Отчёт по лабораторной работе №10

Дисциплина: архитектура компьютера

Кайнова Алина Андреевна

Содержание

# 1 Цель работы

Приобрести навыки написания программ для работы с файлами.

# 2 Задание

1. Написание программ для работы с файлами
2. Выполнение заданий для самостоятельной работы

# 3 Теоретическое введение

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. Для изменения прав доступа служит команда chmod, которая понимает как символьное, так и числовое указание прав. Обработка файлов в операционной системе Linux осуществляется за счет ис- пользования определенных системных вызовов. Для корректной работы и досту- па к файлу при его открытии или создании, файлу присваивается уникальный номер (16-битное целое число) – дескриптор файла. Для создания и открытия файла служит системный вызов syscreat, который использует следующие аргументы: права доступа к файлу в регистре ECX, имя файла в EBX и номер системного вызова sys\_creat в EAX. Для открытия существующего файла служит системный вызов sysopen, который использует 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 syswrite, 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 получить дескриптор файла. Для чтения данных из файла служит системный вызов sysread, 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 syslseek, который использует 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. Значение смещения можно задавать в байтах. Удаление файла осуществляется системным вызовом sys\_unlink, который использует один аргумент – имя файла в регистре EBX.

# 4 Выполнение лабораторной работы

## 4.1 Написание программ для работы с файлами

Создаю каталог для данной лабораторной работы и уже в нём создаю файлы lab10-1.asm, readme-1.txt, readme-2.txt

Figure 1: Создание файлов для лабораторной работы

Figure 1: Создание файлов для лабораторной работы

Ввожу в файл lab10-1.asm код программы из листинга 10.1

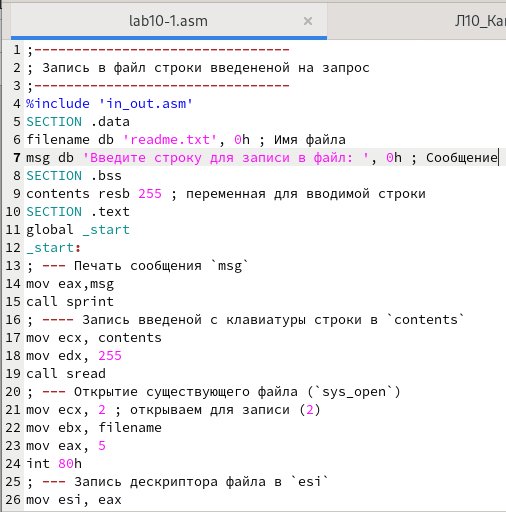


Figure 2: Ввод текста программы

Создаю исполняемый файл и проверяю его работу

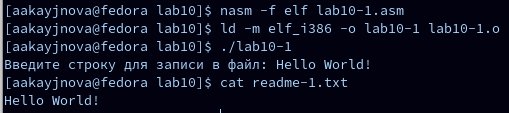


Figure 3: Запуск исполняемого фалйла

Изменяю права доступа к исполняемому файлу lab10-1, запретив его выполнение и проверяю правильность запрета

Figure 4: Запрет на выполнение файла

Figure 4: Запрет на выполнение файла

Файл не выполняется, так как в команде указали “u” - то есть владельца(себя) и “-” - отменить набор прав, “x” - право на исполнение.

Изменяю права доступа к данному файлу с исходным текстом программы, добавив права на исполнение и проверяю правильность добавления прав

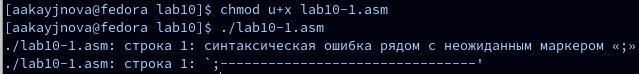


Figure 5: Добавление прав на исполнение

Текстовый файл начинает исполнение, но не исполняется, так как не содержит команд для терминала.

В соответствии со своим вариантом (5) в таблице 10.4 предоставляю права доступа к файлу readme-1.txt, представленные в символьном виде (–x -w- r-x), а для файла readme-2.txt - в двоичном виде (001 101 010) и проверяю правильность выполнения с помощью команды ls -l

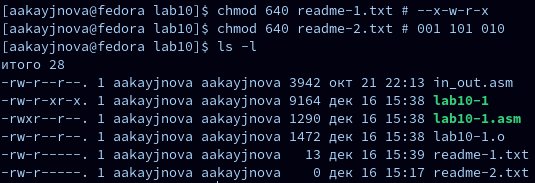


Figure 6: Предоставление прав доступа в символьном и двочином виде

## 4.2 Выполнение заданий для самостоятельной работы

Пишу код программы, выводящей предложение “Как Вас зовут?”, считывающей с клаватуры ФИ, создающую файл, в который записывается сообщение “Меня зовут ФИ”



Figure 7: Написание текста программы

Создаю исполняемый файл и проверяю его работу. Проверяю наличие файла и его содержимое с помощью команд ls и cat

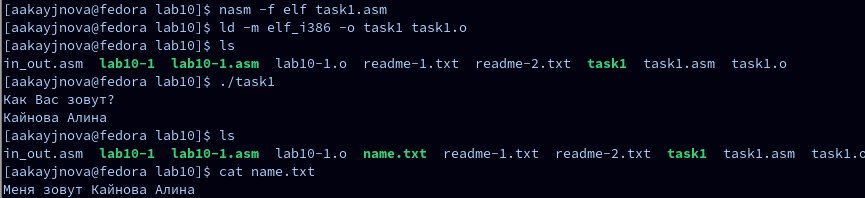


Figure 8: Запуск исполняемого файла

Программа работает корректно.

# 5 Выводы

В ходе данной лабораторной работы мы приобрели навыки написания программ для работы с файлами.

# Список литературы

1. 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