РУДН. Архитектура компьютеров

Отчёт по лабораторной работе №10

Косинов Никита Андреевич, НПМбв-02-20

Содержание

# 1 Цель работы

Очень часто программе необходимо взаимодействовать с файлами: записывать туда данные для дальнейшей работы, брать данные. Цель данной работы - разобраться, как можно создавать, открывать, записывать информацию в файле программой на языке ассемблер, а также разобраться, что такое права доступа и как их изменять.

# 2 Ход работы

Лабораторная работа выполнена с использованием консоли **OC Linux** и языка программирования ассемблера **NASM**.

1. Изменение прав доступа;
2. Написание программы с записью в файл.

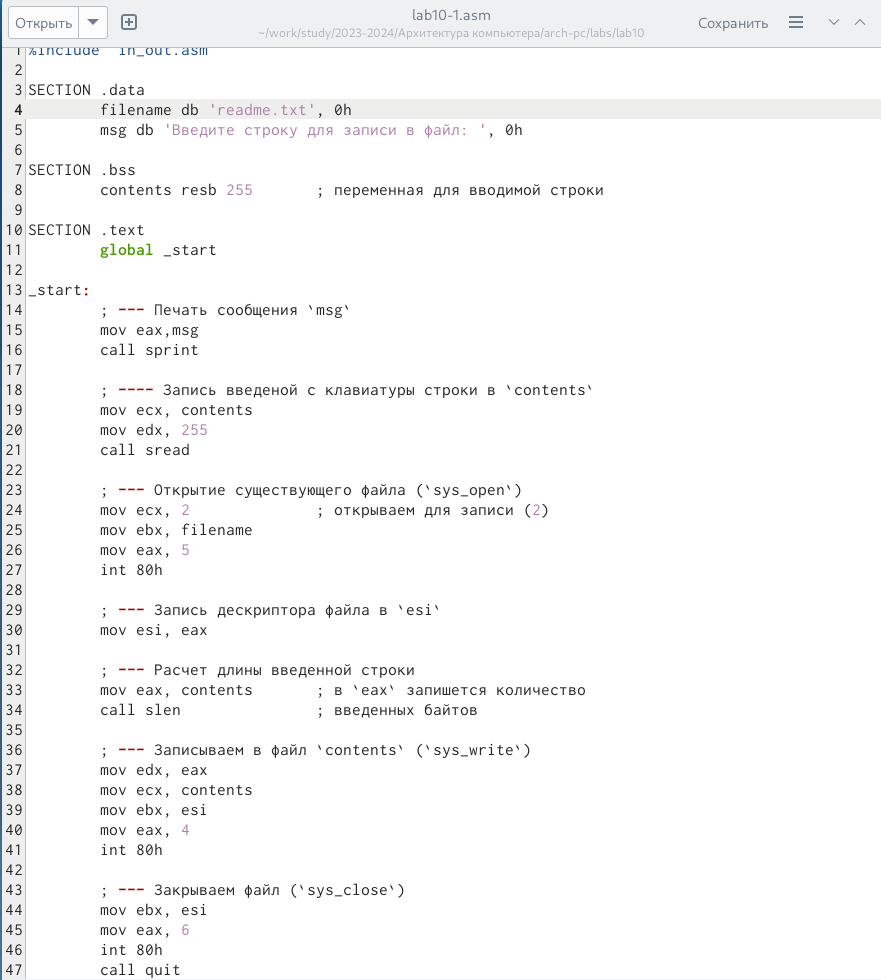
# 3 Изменение прав доступа

1. Создаём рабочий файл.

Создание файла

Создание файла

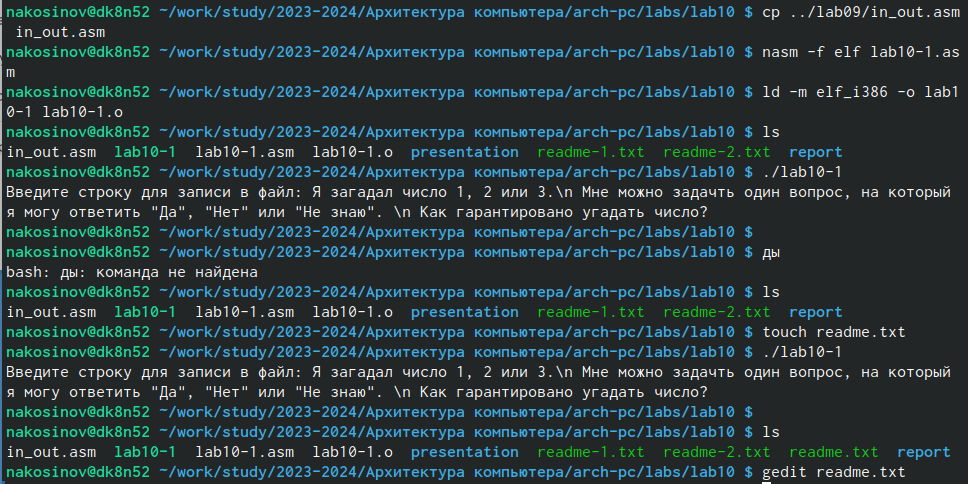
1. Пишем код программы, считывающей строку с терминала и записывающую эту строку в файл.



Код программы lab10-1

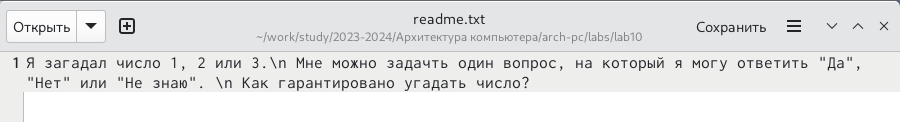
%include 'in\_out.asm'  
  
SECTION .data  
 filename db 'readme.txt', 0h   
 msg db 'Введите строку для записи в файл: ', 0h   
  
SECTION .bss  
 contents resb 255 ; переменная для вводимой строки  
  
SECTION .text  
 global \_start  
  
\_start:  
 ; --- Печать сообщения `msg`  
 mov eax,msg  
 call sprint  
   
 ; ---- Запись введеной 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

1. Компилируем и исполняем. Видим, что первая попытка запуска неудачная, ведь в коде открывается уже существующий файл. Поэтому для проверки работы мы создадим файл *readme* и запустим ещё раз.



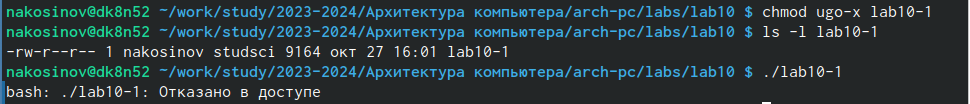
Исполнение lab10-1

1. Откроем текстовый файл *readme* и убедимся, что программа работает корректно.



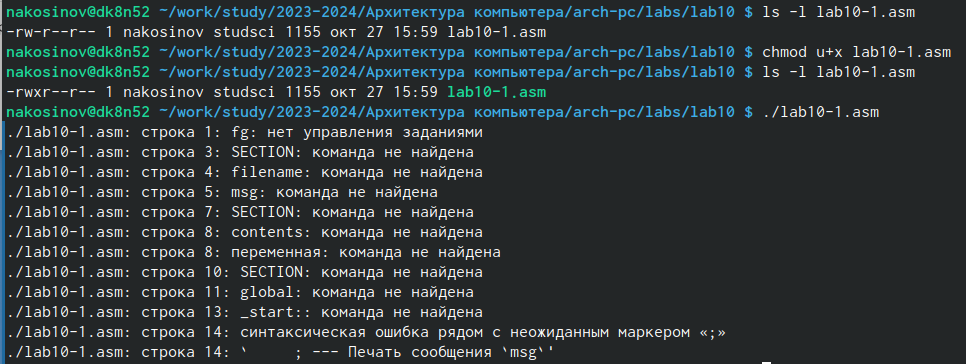
Результат работы lab10-1

1. Изменим права доступа командой *chmod*, запретив исполнения для всех категорий пользователей. Проверим корректность командой *ls* с ключом **-l**. Видим, что у файла *lab10-1* в строке прав доступа отсутствуют буквы *x*. Ожидаемо, исполнить файл у нас не выходит.



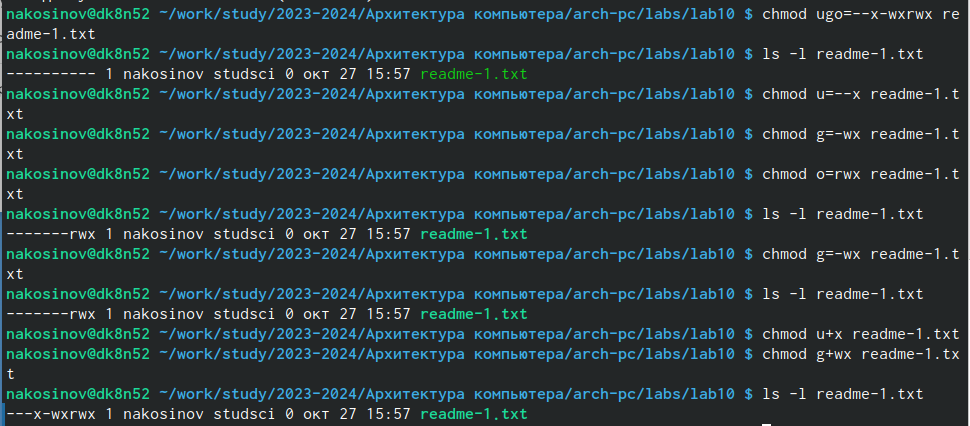
Попытка исполнения файла, для которого запрещено исполнение

1. Добавим право запускать файл владельцу файла *lab10-1.asm*. Попробуем исполнить его. Но этот файл - не скомпилированная программа, и ЭВМ не понимает, как он может его исполнить, показывая нам, что не знает команд, введённых в этот файл.



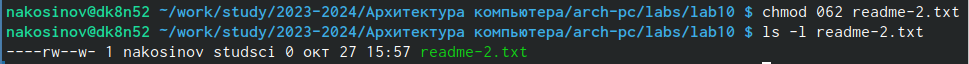
Попытка исполнения текстового файла

1. Изменим права доступа к файлу *readme-1* в символьном виде в соответствии с вариантом 1: *–x -wx rwx*. В ходе нескольких попыток видим, что задать сразу все права таким образом не очень удобно: приходится вводить несколько команд.



Изменение прав доступа в символьном виде

1. Изменим права доступа к файлу *readme-2* в двоичном виде в соответствии с вариантом 1: *000 110 010*. Для этого переводим каждую тройку бит в десчтичную цифру. Одной командой настраиваем права доступа полностью.

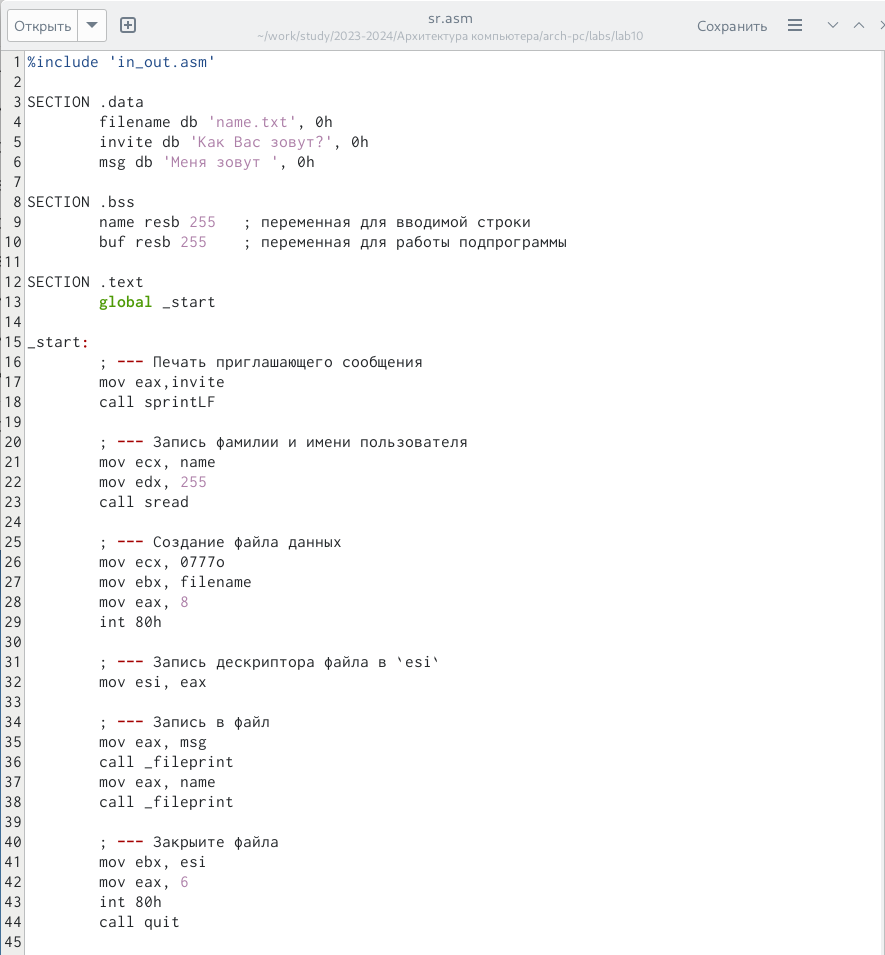


Изменение прав доступа в числовом виде

# 4 Написание программы с записью в файл

Задание для самостоятельной работы заключается в написании программы, запрашивающей имя пользователя и записывающей этой имя в файл, добавляя в начало приветственные слова.

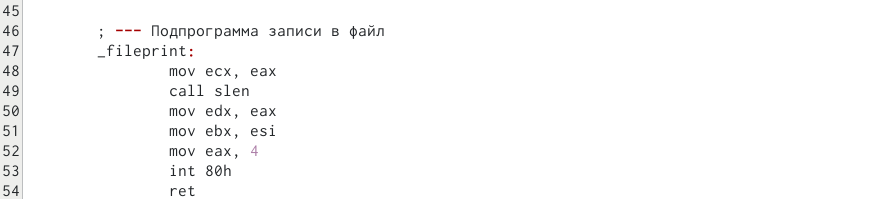
1. Создаём файл с кодом программы и добавляем комментарии.



Основной код программы sr.asm

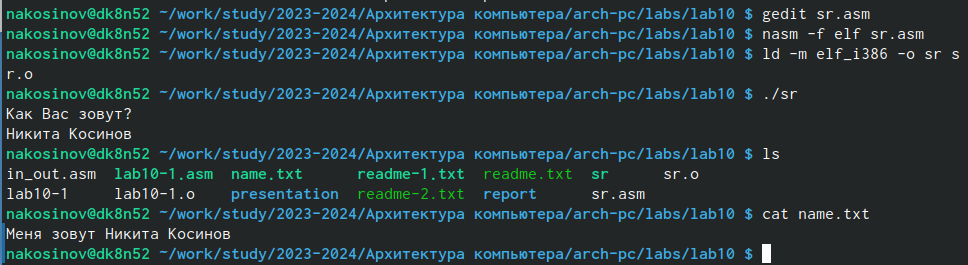
%include 'in\_out.asm'  
  
SECTION .data  
 filename db 'name.txt', 0h   
 invite db 'Как Вас зовут?', 0h   
 msg db 'Меня зовут ', 0h  
  
SECTION .bss  
 name resb 255 ; переменная для вводимой строки  
 buf resb 255 ; переменная для работы подпрограммы  
  
SECTION .text  
 global \_start  
  
\_start:  
 ; --- Печать приглашающего сообщения  
 mov eax,invite  
 call sprintLF  
   
 ; --- Запись фамилии и имени пользователя  
 mov ecx, name  
 mov edx, 255  
 call sread  
   
 ; --- Создание файла данных  
 mov ecx, 0777o  
 mov ebx, filename  
 mov eax, 8  
 int 80h  
   
 ; --- Запись дескриптора файла в `esi`  
 mov esi, eax  
   
 ; --- Запись в файл  
 mov eax, msg  
 call \_fileprint  
 mov eax, name  
 call \_fileprint  
   
 ; --- Закрыите файла  
 mov ebx, esi  
 mov eax, 6  
 int 80h  
 call quit  
   
 ; --- Подпрограмма записи в файл  
 \_fileprint:  
 mov ecx, eax  
 call slen  
 mov edx, eax  
 mov ebx, esi  
 mov eax, 4  
 int 80h  
 ret

1. В файл необходимо дважды записать сообщения: приветственное *“Меня зовут”* и непосредственно имя пользователя. Поэтому, для удобства, выносим печать в файл в подпрограмму \*\_fileprint\*.



Подпрограмма \_fileprint

1. Компилируем программу и запускаем. Вводим свои имя и фамилию. Проверяем, что файл с именем создан командой *ls*. Командой *cat* проверяем, что в созданном файле данные сохранены и в нужном нам формате.



СИсполнение и результат sr

# 5 Выводы

В ходе данной лабораторной работы мы научились работать с файлами средствами **NASM**, а также добавлять и отнимать права доступа к файлам разным группам пользователей через терминал.