Отчет по лабораторной работе №10

дисциплина: Архитектура компьютера

Белоусова Елизавета Валентиновна

Содержание

**1 Цель работы**

Цель данной лабораторной работы - приобретение навыков написания программ для работы с файлами.

**2 Задание**

1. Выполнение лабораторной работы
2. Выполнение заданий для самостоятельной работы

**3 Теоретическое введение**

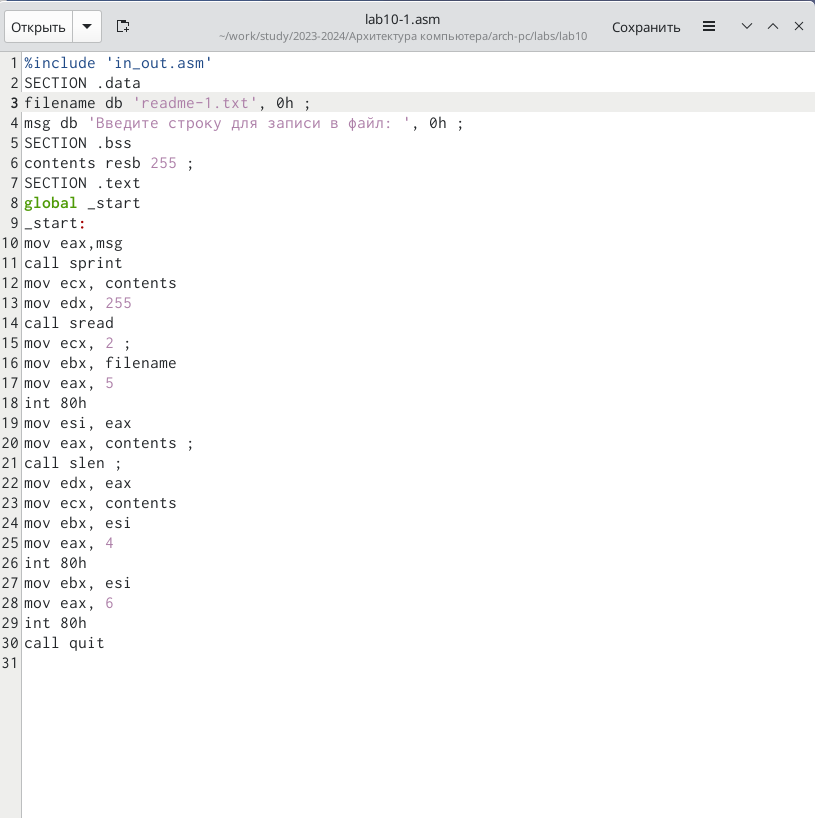
ОС GNU/Linux является многопользовательской операционной системой. И для обеспече- ния защиты данных одного пользователя от действий других пользователей существуют специальные механизмы разграничения доступа к файлам. Кроме ограничения доступа, дан- ный механизм позволяет разрешить другим пользователям 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] [:новая\_группа] или chgrp [ключи] < новая\_группа > 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 доступа. 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. Для создания и открытия файла служит системный вызов sys\_creat, который использует следующие аргументы: права доступа к файлу в регистре ECX, имя файла в EBX и номер системного вызова sys\_creat (8) в EAX. Для открытия существующего файла служит системный вызов sys\_open, который исполь- зует следующие аргументы: права доступа к файлу в регистре EDX, режим доступа к файлу в регистр ECX, имя файла в EBX и номер системного вызова sys\_open (5) в EAX. Для записи в файл служит системный вызов sys\_write, который использует следующие аргументы: количество байтов для записи в регистре EDX, строку содержимого для записи ECX, файловый дескриптор в EBX и номер системного вызова sys\_write (4) в EAX. Системный вызов возвращает фактическое количество записанных байтов в регистр EAX. В случае ошибки, код ошибки также будет находиться в регистре EAX. Прежде чем записывать в файл, его необходимо создать или открыть, что позволит полу- чить дескриптор файла. Для чтения данных из файла служит системный вызов sys\_read, который использует следующие аргументы: количество байтов для чтения в регистре EDX, адрес в памяти для записи прочитанных данных в ECX, файловый дескриптор в EBX и номер системного вызова sys\_read (3) в EAX. Как и для записи, прежде чем читать из файла, его необходимо открыть, что позволит получить дескриптор файла. Для правильного закрытия файла служит системный вызов sys\_close, который использует один аргумент – дескриптор файла в регистре EBX. После вызова ядра происходит удаление дескриптора файла, а в случае ошибки, системный вызов возвращает код ошибки в регистр EAX. Для изменения содержимого файла служит системный вызов sys\_lseek, который исполь- зует следующие аргументы: исходная позиция для смещения EDX, значение смещения в байтах в ECX, файловый дескриптор в EBX и номер системного вызова sys\_lseek (19) в EAX. Удаление файла осуществляется системным вызовом sys\_unlink, который использует один аргумент – имя файла в регистре EBX.

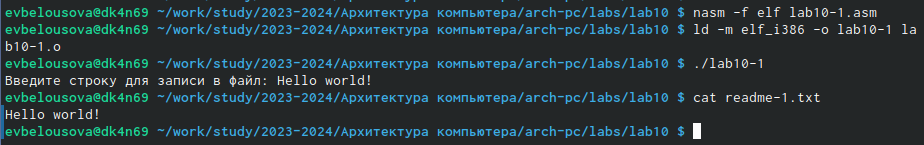
**4 Выполнение лабораторной работы**

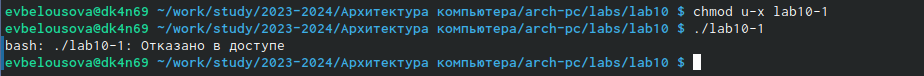
1. Выполнение лаборатороной работы

Перехожу в каталог, созданный для файлов с программами для лабораторной работы №10 (рис.1)  
1  
Рис. 1: Перемещение между директориями

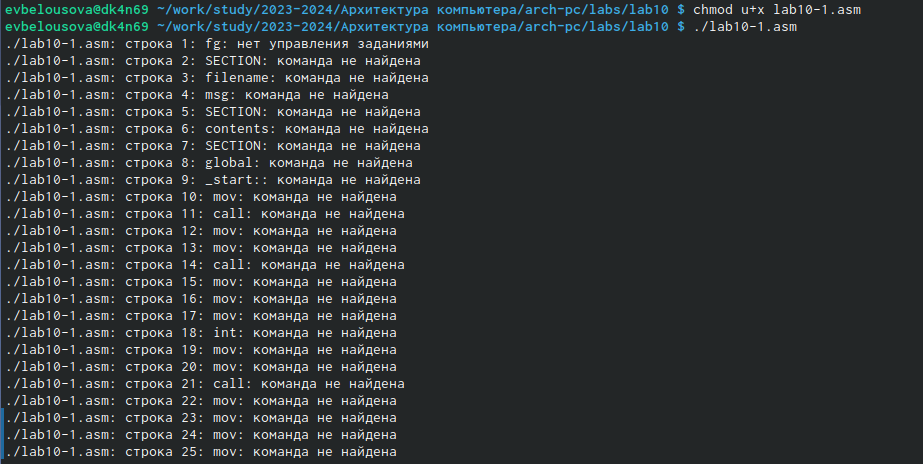
С помощью команды touch создаю файлы lab10-1.asm, readme-1.txt и readme-2.txt. Копирую в текущий каталог файл in\_out.asm, так как он будет использоваться в дальнейшем (рис. 2).  
2  
Рис. 1: Создание файлов

Открываю созданный файл lab10-1.asm, вставляю в него програм- му из листинга 10.1 (рис. 3)  
  
Рис. 3: Редактирование файла

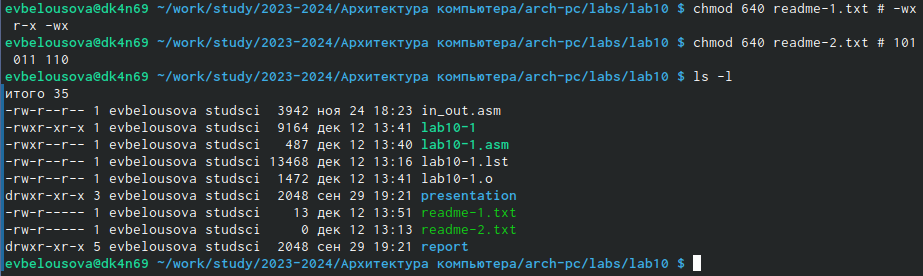
Создаю исполняемый файл и запускаю его (рис. 3).  
  
Рис. 4: Запуск исполняемого файла

С помощью команды chmod изменяю права доступа к исполняемому файлу lab10-1, запрещая его выполнение (рис. 5).  
  
Рис. 5: Изменение прав доступа к файлу

Файл не выполняется, так как я запретила доступ к нему, указав в команде “-х”.

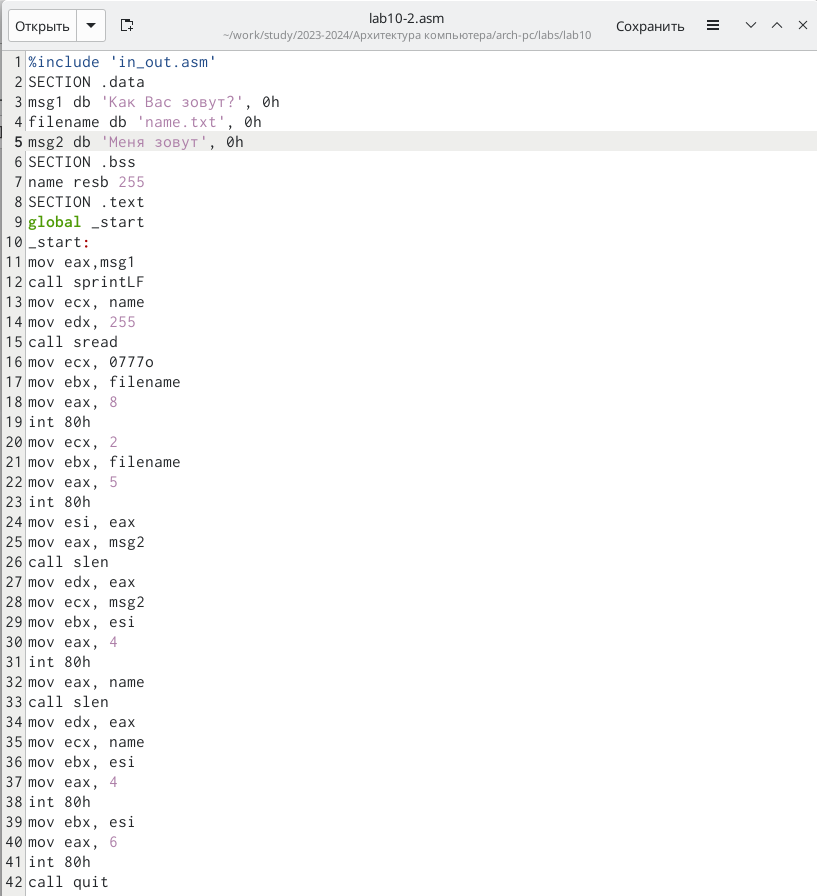
С помощью команды chmod изменяю права доступа к файлу lab10-1.asm с исходным текстом программы, добавив права на исполнение (рис. 6).  
  
Рис. 6:Изменение прав доступа к файлу

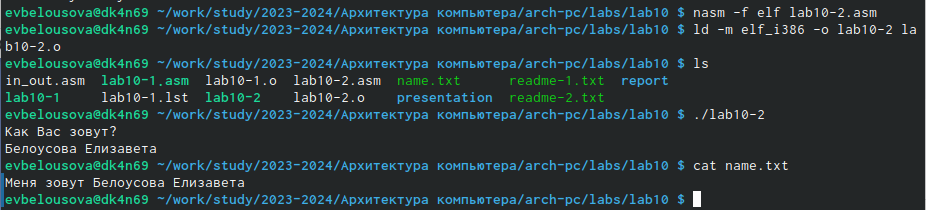
Файл начинает исполняться, однако он не содержит в себе команд для терминала, поэтому не может быть исполнен.

В соответствии со своим вариантом (18) 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 (рис.7)  
  
Рис. 7: Предоставление прав доступа к файлам

1. Выполнение заданий для самостоятельной работы

Создаю файлы lab10-2.asm и name.txt (рис. 8)  
8  
Рис. 8: Создание файлов

Открываю созданный файл и пишу в нем программу, позволяющую вывести на экран вопрос “Как Вас зовут”, ввести с клавиатуры свое имя, а затем создающую файл, который содержит сообщение “Меня зовут ФИ” (рис. 9).  
  
Рис. 9: Редактирование файла

Создаю исполняемый файл и запускаю его. Вижу, что программа работает корректно. (рис. 10)  
  
Рис. 10: Запуск программы

Листинг 10.2 (написанная программа)  
%include ‘in\_out.asm’  
SECTION .data  
msg1 db ‘Как Вас зовут?’, 0h  
filename db ‘name.txt’, 0h  
msg2 db ‘Меня зовут’, 0h  
SECTION .bss  
name resb 255  
SECTION .text  
global \_start  
\_start:  
mov eax,msg1  
call sprintLF  
mov ecx, name  
mov edx, 255  
call sread  
mov ecx, 0777o  
mov ebx, filename  
mov eax, 8  
int 80h  
mov ecx, 2  
mov ebx, filename  
mov eax, 5  
int 80h  
mov esi, eax  
mov eax, msg2  
call slen  
mov edx, eax  
mov ecx, msg2  
mov ebx, esi  
mov eax, 4  
int 80h  
mov eax, name  
call slen  
mov edx, eax  
mov ecx, name  
mov ebx, esi  
mov eax, 4  
int 80h  
mov ebx, esi  
mov eax, 6  
int 80h  
call quit

**5 Выводы**

При выполнении данной лабораторной работы, я приобрела практические навыки написания программ для работы с файлами.

**6 Список литературы**

[Архитектура ЭВМ](https://esystem.rudn.ru/pluginfile.php/2089097/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)