Отчёт по лабораторной работе №10

Дисциплина: Архитектура компьютера

Луангсуваннавонг Сайпхачан

Содержание

Список иллюстраций

# 1 Цель работы

Целью данной лабораторной работы является 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.Для обеспечения 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.

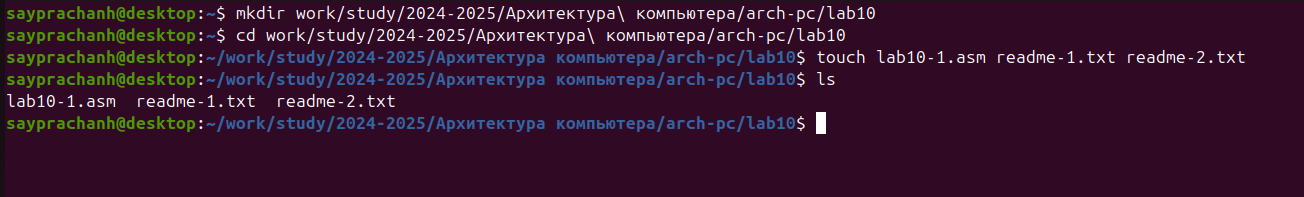
Права доступа определяют набор действий (чтение,запись,выполнение),разрешённых для выполнения пользователям системы над файлами.Для каждого файла пользователь может входить в одну из трех групп: владелец, член группы владельца, все остальные. Для каждой из этих групп может быть установлен свой набор прав доступа.Владельцем файла является его создатель.

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 отсутствии крытии или создании,файлу присваивается уникальный номер (16-битное целое число) дескриптор файла.

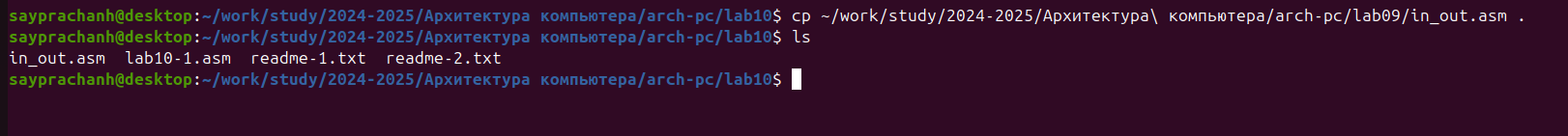
# 4 Выполнение лабораторной работы

Я создаю новый каталог, в котором я создам файлы с программами для лабораторной работы № 10, используя команду mkdir. Затем, используя команду touch, я создаю несколько файлов, таких как lab10-1.asm, readme-1.txt и readme-2.txt (Рис .4.1)



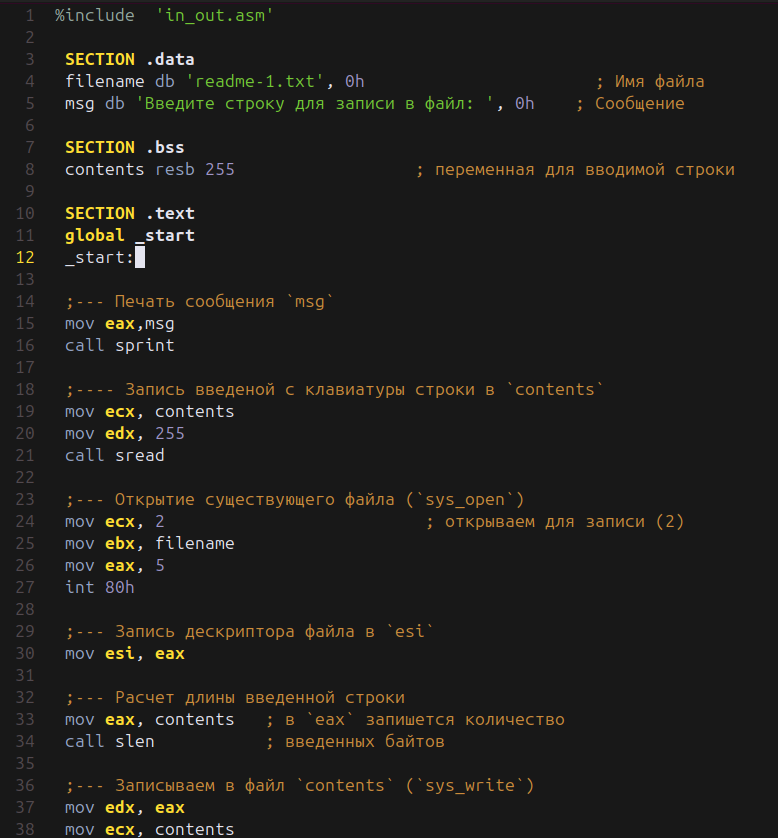
Создание файлов

Я копирую файл in\_out.asm из последней лабораторной работы, так как он будет использоваться с другой программой.(Рис .4.2)



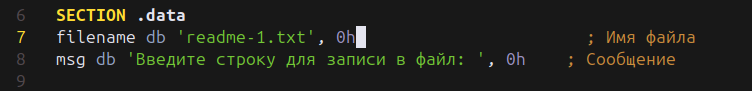
Копирование файла

Я открываю созданный файл lab10-1.asm и вставляю программу, которая будет записывать данные в файл сообщения, которым в данном случае является файл .txt.(Рис .4.3)



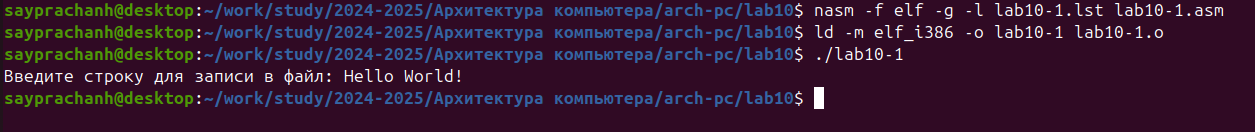
Редактирование файла

В исходном коде программы указано, что входные данные будут записываться в readme.txt.(Рис .4.4) Однако я не хочу создавать новый текстовый файл, я решил вместо этого заставить программу записывать входные данные в файл readme-1.txt.



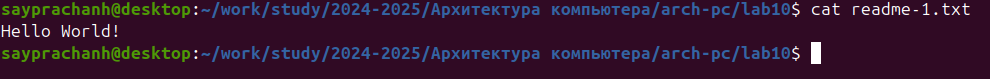
Редактирование файла

Я создаю исполняемый файл и запускаю его. Я ввожу в программу текстовую строку “Hello World!”(Рис .4.5)



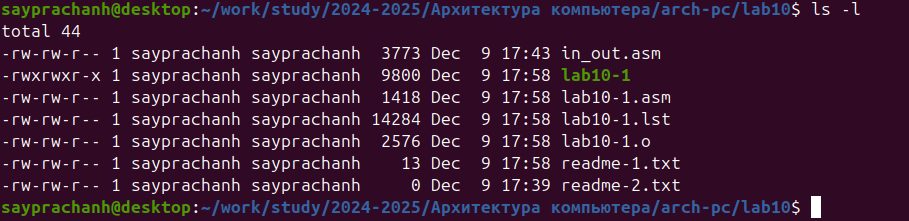
Запуск исполняемого файла

Используя команду cat, я отображаю содержимое текстового файла readme-1.txt.(Рис .4.6) Как мы видим, в нем отображаются данные, которые я ввел ранее. Это означает, что программа работает правильно.



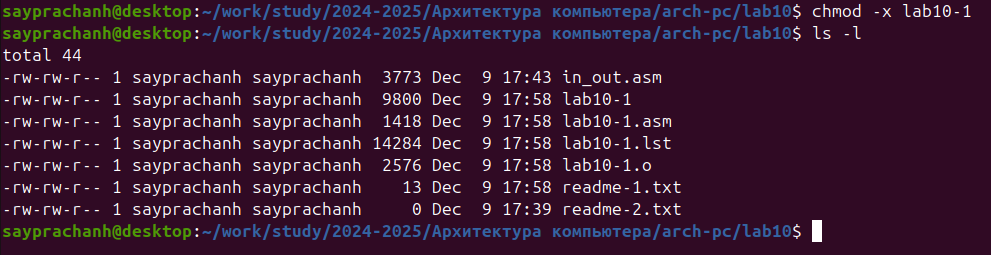
Чтение содержимого файла

Используя утилиту ls и клавишу -l, я перечисляю информацию о файлах в каталоге “lab10”. Эта информация включает имена файлов, время и дату их создания, а также права доступа к файлам, с которыми мы будем работать дальше.(Рис .4.7)



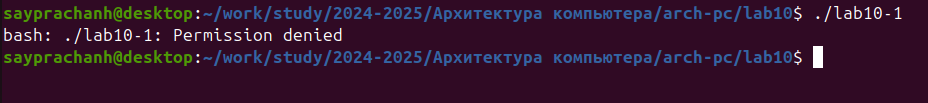
Список информации о файлах

Используя команду chmod, я изменяю права доступа к файлу исполняемого файла lab10-1, добавляя -x к команде chmod, я удаляю разрешение на выполнение файла. Используя команду ls -l, я проверяю работу команды. В результате в исполняемом файле lab10-1 в части прав доступа к файлу отсутствует символ “x”, что означает, что программа не может быть запущена.(Рис .4.8)



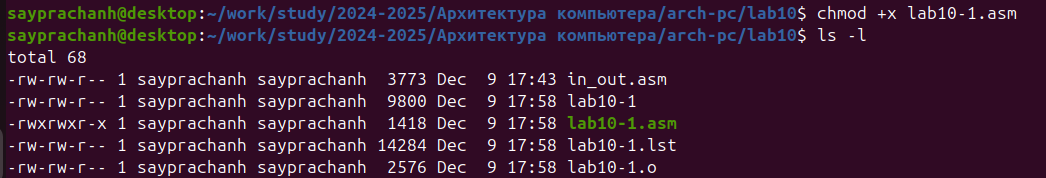
Изменение разрешения доступа к файлу

Когда я пытаюсь запустить файл, он выдает ошибку ‘В разрешении отказано’.(Рис .4.9) Это означает, что его невозможно выполнить. Поскольку, когда я менял разрешение на доступ к файлу с помощью команды chmod, я не указывал, у кого я буду снимать разрешение. По умолчанию команда удалит разрешение на выполнение у всех, включая владельца, группу и других.



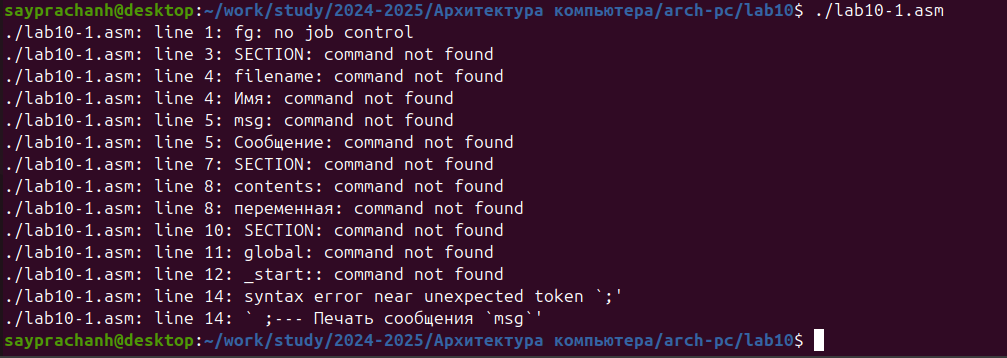
Запуск исполняемого файла

Используя команду chmod +x, я добавляю разрешение на выполнение в файл lab10-1.asm, затем проверяю работу команды, используя команду ls -l.(Рис .4.10)



Изменение разрешения доступа к файлу

Я пытаюсь запустить файл lab10-1.asm, он выдает ошибку(Рис .4.11). Поскольку файл lab10-1.asm не является исполняемым файлом, это файл исходного кода (написанный на ассемблере). Следовательно, он не может быть выполнен напрямую, и когда вы попытаетесь его выполнить, он выдаст ошибку, и программа ничего не сделает.



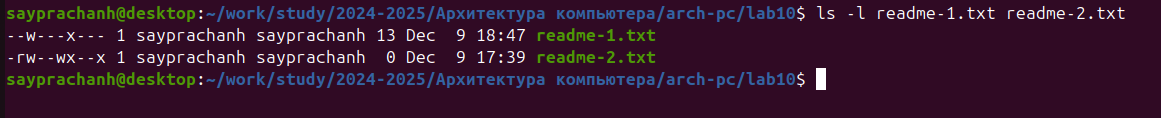
Запуск файла

Используя команду chmod, я установил соответствующие права доступа к файлам readme-1.txt и readme-2.txt.(Рис .4.12) Согласно моему варианту (вариант 13), для файла readme-1.txt будет в символьном виде (-w- –x —), а для файла readme-2.txt - в двоичной системе (110 011 001) Для readme-2.txt Я преобразую двоичную систему счисления в восьмеричную, в этом случае соответствующие права доступа к файлам будут равны 631 или в буквенном (rw- -wx –x)

Изменение разрешения доступа к файлу

Изменение разрешения доступа к файлу

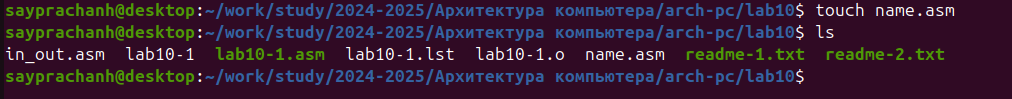
Затем, используя команду ls -l, я проверяю работу выполненной команды(Рис .4.13). Работа выполнена правильно.



Список информации о файлах

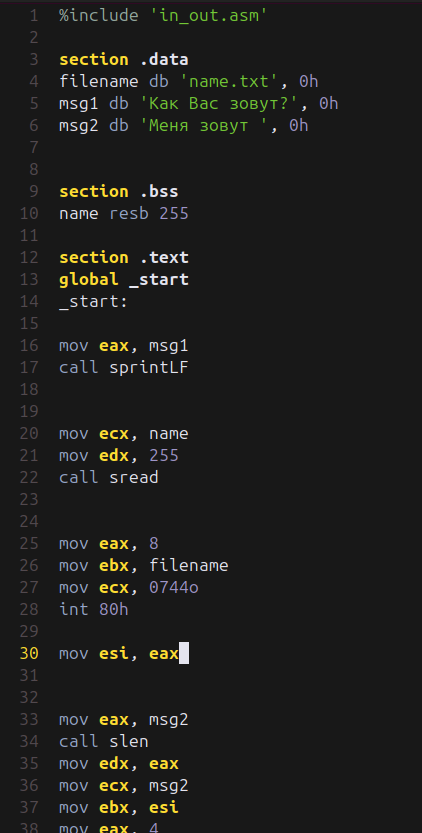
# 5 Выполнение заданий для самостоятельной работы

Используя команду “touch”, я создаю файлы name.asm(Рис .5.1)



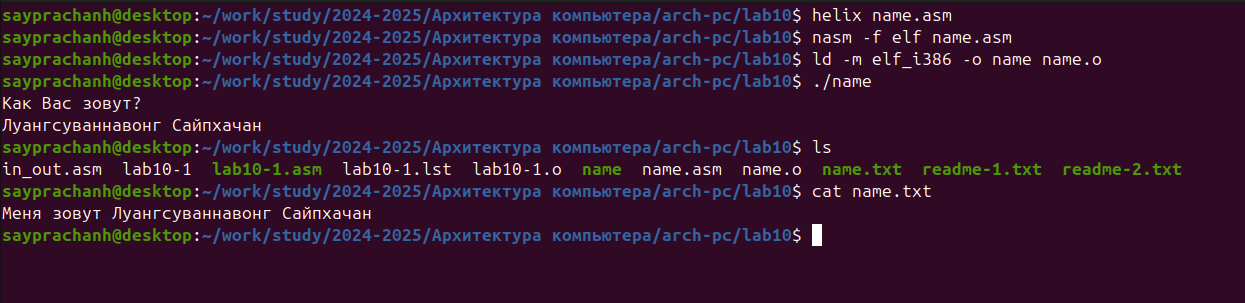
Создание файла

Я вхожу в программу, которая будет работать в соответствии с алгоритмом: выводит приглашение “Как вас зовут?”, запрашивает ввод данных, создает name.txt файл, записывает сообщение “Меня зовут” в name.txt файл, записывает вводимые данные в файл, затем закрывает файл.(Рис .5.2)



Редактирование файла

Я создаю исполняемый файл и запускаю его(Рис .5.3). Я ввожу свою фамилию и свое имя в программу. Я проверяю работу программы, сначала используя команду ls, мы видим, что name.txt был создан в каталоге. Затем, используя команду cat, я отображаю содержимое файла name.txt, и действительно, в нем есть мое имя и текст сообщения.



Запуск исполняемого файла

**Программа для выполнения задачи**

%include 'in\_out.asm'  
  
section .data  
filename db 'name.txt', 0h  
msg1 db 'Как Вас зовут?', 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 eax, 8  
mov ebx, filename  
mov ecx, 0744o  
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

# 6 Выводы

При выполнении данной лабораторной работы, Я приобрел навыки написания написания программ для работы с файлами.

# 7 Ответы на вопросы для самопроверки

**1. Каким образом в Unix-подобных ОС определяются права доступа к файлу?**

В Unix-подобных ОС права делятся на три категории: владельца, группы и остальных пользователей. Каждая категория может иметь права на чтение (r), запись (w) и выполнение (x). Эти права отображаются как строка, например, rwxr-xr--.

**2. Как ОС определяет,является ли файл исполняемым?Как регулировать права на чтение и запись?**

Файл исполним, если у пользователя есть право на выполнение (x). Чтобы изменить права на чтение или запись, используется команда chmod (например, chmod +r для чтения, chmod +w для записи).

**3. Как разграничить права доступа для различных категорий пользователей?**

Права доступа можно настроить с помощью команд:

chmod — для изменения прав на файл,

chown — для изменения владельца файла,

chgrp — для изменения группы файла.

**4. Какой номер имеют системные вызовы sys\_read, sys\_write, sys\_open, sys\_close,sys\_creat.**

Номера системных вызовов:

sys\_read: 0

sys\_write: 1

sys\_open: 2

sys\_close: 3

sys\_creat: 8 (обычно используется через sys\_open с флагом O\_CREAT).

**5. Какие регистры и как используют системные вызовы sys\_read,sys\_write,sys\_open,sys\_close,sys\_creat.**

Для системных вызовов используются следующие регистры:

sys\_read/sys\_write: EBX (дескриптор файла), ECX (буфер), EDX (количество байт).

sys\_open/sys\_creat: EBX (имя файла), ECX (флаги), EDX (режим).

sys\_close: EBX (дескриптор файла).

**6. Что такое дескриптор файла?**

Дескриптор файла — это число, которое процесс использует для работы с файлом. Он помогает операционной системе отслеживать открытые файлы и управлять операциями с ними, такими как чтение, запись и закрытие.

# 8 Список литературы

[Архитектура ЭВМ](https://esystem.rudn.ru/pluginfile.php/2089097/mod_resource/content/0/Лабораторная%20работа%20№10.%20Работа%20с%20файлами%20средствами%20Nasm.pdf)