

Отчёт по лабораторной работе №10

Лобанова Екатерина Евгеньевна

Содержание

1 Цель работы	5
2 Выполнение лабораторной работы	6
2.1 Задание для самостоятельной работы	9
3 Выводы	12

Список иллюстраций

2.1	Создаем каталог с помощью команды mkdir и файлы с помощью команды touch	6
2.2	Заполняем файл	7
2.3	Запускаем файл и проверяем его работу	7
2.4	Используем команду chmod для установки нужных прав, после этого пытаемся запустить файл	8
2.5	Используем команду chmod для установки нужных прав, после этого пытаемся запустить файл	8
2.6	Используем команду chmod для установки нужных прав, после этого проверяем правильность выполнения командой ls -l	9
2.7	Создаем файл командой touch	9
2.8	Пишем программу в midnight commander	10
2.9	Проверяем работу программы	11

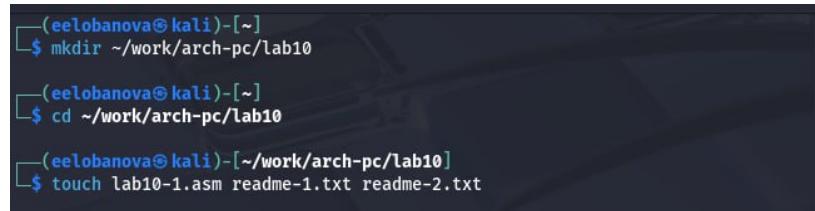
Список таблиц

1 Цель работы

Приобрести навыки написания программ для работы с файлам, научиться управлять доступом к файлам.

2 Выполнение лабораторной работы

Создаем каталог для программ ЛБ10, и в нем создаем файлы (рис. 2.1).



```
(eelobanova@kali)-[~]
└─$ mkdir ~/work/arch-pc/lab10
(eelobanova@kali)-[~]
└─$ cd ~/work/arch-pc/lab10
(eelobanova@kali)-[~/work/arch-pc/lab10]
└─$ touch lab10-1.asm readme-1.txt readme-2.txt
```

Рисунок 2.1: Создаем каталог с помощью команды `mkdir` и файлы с помощью команды `touch`

Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 10.1 (рис. 2.2).

```
GNU nano 8.4                                lab10-1.asm *
%include 'in_out.asm'
SECTION .data
filename db 'readme.txt', 0h ; Имя файла
msg db 'Введите строку для записи в файл: ', 0h ; Сообщение
SECTION .bss
contents resb 255 ; переменная для вводимой строки
SECTION .text
global _start
_start:
; --- Печать сообщения `msg`
mov eax, msg
call sprint
; ---- Запись введенной с клавиатуры строки в `contents`
mov ecx, contents
mov edx, 255
call sread
; --- Открытие существующего файла (`sys_open`)
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
mov eax, 4
int 80h
; --- Закрываем файл (`sys_close`)
mov ebx, esi
mov eax, 6
int 80h
call quit
```

Рисунок 2.2: Заполняем файл

Создаем исполняемый файл и запускаем его (рис. 2.3).

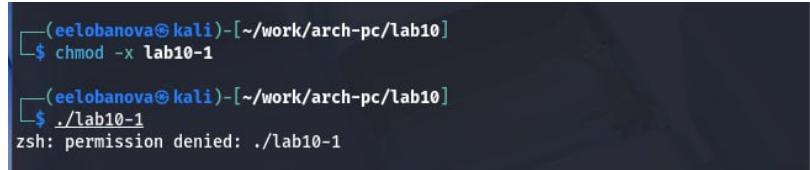
```
(eelobanova㉿kali)-[~/work/arch-pc/lab10]
└─$ nasm -f elf -l lab10-1.lst lab10-1.asm

(eelobanova㉿kali)-[~/work/arch-pc/lab10]
└─$ ld -m elf_i386 -o lab10-1 lab10-1.o

(eelobanova㉿kali)-[~/work/arch-pc/lab10]
└─$ ./lab10-1
Введите строку для записи в файл: Hello world!
```

Рисунок 2.3: Запускаем файл и проверяем его работу

Изменяем права доступа к файлу, запретив его выполнение. Пробуем запустить файл (рис. 2.4).



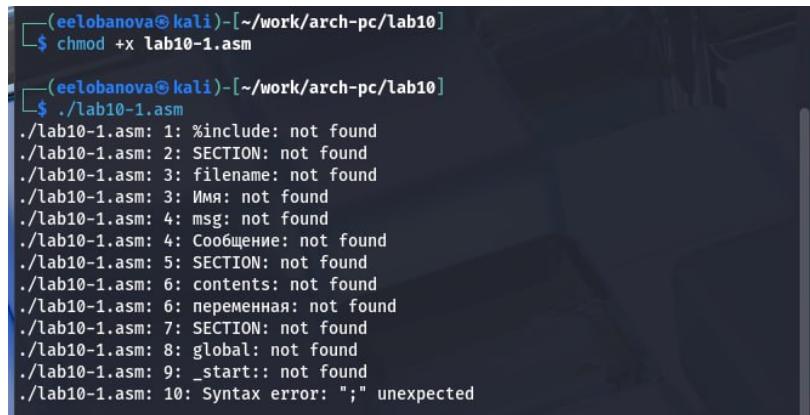
```
(eelobanova㉿kali)-[~/work/arch-pc/lab10]
└─$ chmod -x lab10-1

(eelobanova㉿kali)-[~/work/arch-pc/lab10]
└─$ ./lab10-1
zsh: permission denied: ./lab10-1
```

Рисунок 2.4: Используем команду chmod для установки нужных прав, после этого пытаемся запустить файл

Выдало: отказано в доступе. Значит мы поставили правильный запрет на выполнение.

Изменяем права доступа к файлу с исходным текстом программы, добавив права на исполнение. Пробуем запустить файл (рис. 2.5).



```
(eelobanova㉿kali)-[~/work/arch-pc/lab10]
└─$ chmod +x lab10-1.asm

(eelobanova㉿kali)-[~/work/arch-pc/lab10]
└─$ ./lab10-1.asm
./lab10-1.asm: 1: %include: not found
./lab10-1.asm: 2: SECTION: not found
./lab10-1.asm: 3: filename: not found
./lab10-1.asm: 3: Имя: not found
./lab10-1.asm: 4: msg: not found
./lab10-1.asm: 4: Сообщение: not found
./lab10-1.asm: 5: SECTION: not found
./lab10-1.asm: 6: contents: not found
./lab10-1.asm: 6: переменная: not found
./lab10-1.asm: 7: SECTION: not found
./lab10-1.asm: 8: global: not found
./lab10-1.asm: 9: _start:: not found
./lab10-1.asm: 10: Syntax error: ";" unexpected
```

Рисунок 2.5: Используем команду chmod для установки нужных прав, после этого пытаемся запустить файл

lab10-1.asm является файлом с исходным кодом программы на языке ассемблера, искусственно добавление права на исполнение не даст ожидаемого результата. Такие файлы нужно компилировать или ассемблировать в машинный код, а затем выполнять.

ВАРИАНТ 17

Предоставляем права доступа к 2ум файлам, согласно варианту 17 в символьном и двоичном виде, затем проверяем работу команд. (рис. 2.6).

```
(eelobanova㉿kali)-[~/work/arch-pc/lab10]
└─$ chmod u=rwx,g=rx,o=rw readme-1.txt

(eelobanova㉿kali)-[~/work/arch-pc/lab10]
└─$ chmod 536 readme-2.txt

(eelobanova㉿kali)-[~/work/arch-pc/lab10]
└─$ ls -l
total 52
-rw-rw-r-- 1 eelobanova eelobanova 3942 Oct 28 15:43 in_out.asm
-rw-rw-r-- 1 eelobanova eelobanova 9164 Nov 25 19:40 lab10-1
-rwxrwxr-x 1 eelobanova eelobanova 1140 Nov 25 19:30 lab10-1.asm
-rw-rw-r-- 1 eelobanova eelobanova 13446 Nov 25 19:40 lab10-1.lst
-rw-rw-r-- 1 eelobanova eelobanova 1472 Nov 25 19:40 lab10-1.o
-rwxrwxr-x 1 eelobanova eelobanova 9164 Nov 25 19:35 lab10-3
-r-x-wxrw- 1 eelobanova eelobanova 0 Nov 25 19:26 readme-1.txt
-r-x-wxrw- 1 eelobanova eelobanova 0 Nov 25 19:26 readme-2.txt
```

Рисунок 2.6: Используем команду chmod для установки нужных прав, после этого проверяем правильность выполнения командой ls -l

2.1 Задание для самостоятельной работы

Создаем новый файл (рис. 2.7).

```
(eelobanova㉿kali)-[~/work/arch-pc/lab10]
└─$ touch lab10-2.asm
```

Рисунок 2.7: Создаем файл командой touch

Пишем программу, которая выполнит представленный список действий (рис. 2.8).

```
GNU nano 8.4          lab10-2.asm *
%include 'in_out.asm'

SECTION .data
    msg: DB 'Как вас зовут? ',0
    filename: DB 'name.txt',0
    message: DB 'Меня зовут ',0

SECTION .bss
    name: RESB 80

SECTION .text
    global _start

_start:
    mov eax,msg
    call sprint
    mov ecx, name
    mov edx,80
    call sread
    mov ecx,07770
    mov ebx,filename
    mov eax,8
    int 80h
    mov esi,eax
    mov eax,message
    call slen
    mov edx,eax
    mov ecx,message
    mov ebx,esi
    mov eax,4
    int 80h
    mov ebx,esi
    mov eax,6
    int 80h
    mov ecx,1
    mov ebx,filename
    mov eax,5
    int 80h
    mov esi,eax
    mov edx,2
    mov ecx,0
    mov ebx,eax
    mov eax,19
    int 80h
    mov eax,name
    call slen
    mov edx,eax
    mov ecx,name
    mov ebx,esi
```

Рисунок 2.8: Пишем программу в midnight commander

Создаем исполняемый файл и запускаем его, после этого проверяем создался ли новый файл, затем смотрим, как он заполнен (рис. 2.9).

```
(eelobanova㉿kali)-[~/work/arch-pc/lab10]
└─$ nano lab10-2.asm

(eelobanova㉿kali)-[~/work/arch-pc/lab10]
└─$ nasm -f elf -g -l lab10-2.lst lab10-2.asm

(eelobanova㉿kali)-[~/work/arch-pc/lab10]
└─$ ld -m elf_i386 -o lab10-2 lab10-2.o

(eelobanova㉿kali)-[~/work/arch-pc/lab10]
└─$ ./lab10-2
Как вас зовут? Екатерина

(eelobanova㉿kali)-[~/work/arch-pc/lab10]
└─$ ls
in_out.asm  lab10-1.lst  lab10-2.asm  lab10-3      readme-2.txt
lab10-1     lab10-1.o   lab10-2.lst  name.txt
lab10-1.asm lab10-2     lab10-2.o   readme-1.txt

(eelobanova㉿kali)-[~/work/arch-pc/lab10]
└─$ cat name.txt
Меня зовут Екатерина
```

Рисунок 2.9: Проверяем работу программы

3 Выводы

Мы научились писать программы для работы с файлами и научились предоставлять права доступа к ним.