Отчёт по лабораторной работе №10

Дисциплина: архитектура компьютера

Чернятьева Олеся Олеговна

Содержание

# 1 Цель работы

Приобретение навыков написания программ для работы с файлами в NASM.

# 2 Выполнение лабораторной работы

**Шаг 1**

С помощью утилиты mkdir создаю директорию lab10, перехожу в нее и создаю файл для работы. (рис. [[1](#fig:001)])

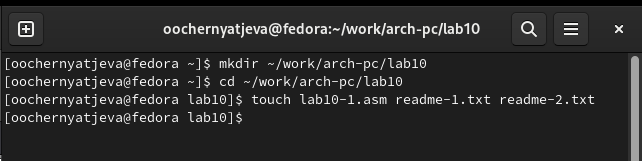


Figure 1: Создание директории

**Шаг 2**

Открываю созданный файл lab10-1.asm, вставляю в него программу из Листинга.(рис. [[2](#fig:002)]).

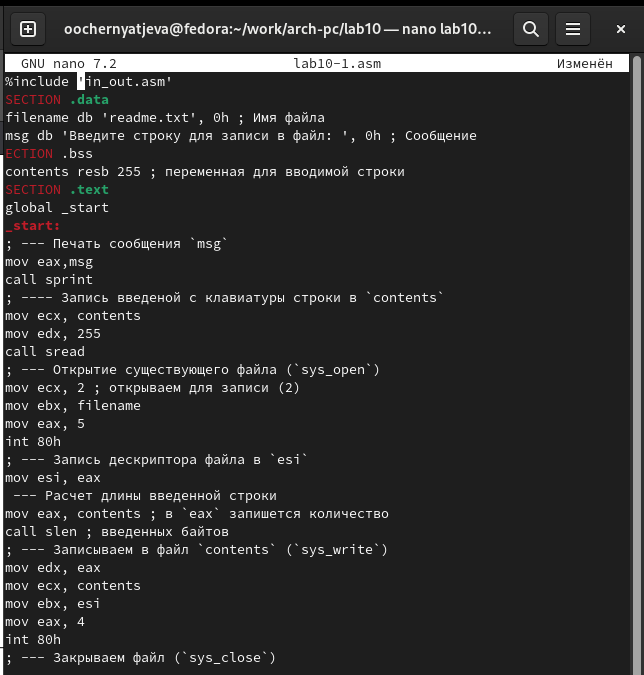


Figure 2: Редактирование текста

**Шаг 3**

Создаю исполняемый файл программы и ввожу свое имя и фамилию, и сразу проверяю работую (рис. [[3](#fig:003)]).

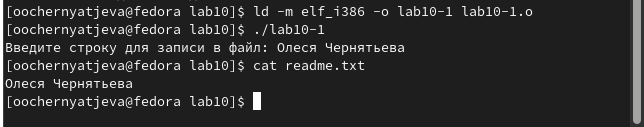


Figure 3: Запуск исполняемого файла

**Шаг 4**

С помощью команды chmod изменяю права доступа к исполняемому файлу lab10-1, запретив его выполнение. (рис. [[4](#fig:004)]).

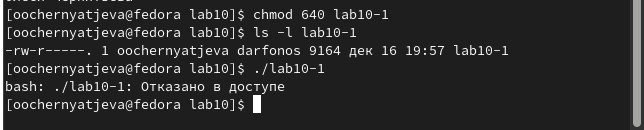


Figure 4: Доступ к файлу

1. В символьном представлении есть возможность явно указывать какой группе какие права необходимо добавить или отнять. Так ‘6’ дает право владельцу лишь для чтения и записи, но не для исполнения.
2. ‘4’ дает право группе лишь для чтения.
3. ‘0’ отнимает все права для всех прочих пользователей.

**Шаг 5**

С помощью команды chmod изменяю права доступа к файлу lab10-1.asm с исходным текстом программы, добавив права на исполнение. (рис. [[5](#fig:005)]).

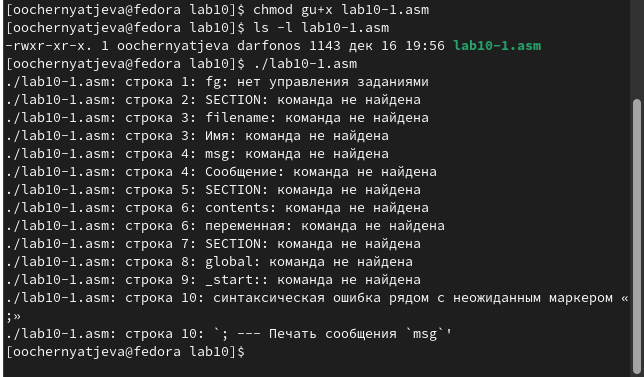


Figure 5: Изменение прав к файлу

* С Помощью аргументов “u” и “+” добавляем право владельцу на исполнение файла, однако в файле находится исходный код программы, который еще не ассемблирован, и его выполнение невозможно.
* Происходит исполнение файла, но она не будет работать.

**Шаг 6**

В соответствии со своим 10-ым вариантом в таблице, предоставляю права доступа к файлу readme-1.txt представленные в символьном виде. (рис. [[6](#fig:006)]).

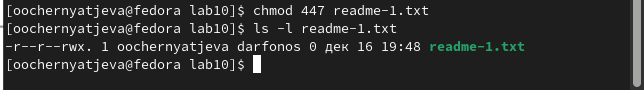


Figure 6: Предоставляю права доступа к файлу

* Вначале проверяю изначальные права, которые имеют пользователи к файлу по умолчанию, затем даю ей другие права в соответствии с вариатном.

**Шаг 7**

Предоставляю права доступа к файлу readme-2.txt представленные в символьном – в двочном виде. (рис. [[7](#fig:007)]).

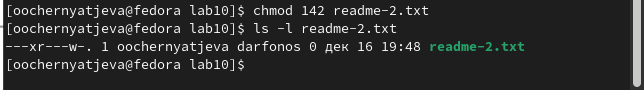


Figure 7: Предоставляю права доступа к файлу

# 3 Самостоятельная работа

**Шаг 1**

Создаю новый файл для самостояльной работы(рис.[[8](#fig:008)]).

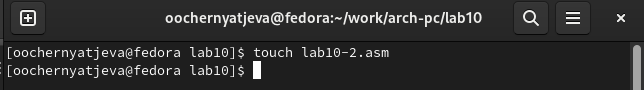


Figure 8: Создание файла для самостоятельной

**Шаг 2**

Пишу в неё текст программы, которая будет запрашивать имя пользователя, будет создавать файл и записывать в нее введенное имя, добавив в начале текст : “Меня зовут”. (рис. [[9](#fig:009)]).

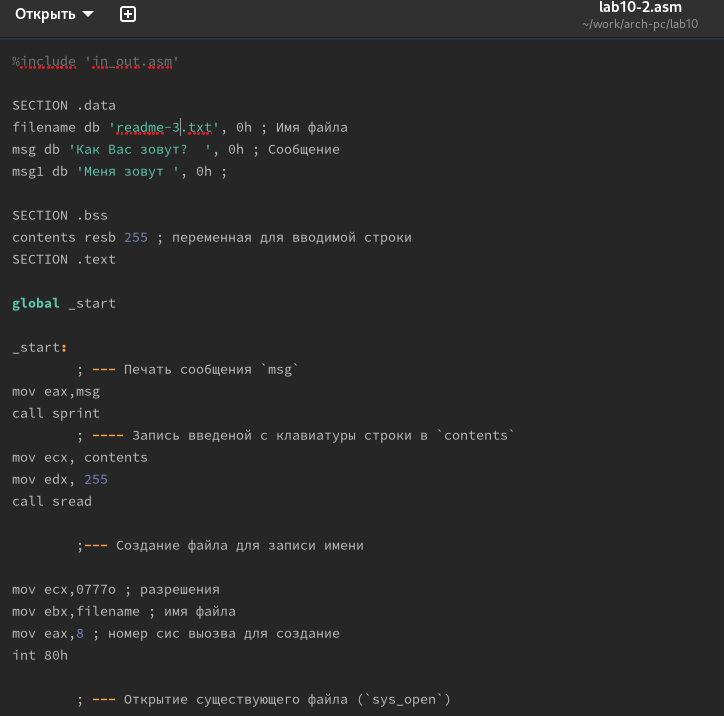


Figure 9: Написание программы для самостоятельной

**Шаг 3**

Создаю исполняемый файл и запускаю её, ввожу свое имя и фамилию. (рис. [[10](#fig:010)]).

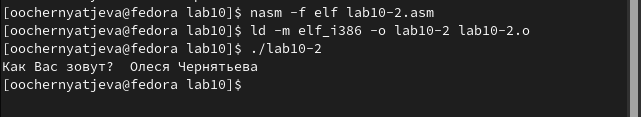


Figure 10: Исполнение файла

**Шаг 4**

Сразу проверяю с помощью атрибуты ***ls*** создался ли необходимый файл, куда должны записываться введенные символы. С помощью атрибуты ***cat*** проверяю содержимое созданного файла. (рис. [[11](#fig:011)]).

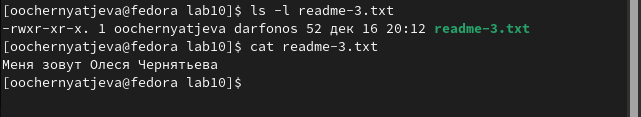


Figure 11: вставляю программу

**Программа отработала корректно!!**

**Текст программы в самостоятельной работе**

%include 'in\_out.asm'  
  
SECTION .data  
filename db 'readme-3.txt', 0h ; Имя файла  
msg db 'Как Вас зовут? ', 0h ; Сообщение  
msg1 db 'Меня зовут ', 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  
  
 ;--- Создание файла для записи имени  
   
mov ecx,0777o ; разрешения  
mov ebx,filename ; имя файла  
mov eax,8 ; номер сис выозва для создание  
int 80h  
  
 ; --- 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 msg1  
mov eax,msg1   
call slen   
  
 ; --- Записываем в файл `msg1` (`sys\_write`)  
mov edx, eax  
mov ecx, msg1  
mov ebx, esi  
mov eax, 4  
int 80h  
  
 ; --- 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

# 4 Вывод

В ходе выполениния работы были получены навыки по работе с файлами в NASM.