

Отчёт по лабораторной работе №10

Архитектура компьютеров и операционные системы.

Брыляков Никита Евгеньевич

Содержание

| | | |
|----------|--|-----------|
| 1 | Цель работы | 4 |
| 2 | Задание | 5 |
| 3 | Теоретическое введение | 6 |
| 4 | Выполнение лабораторной работы | 8 |
| 4.1 | Выполнение лабораторной работы. | 8 |
| 4.2 | Задания для самостоятельной работы | 11 |
| 5 | Вывод | 14 |
| 6 | Список литературы | 15 |

Список иллюстраций

| | | |
|-----|--|----|
| 4.1 | Создание каталога и файла внутри | 8 |
| 4.2 | Ввод программы | 9 |
| 4.3 | Создание и запуск | 9 |
| 4.4 | Проверка файла | 10 |
| 4.5 | Ввод команды | 10 |
| 4.6 | Добавление прав на исполнение | 10 |
| 4.7 | Изменение прав | 11 |
| 4.8 | Ввод программы | 12 |
| 4.9 | Создание и запуск | 13 |

1 Цель работы

Приобретение навыков написания программ для работы с файлами.

2 Задание

1. Выполнение лабораторной работы.
2. Задание 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 использования определенных системных вызовов. Для корректной работы и доступа 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 `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`. 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 `sys_read`, который использует следующие аргументы: количество байтов для чтения в регистре EDI, 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.

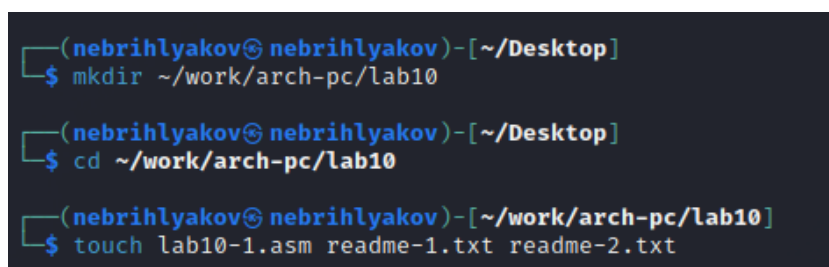
Для изменения содержимого файла служит системный вызов `sys_lseek`, который использует следующие аргументы: исходная позиция для смещения EDI, значение смещения в байтах в ECX, файловый дескриптор в EBX и номер системного вызова `sys_lseek` (19) в EAX. Значение смещения можно задавать в байтах.

Удаление файла осуществляется системным вызовом `sys_unlink`, который использует один аргумент – имя файла в регистре EBX.

4 Выполнение лабораторной работы

4.1 Выполнение лабораторной работы.

Создаю каталог для программ лабораторной работы № 10, перехожу в него и создаю файлы lab10-1.asm, readme-1.txt и readme-2.txt. (рис. [4.1]).



```
(nebrihlyakov@nebrihlyakov)~[~/Desktop]
$ mkdir ~/work/arch-pc/lab10

(nebrihlyakov@nebrihlyakov)~[~/Desktop]
$ cd ~/work/arch-pc/lab10

(nebrihlyakov@nebrihlyakov)~[~/work/arch-pc/lab10]
$ touch lab10-1.asm readme-1.txt readme-2.txt
```

Рис. 4.1: Создание каталога и файла внутри

Ввожу в файл lab10-1.asm текст программы, записывающей в файл сообщения, из листинга 10.1. (рис. [4.2]).


```

1 ;-----
2 ; Запись в файл строки введенной на запрос
3 ;-----
4 %include 'in_out.asm'
5 SECTION .data
6 filename db 'readme.txt', 0h ; Имя файла
7 msg db 'Введите строку для записи в файл: ', 0h ; Сообщение
8 SECTION .bss
9 contents resb 255 ; переменная для вводимой строки
10 SECTION .text
11 global _start
12 _start:
13 ; — Печать сообщения `msg`
14 mov eax,msg
15 call sprint
16 ; — Запись введенной с клавиатуры строки в `contents`
17 mov ecx, contents
18 mov edx, 255
19 call sread
20 ; — Открытие существующего файла (`sys_open`)
21 mov ecx, 2 ; открываем для записи (2)
22 mov ebx, filename
23 mov eax, 5
24 int 80h
25 ; — Запись дескриптора файла в `esi`
26 mov esi, eax
27 ; — Расчет длины введенной строки
28 mov eax, contents ; в `eax` запишется количество
29 call slen ; введенных байтов
30 ; — Записываем в файл `contents` (`sys_write`)
31 mov edx, eax
32 mov ecx, contents
33 mov ebx, esi
34 mov eax, 4
35 int 80h
36 ; — Закрываем файл (`sys_close`)

```

Рис. 4.2: Ввод программы

Создаю исполняемый файл и проверяю его работу. (рис. [4.3]).

```

(nebrihlyakov@nebrihlyakov)-[~/work/arch-pc/lab10]
$ nasm -f elf lab10-1.asm

(nebrihlyakov@nebrihlyakov)-[~/work/arch-pc/lab10]
$ ld -m elf_i386 -o lab10-1 lab10-1.o

(nebrihlyakov@nebrihlyakov)-[~/work/arch-pc/lab10]
$ ./lab10-1
Введите строку для записи в файл: qwerty

```

Рис. 4.3: Создание и запуск

(рис. [4.4]).

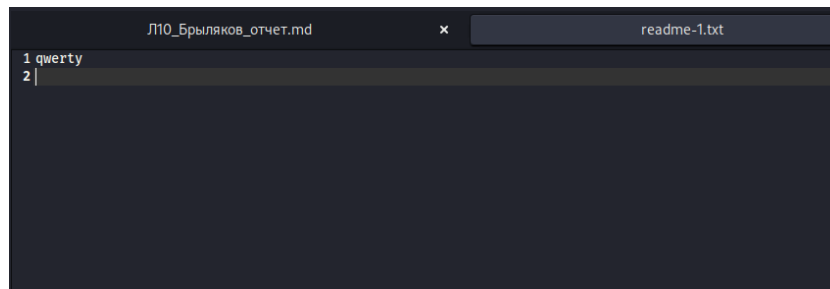


Рис. 4.4: Проверка файла

С помощью команды `chmod u-x` изменяю права доступа к исполняемому файлу `lab10-1`, запретив его выполнение и пытаюсь выполнить файл. (рис. [4.5]).

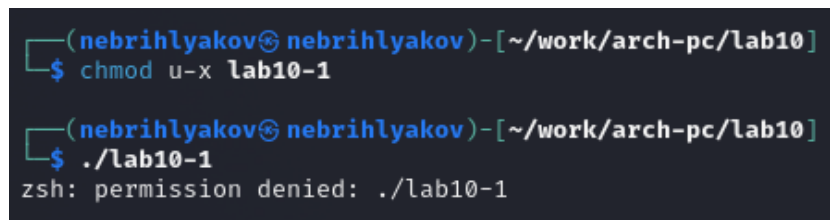


Рис. 4.5: Ввод команды

Файл не выполняется, так как в команде написано “u” - владелец, “-” - отменить набор прав, “x” - право на исполнение программы.

С помощью команды `chmod u+x` изменяю права доступа к файлу `lab10-1.asm` с исходным текстом программы, добавив права на исполнение, и выполняю его. (рис. [4.6]).

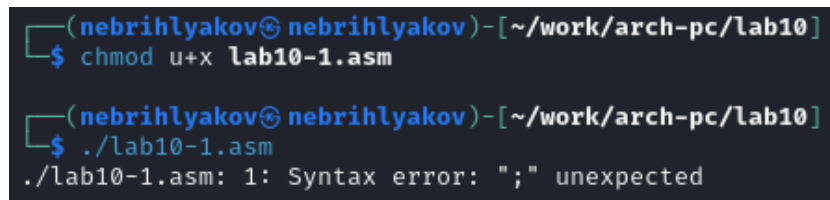


Рис. 4.6: Добавление прав на исполнение

Файл нельзя выполнить, так как он не содержит указаний для терминала, а просто является текстом программы.

Предоставляю права доступа к файлам в соответствии со своим вариантом (5). (рис. [4.7]).

```
(nebrihlyakov@nebrihlyakov)-[~/work/arch-pc/lab10]
$ chmod 125 readme-1.txt

(nebrihlyakov@nebrihlyakov)-[~/work/arch-pc/lab10]
$ chmod 152 readme-2.txt

(nebrihlyakov@nebrihlyakov)-[~/work/arch-pc/lab10]
$ ls -l
total 28
-rw-r--r-- 1 nebrihlyakov nebrihlyakov 3942 Nov 11 08:56 in_out.asm
-rw-r--xr-x 1 nebrihlyakov nebrihlyakov 9164 Dec 16 11:48 lab10-1
-rwxr--r-- 1 nebrihlyakov nebrihlyakov 1287 Dec 16 11:39 lab10-1.asm
-rw-r--r-- 1 nebrihlyakov nebrihlyakov 1472 Dec 16 11:48 lab10-1.o
--x-w-r-x 1 nebrihlyakov nebrihlyakov 7 Dec 16 11:50 readme-1.txt
--xr-x-w- 1 nebrihlyakov nebrihlyakov 0 Dec 16 11:36 readme-2.txt
```

Рис. 4.7: Изменение прав

4.2 Задания для самостоятельной работы

Пишу код программы по заданию. (рис. [4.8]).

```

1 %include 'in_out.asm'
2 SECTION .data
3 msg1 db 'Как Вас зовут?', 0h
4 filename db 'name.txt', 0h
5 msg2 db 'Меня зовут ', 0h
6 SECTION .bss
7 name resb 255
8 SECTION .text
9 global _start
10 _start:
11 mov eax, msg1
12 call sprintf
13 mov ecx, name
14 mov edx, 255
15 call sread
16 mov ecx, 0777o
17 mov ebx, filename
18 mov eax, 8
19 int 80h
20 mov ecx, 2
21 mov ebx, filename
22 mov eax, 5
23 int 80h
24 mov esi, eax
25 mov eax, msg2
26 call slen
27 mov edx, eax
28 mov ecx, msg2
29 mov ebx, esi
30 mov eax, 4
31 int 80h
32 mov eax, name
33 call slen
34 mov edx, eax
35 mov ecx, name
36 mov ebx, esi
37 mov eax, 4
38 int 80h
39 mov ebx, esi
40 mov eax, 6
41 int 80h
42 call quit
43

```

Рис. 4.8: Ввод программы

Создаю исполняемый файл и проверяю его работу. Программа работает корректно (рис. [4.9]).

```
(nebrihlyakov@nebrihlyakov)-[~/work/arch-pc/lab10]
$ nasm -f elf zadanie.asm

(nebrihlyakov@nebrihlyakov)-[~/work/arch-pc/lab10]
$ ld -m elf_i386 -o zadanie zadanie.o

(nebrihlyakov@nebrihlyakov)-[~/work/arch-pc/lab10]
$ ./zadanie
Как Вас зовут?
Брыляков Никита

(nebrihlyakov@nebrihlyakov)-[~/work/arch-pc/lab10]
$ cat name.txt
Меня зовут Брыляков Никита
```

Рис. 4.9: Создание и запуск

5 Вывод

При выполнении данной лабораторной работы я приобрёл навыки написания программ для работы с файлами.

6 Список литературы

- [illegible]