РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук Кафедра прикладной информатики и теории вероятностей

Архипов Олег Константинович

Содержание

# 1 Цель работы

Приобретение навыков написания программ для работы с файлами.

# 2 Выполнение лабораторной работы

Создаю каталог для программ ЛР10, перехожу в него и создаю там 3 файла (рис. [1](#fig:001)).

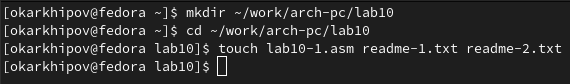


Figure 1: Папка и файлы ЛР10

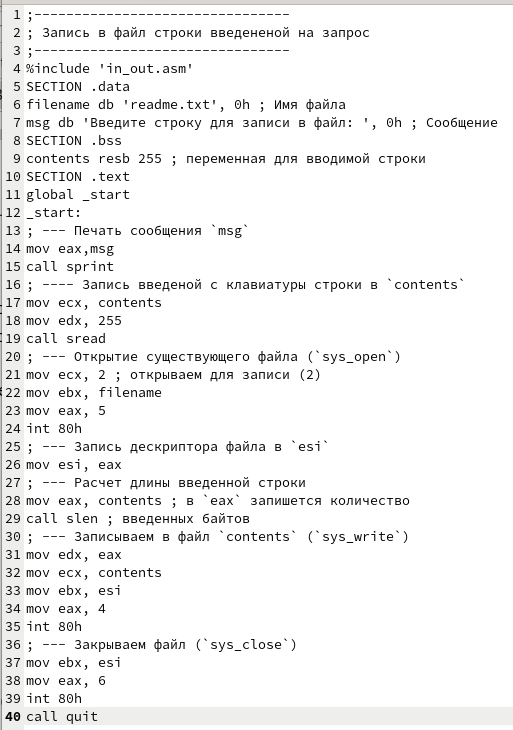


Figure 2: Текст программы записи в файл сообщения

Ввожу в файл lab10-1.asm текст программы (рис. [2](#fig:002)).

Создаю ещё один файл readme.asm , после чего запускаю процессы трансляции и компановки, получаю исполнительный файл main , запускаю его, на запрос ввода строки пишу ‘Hello world!’ , затем, при помощи команды ‘ls -l’ проверяю содержимое папки lab10 , а командой ‘cat readme.txt’ вывожу на экран содержимое файла readme.txt (рис. [3](#fig:003)).

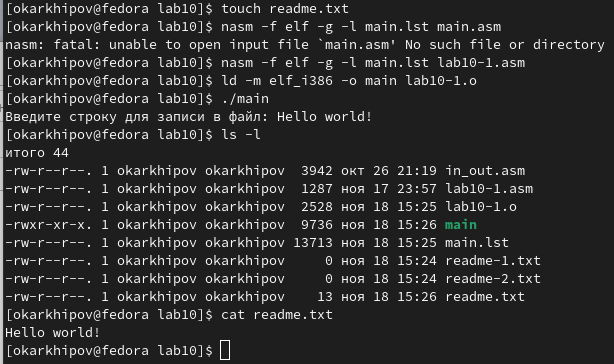


Figure 3: Создание исполняемого файла, его запуск и проверка результата

После этого переименовываю исполняемый файл main в lab10-1 и снова проверяю содержимое папки (рис. [4](#fig:004)).

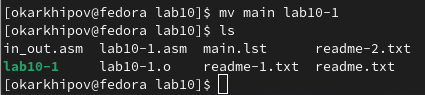


Figure 4: Новое имя исполняемого файла

Изменяю права доступа к файлу lab10-1 , запрещая его выполнение для всех видов пользователей (опция a-x). Попытка запуска не удается, т.к. только что я запретил данный процесс всем, включая себя (рис. [5](#fig:005)).

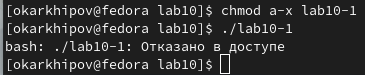


Figure 5: Запрет выполнения программы

Теперь предоставлю всем пользователям право запускать файл lab10-1.asm и попробую его выполнить, получаю отчет о синтаксической ошибке, связанной с наличием комментария в начале исходного текста программы (рис. [6](#fig:006)). Удаляю этот комментарий (рис. [7](#fig:007)), проверяю, как изменились права доступа (рис. [8](#fig:008)) и пробую запустить еще раз, но снова получаю ошибку (рис. [9](#fig:009)), т.к. файлы asm написаны не на языке программирования, а потому процессор не может распознать заложенные в программе команды без процедуры трансляции.

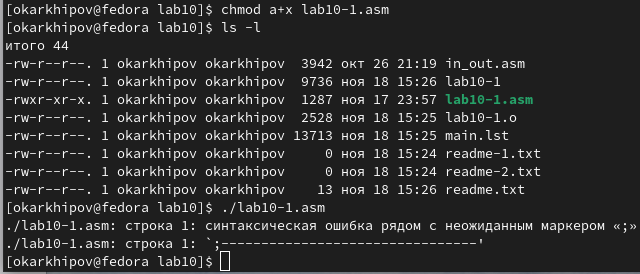


Figure 6: Добавление прав доступа в исходный текстовый файл

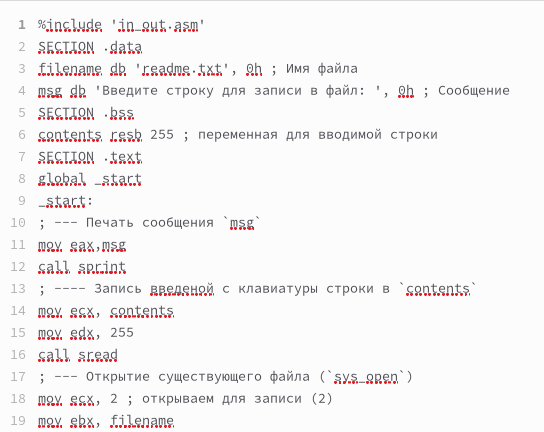


Figure 7: Изменение исходного текстового файла

Figure 8: Проверка изменений доступа к lab10-1.asm

Figure 8: Проверка изменений доступа к lab10-1.asm

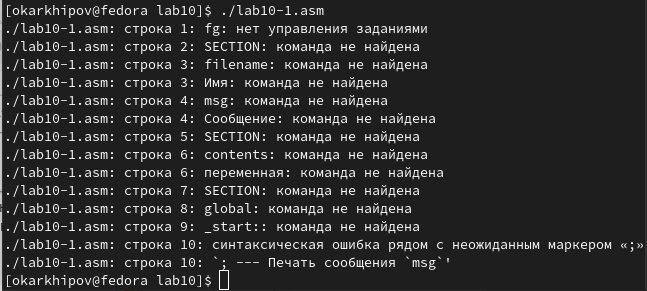


Figure 9: Повторная попытка запуска исходного текстового файла

Теперь необходимо в соответствии с моим вариантом в прошлых ЛР предоставить права доступа к файлу readme- 1.txt представленные в символьном виде, а для файла readme-2.txt – в двочном виде (рис. [10](#fig:010)).

Figure 10: Мой вариант

Figure 10: Мой вариант

Делаю это как показано на рис. [11](#fig:011): ‘u=w’ предоставляет владельцу право записи, ‘g=-’ - группу владельца лишает всех прав, а ‘0=w’ дает всем остальным пользователям также право записи, затем проверяю результат командой ‘ls -l’ .

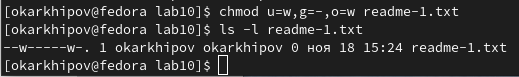


Figure 11: Изменение прав доступа к readme-1.txt

Чтобы изменить права доступа в соответствии с двоичным представлением вспомню, что 1 означает наличие одной из букв r , w , x , а 0 соответствует ‘-’, т.е. отсутствию доступа. Руководствуясь этим знанием, введу команду

chmod u=x,g=wx,o=rw readme-2.txt

и для проверки правильности

ls -l readme-2.txt

Результат можно увидеть на рис. [12](#fig:012).

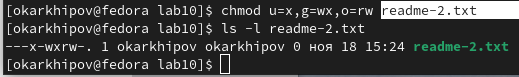


Figure 12: Изменение прав доступа к readme-2.txt

# 3 Задание для самостоятельной работы

Написать программу работающую по следующему алгоритму:

* Вывод приглашения “Как Вас зовут?”
* Ввести с клавиатуры свои фамилию и имя
* Создать файл с именем name.txt
* Записать в файл сообщение “Меня зовут”
* Дописать в файл строку введенную с клавиатуры
* Закрыть файл

Создаю файл для самостоятельной работы (рис. [13](#fig:013)).

Figure 13: Файл СР

Figure 13: Файл СР

Ввожу в новый файл следующий код:

%include 'in\_out.asm'  
SECTION .data  
filename db 'name.txt',0h ; имя файла  
msg1 db 'Как Вас зовут? ',0h ; сообщение 1  
msg2 db 'Меня зовут ',0h ; сообщение 2  
SECTION .bss  
cont resb 255 ; переменная для вводимой строки  
SECTION .text  
global \_start  
\_start:  
mov eax, msg1  
call sprint ; вывод сообщения msg1  
mov ecx,cont  
  
mov ecx, cont  
mov edx, 255  
call sread ; запись введенной с клавиатуры строки  
  
; создание файла name.txt (sys\_creat)  
mov ecx, 0777o ; установка прав доступа  
mov ebx, filename ; имя созданного файла  
mov eax, 8 ; код системного вызова creat  
int 80h  
  
; запись дискриптора файла в 'esi'  
mov esi, eax  
  
; расчет длины msg2  
mov eax, msg2  
call slen  
mov edi, eax ; запись длины в 'edi'  
  
; запись в файл сообщения msg2 (sys\_write)  
mov edx, edi ; запись длины msg2 в байтах  
mov ecx, msg2 ; сообщение  
mov ebx, esi ; дискриптор  
mov eax, 4 ; код системного вызова  
int 80h  
  
; расчет длины введенной строки  
mov eax, cont  
call slen  
  
; запись в файл введенной строки (sys\_write)  
mov edx, eax ; запись длины cont в байтах  
mov ecx, cont ; сообщение  
mov ebx, esi ; дискриптор  
mov eax, 4 ; код системного вызова  
int 80h  
  
; закрыть файл (sys\_close)  
mov ebx, esi ; дискриптор  
mov eax, 6 ; код системного вызова  
int 80h  
  
call quit

Создаю исполняемый файл, запускаю его, ввожу свои фамилию и имя (рис. [14](#fig:014)).

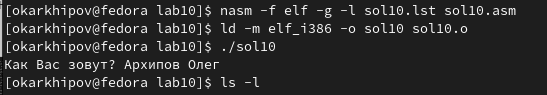


Figure 14: Запуск программы sol10

Далее командой ‘ls -l’ проверяю создавшиеся файлы: ‘name.txt’ , ‘sol10.lst’ , ‘sol10.o’ , ‘sol10’ . И наконец, командой ‘cat name.txt’ считываю содержимое файла ‘name.txt’ (рис. [15](#fig:015)). Делаю вывод, что код был корректен.

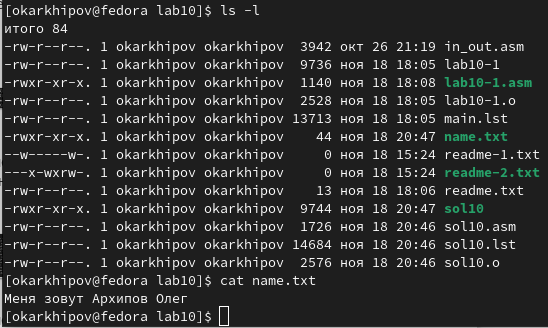


Figure 15: Результат работы программы sol10

# 4 Выводы

Были приобретены навыки работы с файлами при помощи NASM.

# Список литературы