Работа с файлами средствами Nasm

Лабораторная работа №10

Губайдуллина Софья Романовна

Содержание

# 1 Цель работы

Приобретение навыков написания программ для работы с файлами.

# 2 Задание

1. Использование команд chmod для изменения прав доступа к файлу;
2. Предоставление прав доступа к файлам в символьном и двочном виде;
3. Выполнение самостоятельной работы.

# 3 Теоретическое введение

ОС GNU/Linux является многопользовательской операционной системой. И для обеспечения защиты данных одного пользователя от действий других пользователей существуют специальные механизмы разграничения доступа к файлам. Кроме ограничения доступа, данный механизм позволяет разрешить другим пользователям доступ данным для совместной работы.

Для предоставления прав доступа другому пользователю или другой группе командой chown [ключи] [:новая\_группа] или chgrp [ключи] < новая\_группа > .

Буква означает наличие права (установлен в единицу второй бит триады r — чтение, первый бит w — запись, нулевой бит х — исполнение), а дефис означает отсутствие права (нулевое значение соответствующего бита). Также права доступа могут быть представлены как восьмеричное число. Так, права доступа rw- (чтение и запись, без исполнения) понимаются как три двоичные цифры 110 или как восьмеричная цифра 6. Тип файла определяется первой позицией, это может быть: каталог — d, обычный файл — дефис (-) или символьная ссылка на другой файл — l. Следующие 3 набора по 3 символа определяют конкретные права для конкретных групп: r — разрешено чтение файла, w — разрешена запись в файл; x — разрешено исполнение файл и дефис (-) — право не дано. Для изменения прав доступа служит команда chmod, которая понимает как символьное, так и числовое указание прав.

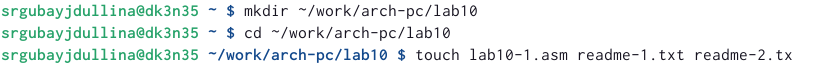
В операционной системе Linux существуют различные методы управления файлами, например, такие как создание и открытие файла, только для чтения или для чтения и записи, добавления в существующий файл, закрытия и удаления файла, предоставление прав доступа.

Общий алгоритм работы с системными вызовами в Nasm можно представить в следующем виде: 1. Поместить номер системного вызова в регистр EAX; 2. Поместить аргументы системного вызова в регистрах EBX, ECX и EDX; 3. Вызов прерывания (int 80h); 4. Результат обычно возвращается в регистр EAX.

* 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 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 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.
* Для правильного закрытия файла служит системный вызов sys\_close, который использует один аргумент – дескриптор файла в регистре EBX.
* Для изменения содержимого файла служит системный вызов sys\_lseek.

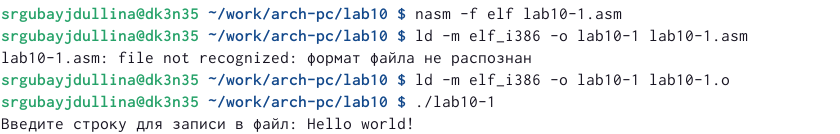
# 4 Выполнение лабораторной работы

1. Чтобы начать лабораторную работу, мне необходимо создать новые каталог и файл (рис. ??).



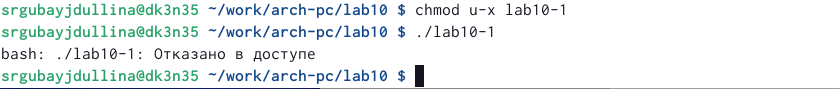
Создание новых каталога и файла

В новый созданный файл lab10-1.asm записываю листинг 10.1 (программу записи в файл сообщения), создаю исполняемый файл и проверяю работу файла (рис. ??).



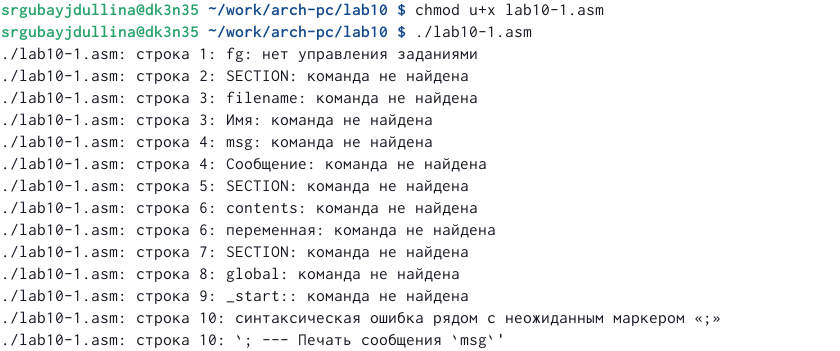
Создание исполняемого файла и проверка работы

С помощью команды chmod u-x изменяю права доступа к исполняемому файлу lab10-1, запретив его выполнение, а именно: как владельц (u) отменяю набор прав (“-”) на исполнение файла (x) (рис. ??).



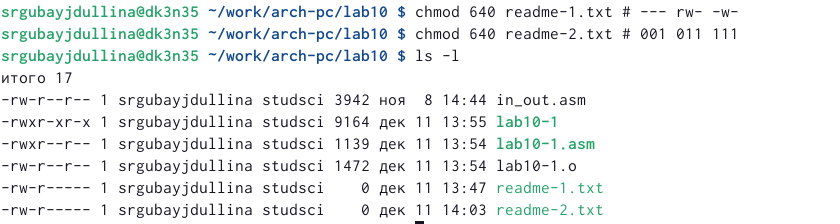
Запрет на исполнение файла при помощи команды chmod

Теперь с помощью команды chmod u+x меняю права доступа к файлу lab10-1.asm с исходным текстом программы, добавив права на исполнение (как владельц (u) добавляю набор прав (“+”) на исполнение файла (x) (рис. ??).



Добавляю право на исполнение при помощи комианды chmod

1. Далее мне требуется в соответствии с вариантом 20 в таблице 10.4 предоставить 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. ??).

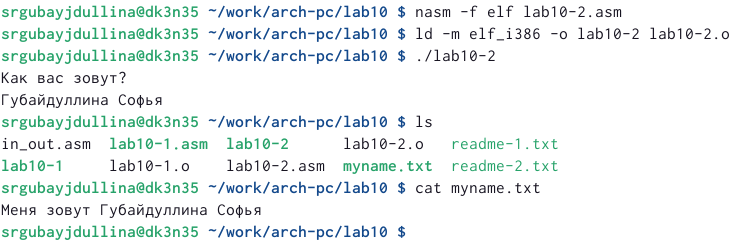


Представление прав доступа в символьном и двоичном виде

1. Для выполнения самостоятельной работы создаю новый файл для работы lab10-2.asm. В нём я пишу программу для вывода приглашения о том, как зовут пользователя, с созданием нового файла и записи в него имени из сообщения. Ниже привожу листинг моей программы:

%include ‘in\_out.asm’ SECTION .data filename db ‘myname.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,myname mov edx,255 callsread 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 ebx,esi mov eax,6 int 80h call quit

Создаю исполняемый файл и проверяю работу. При помощи ls и cat проверяю наличие файла и его содержимое (рис. ??)



Проверка работы программы для самостоятельной работы

# 5 Выводы

В ходе выполнения лабораторной работы я приобрела навыки написания программ для работы с файлами.