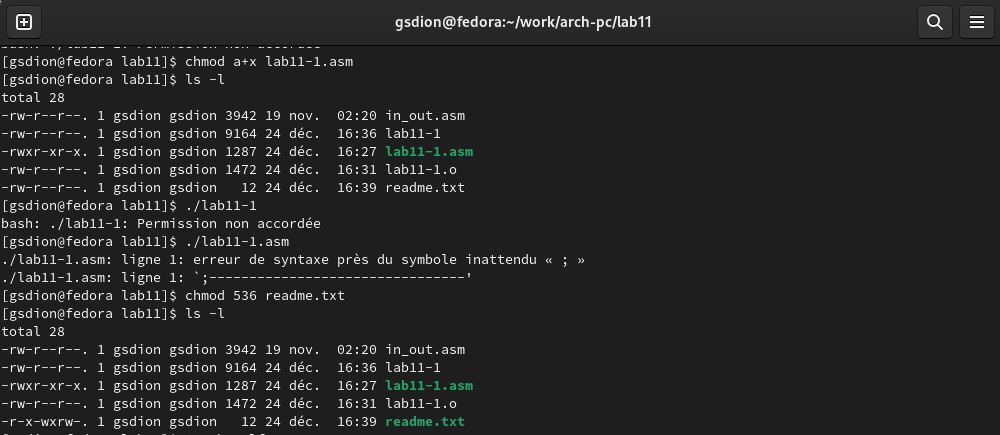


Рис. 4: Ошибки исполнения файла lab11-1.asm

Зададим файлу readme.txt права использования как во варианте 17 и проверим как получилось.(рис. 5)

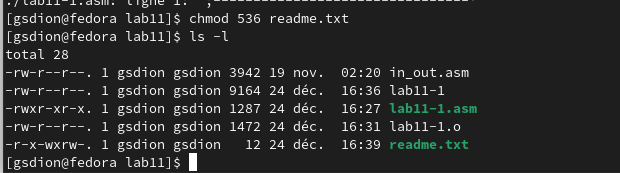


Рис. 5: Права доступа к файлу readme.txt r-x -wx rw-

# 5 Задания для самостоятельной работы

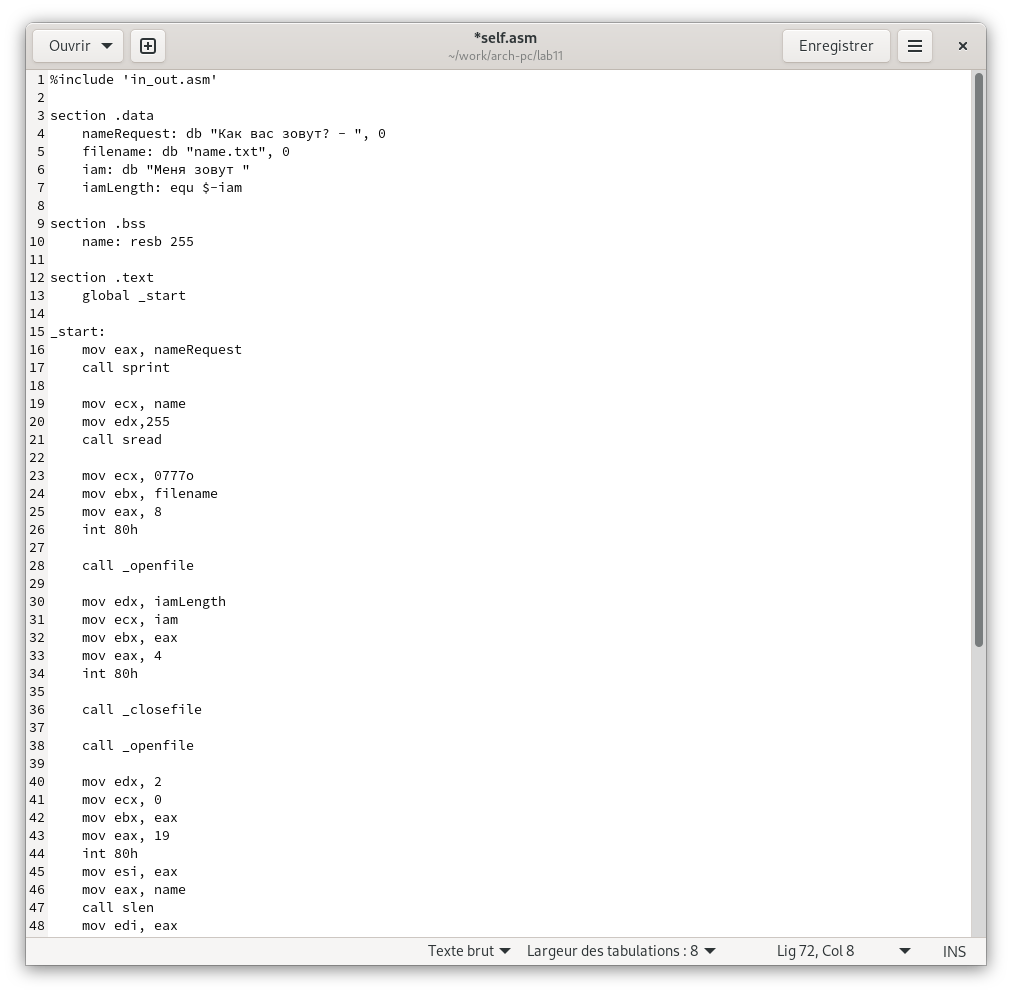


Рис. 6: часть текста программы

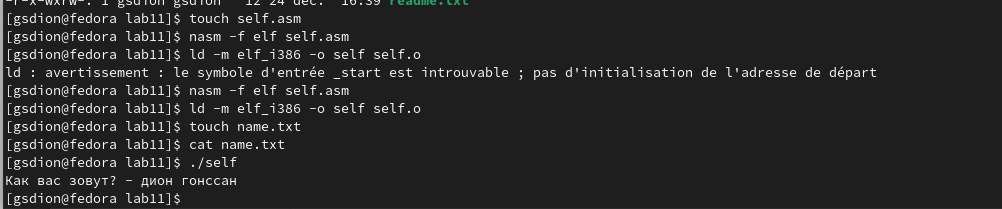


Рис. 7: работа полученной програмы

# 6 Выводы

В заключение мы приобрели навыки работы с файлами в NASM и разрешениями файлов.

# Список литературы

1. [Расширенный ассемблер: NASM](https://www.opennet.ru/docs/RUS/nasm/)
2. [MASM, TASM, FASM, NASM под Windows и Linux](https://habr.com/ru/post/326078/)