

Лабораторная работа №10

Работа с файлами средствами Nasm

Коровкин Никита Михайлович

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	12

Список иллюстраций

2.1	Создание рабочей папки и файлов	6
2.2	Копирование файла	6
2.3	Вставляем код из листинга	7
2.4	Запуск кода	7
2.5	Изменение прав доступа	7
2.6	Добавляем права на запуск	8
2.7	Выдача прав доступа	8
2.8	Создание второго файла	9
2.9	Код для самостоятельной работы	9
2.10	Код для самостоятельной работы	10
2.11	Код для самостоятельной работы	10
2.12	Запуск и проверка кода	11

Список таблиц

1 Цель работы

Приобрести навыки работы с файлами в Ассемблере, а также разобраться с понятием прав доступа в linux

2 Выполнение лабораторной работы

Для начала нам необходимо создать рабочую папку и файл lab10-1.asm, а также файлы readme-1.txt и readme-2.txt(рис.1)

```
liveuser@localhost-live:~$ mkdir ~/work/arch-pc/lab10
liveuser@localhost-live:~$ cd ~/work/arch-pc/lab10
liveuser@localhost-live:~/work/arch-pc/lab10$ touch lab10-1.asm readme-1.txt readme-2.txt
liveuser@localhost-live:~/work/arch-pc/lab10$
```

Рис. 2.1: Создание рабочей папки и файлов

Скопируем файл in_out.asm из папки прошлой работы.(рис.2)

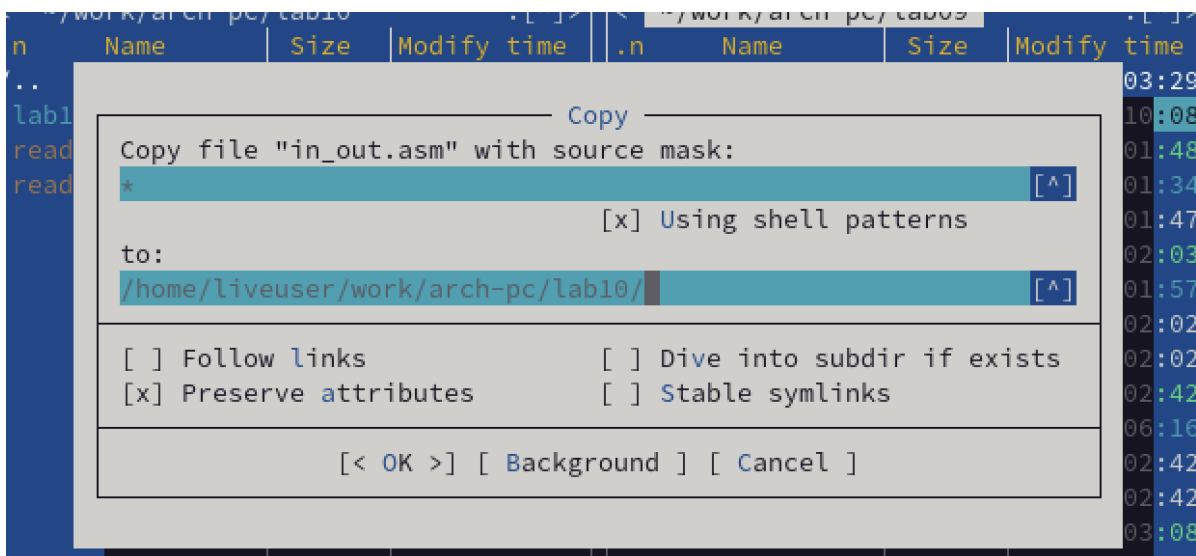


Рис. 2.2: Копирование файла

Теперь вставим в файл lab10-1.asm код из листинга 10.1(рис.3)

```

mov ecx, 2 ; открываем для записи (2)
mov ebx, filename
mov eax, 5
int 80h
; --- Запись дескриптора файла в `esi`
mov esi, eax
; --- Расчет длины введенной строки
mov eax, contents ; в `eax` запишется количество
call slen ; введенных байтов
; --- Записываем в файл `contents` (`sys_write`)
mov edx, eax
mov ecx, contents
mov ebx, esi

```

Рис. 2.3: Вставляем код из листинга

Запустим программу и посмотрим на результат.(рис.4)

```

liveuser@localhost-live:~$ cd ~/work/arch-pc/lab10
liveuser@localhost-live:~/work/arch-pc/lab10$ nasm -f elf lab10-1.asm
liveuser@localhost-live:~/work/arch-pc/lab10$ ld -m elf_i386 -o lab10-1 lab10-1.o
liveuser@localhost-live:~/work/arch-pc/lab10$ ./lab10-1
Введите строку для записи в файл: hi

```

Рис. 2.4: Запуск кода

Как видно из вывода, файл выполнен, но ничего не произошло, так как в коде прописано записать данные в файл readme.txt, которого не существует.

Тогда попробуем изменить права доступа для программы lab10-1 так, чтобы запретить всем группам пользователей запускать эту программу и попробуем запустить ее еще раз.(рис.5)

```

liveuser@localhost-live:~/work/arch-pc/lab10$ chmod ugo-x lab10-1
liveuser@localhost-live:~/work/arch-pc/lab10$ ./lab10-1
bash: ./lab10-1: Permission denied
liveuser@localhost-live:~/work/arch-pc/lab10$

```

Рис. 2.5: Изменение прав доступа

После изменения прав доступа мы попробовали вновь запустить программу, и нам отказали в доступе.

Теперь попробуем добавить к файлу с кодом lab10-1.asm права на запуск, и попробуем его запустить.(рис.6)

```
liveuser@localhost-live:~/work/arch-pc/lab10$ chmod ugo+x lab10-1.asm
liveuser@localhost-live:~/work/arch-pc/lab10$ ./lab10-1.asm
./lab10-1.asm: line 1: fg: no job control
./lab10-1.asm: line 2: SECTION: command not found
./lab10-1.asm: line 3: filename: command not found
./lab10-1.asm: line 3: Имя: command not found
./lab10-1.asm: line 4: msg: command not found
./lab10-1.asm: line 4: Сообщение: command not found
./lab10-1.asm: line 5: SECTION: command not found
./lab10-1.asm: line 6: contents: command not found
./lab10-1.asm: line 6: переменная: command not found
./lab10-1.asm: line 7: SECTION: command not found
./lab10-1.asm: line 8: global: command not found
./lab10-1.asm: line 9: _start:: command not found
./lab10-1.asm: line 10: syntax error near unexpected token `;'
./lab10-1.asm: line 10: `; --- Печать сообщения `msg`'
```

Рис. 2.6: Добавляем права на запуск

Терминал вывел много ошибок, так как файл не предназначен для запуска.

Теперь попробуем выдать права доступа для файлов readme-1.txt и readme-2.txt согласно варианту 16. Так, мы используем chmod и пишем права доступа в восьмиричном виде.(рис.7)

```
liveuser@localhost-live:~/work/arch-pc/lab10$ chmod 152 readme-1.txt
liveuser@localhost-live:~/work/arch-pc/lab10$ chmod 125 readme-2.txt
liveuser@localhost-live:~/work/arch-pc/lab10$ ls -l
total 24
-rw-r--r--. 1 liveuser liveuser 3942 Nov  7 10:08 in_out.asm
-rw-r--r--. 1 liveuser liveuser 9164 Dec  9 09:04 lab10-1
-rwxr-xr-x. 1 liveuser liveuser 1140 Dec  9 09:00 lab10-1.asm
-rw-r--r--. 1 liveuser liveuser 1472 Dec  9 09:04 lab10-1.o
---xr-x-w-. 1 liveuser liveuser   0 Dec  9 03:36 readme-1.txt
---x-w-r-x. 1 liveuser liveuser   0 Dec  9 03:36 readme-2.txt
liveuser@localhost-live:~/work/arch-pc/lab10$
```

Рис. 2.7: Выдача прав доступа

Права доступа установлены корректно.

#Выполнение самостоятельной работы

Для начала создадим файл с которым будем работать.(рис.8)

```
liveuser@localhost-live:~/work/arch-pc/lab10$ touch lab10-2.asm
```

Рис. 2.8: Создание второго файла

Теперь напишем код, который должен создать файл name.txt, записать туда фразу “Меня зовут”, запросить Фамилию и Имя пользователя и дописать их в файл.(рис.9-11)

```
%include 'in_out.asm'
SECTION .data
filename db 'name.txt', 0h ; Имя файла
msg db 'Как вас зовут? ', 0h ; Сообщение
name db 'Меня зовут ', 0h
SECTION .bss
sentence resb 255 ; переменная для вводимой строки
SECTION .text
global _start

_start:
mov eax,msg ; переносим в регистр eax
call sprint ; выводим сообщение
mov ecx, sentence ; переменная для вводимой строки в регистре
mov edx, 255 ; выделяем память
call sread ; считываем

mov ecx, 0777o ; создание файла
mov ebx, filename
```

Рис. 2.9: Код для самостоятельной работы

```

mov ecx, 0777o ; создание файла
mov ebx, filename
mov eax, 8 ; записываем дескриптор файла
int 80h ; вызываем ядро

mov edx, 255 ;
mov ecx, name ; адрес строки для записи в файл
mov ebx, eax ; переносим дескриптор файла
mov eax, 4 ;
int 80h ; вызываем ядро

mov eax, 5

mov edx, 2 ; значение смещения -- конец файла
mov ecx, 0 ; смещение на 0 байт
mov ebx, eax ; дескриптор файла
mov eax, 19 ; номер системного вызова `sys_lseek`
int 80h ; вызов ядра

```

Рис. 2.10: Код для самостоятельной работы

```

mov edx, 2 ; значение смещения -- конец файла
mov ecx, 0 ; смещение на 0 байт
mov ebx, eax ; дескриптор файла
mov eax, 19 ; номер системного вызова `sys_lseek`
int 80h ; вызов ядра
mov edx, 9 ; Запись в конец файла
mov ecx, sentence ; строки из переменной `msg`
mov eax, 4
int 80h

mov ebx, esi
mov eax, 6
int 80h
call quit

```

Рис. 2.11: Код для самостоятельной работы

Теперь запустим код, проверим, создан ли файл, и то, что в него записа-

лось.(рис.12)

```
liveuser@localhost-live:~/work/arch-pc/lab10$ nasm -f elf lab10-2.asm
liveuser@localhost-live:~/work/arch-pc/lab10$ ld -m elf_i386 -o lab10-2 lab10-2.o
liveuser@localhost-live:~/work/arch-pc/lab10$ ./lab10-2
Как вас зовут? Nikita Korovkin
liveuser@localhost-live:~/work/arch-pc/lab10$ ls
in_out.asm  lab10-2      lab10-2.o  readme-1.txt
lab10-1.asm lab10-2.asm  name.txt   readme-2.txt
liveuser@localhost-live:~/work/arch-pc/lab10$ cat name.txt
Меня зовут Nikita Korovkin
liveuser@localhost-live:~/work/arch-pc/lab10$
```

Рис. 2.12: Запуск и проверка кода

Все работает верно. Работа выполнена правильно.

3 Выводы

В ходе выполнения данной лабораторной работы были получены навыки работы с файлами в Ассемблере, а также знания о правах доступа файлов.