

# **Отчёт лабораторной работы №10**

**Дисциплина: архитектура компьютера**

Худдыева Дженнет

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>9</b>
4.1	Написание программ для работы с файлами . . . . .	9
4.2	Задание для самостоятельной работы . . . . .	12
<b>5</b>	<b>Выводы</b>	<b>17</b>

## Список иллюстраций

4.1	Создание файлов для лабораторной работы . . . . .	9
4.2	Ввод текста программы из листинга 10.1 . . . . .	10
4.3	Запуск исполняемого файла . . . . .	10
4.4	Запрет на выполнение файла . . . . .	10
4.5	Добавление прав на исполнение . . . . .	11
4.6	Предоставление прав доступа в символьном и двоичном виде . .	11
4.7	Написание текста программы . . . . .	12
4.8	Запуск исполняемого файла и проверка его работы . . . . .	13

## **Список таблиц**

# **1 Цель работы**

Приобретение навыков написания программ для работы с файлами.

## **2 Задание**

1. Написание программ для работы с файлами.
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 Написание программ для работы с файлами

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])



```
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/arch-pc/lab10$ touch lab10-1.asm
```

Рис. 4.1: Создание файлов для лабораторной работы

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

```

1 %include 'in_out.asm'
2 SECTION .data
3 filename db 'readme-1.txt', 0h ; Имя файла
4 msg db 'Введите строку для записи в файл: ', 0h ; Сообщение
5 SECTION .bss
6 contents resb 255 ; переменная для вводимой строки
7 SECTION .text
8 global _start
9 _start:
10 ; --- Печать сообщения msg
11 mov eax,msg
12 call sprint
13 ; ---- Запись введенной с клавиатуры строки в contents
14 mov ecx, contents
15 mov edx, 255
16 call sread
17 ; --- Открытие существующего файла (sys_open)
18 mov ecx, 2 ; открываем для записи (2)
19 mov ebx, filename
20 mov eax, 5
21 int 80h
22 ; --- Запись дескриптора файла в esi
23 mov esi, eax
24 ; --- Расчет длины введенной строки

```

Рис. 4.2: Ввод текста программы из листинга 10.1

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

```

dkhuddiheva@dkhuddiheva-VirtualBox:~/work/arch-pc/lab10$ nasm -f elf lab10-1.asm
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/arch-pc/lab10$ ld -m elf_i386 -o lab10-1 lab10-1.o
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/arch-pc/lab10$ ./lab10-1
Введите строку для записи в файл: Hello world!
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/arch-pc/lab10$ cat readme-1.txt
Hello world!
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/arch-pc/lab10$

```

Рис. 4.3: Запуск исполняемого файла

Далее с помощью команды `chmod u-x` изменяю права доступа к исполняемому файлу `lab10-1`, запретив его выполнение и пытаюсь выполнить файл. (рис. [4.4])

```

dkhuddiheva@dkhuddiheva-VirtualBox:~/work/arch-pc/lab10$ chmod u-x lab10-1
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/arch-pc/lab10$ ./lab10-1
bash: ./lab10-1: Отказано в доступе
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/arch-pc/lab10$

```

Рис. 4.4: Запрет на выполнение файла

Файл не выполняется, т.к в команде я указала “u” - владелец (себя), “-” - отменить набор прав, “x” - право на исполнение.

С помощью команды `chmod u+x` изменяю права доступа к файлу `lab10-1.asm` с исходным текстом программы, добавив права на исполнение, и пытаюсь выполнить его. (рис. [4.5])

```
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/arch-pc/lab10$ chmod u+x lab10-1.asm
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/arch-pc/lab10$ ./lab10-1.asm
./lab10-1.asm: строка 1: fg: нет управления заданиями
./lab10-1.asm: строка 2: SECTION: команда не найдена
./lab10-1.asm: строка 3: filename: команда не найдена
./lab10-1.asm: строка 3: Имя: команда не найдена
./lab10-1.asm: строка 4: msg: команда не найдена
./lab10-1.asm: строка 4: Сообщение: команда не найдена
./lab10-1.asm: строка 5: SECTION: команда не найдена
./lab10-1.asm: строка 6: contents: команда не найдена
./lab10-1.asm: строка 6: переменная: команда не найдена
./lab10-1.asm: строка 7: SECTION: команда не найдена
./lab10-1.asm: строка 8: global: команда не найдена
./lab10-1.asm: строка 9: _start:: команда не найдена
./lab10-1.asm: строка 10: синтаксическая ошибка рядом с неожиданным маркером «;»
./lab10-1.asm: строка 10: `; --- Печать сообщения msg'
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/arch-pc/lab10$
```

Рис. 4.5: Добавление прав на исполнение

Текстовый файл начинает исполнение, но не исполняется, т.к не содержит в себе команд для терминала.

В соответствии со своим вариантом (17) в таблице 10.4 предоставляю права доступа к файлу `readme1.txt` представленные в символьном виде, а для файла `readme-2.txt` – в двоичном виде:

`r-x -wx rw- 010 000 010`

И проверяю правильность выполнения с помощью команды `ls -l`. (рис. [4.6])

```
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/arch-pc/lab10$ chmod 536 readme-1.txt
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/arch-pc/lab10$ chmod 536 readme-2.txt
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/arch-pc/lab10$ ls -l
итого 52
-rw-rw-r-- 1 dkhuddiheva dkhuddiheva 3942 ноя 16 22:13 in_out.asm
-rwxrwxr-x 1 dkhuddiheva dkhuddiheva 9164 дек 10 15:40 lab10-1
-rwxrw-r-- 1 dkhuddiheva dkhuddiheva 1126 дек 10 15:39 lab10-1.asm
-rw-rw-r-- 1 dkhuddiheva dkhuddiheva 1472 дек 10 15:39 lab10-1.o
-rwxrwxr-x 1 dkhuddiheva dkhuddiheva 37 дек 14 22:28 name.txt
-r-x-wxrw- 1 dkhuddiheva dkhuddiheva 13 дек 10 15:40 readme-1.txt
-r-x-wxrw- 1 dkhuddiheva dkhuddiheva 0 дек 10 04:41 readme-2.txt
-rwxrwxr-x 1 dkhuddiheva dkhuddiheva 9156 дек 14 22:28 task1
-rw-rw-r-- 1 dkhuddiheva dkhuddiheva 598 дек 14 22:23 task1.asm
-rw-rw-r-- 1 dkhuddiheva dkhuddiheva 1536 дек 14 22:27 task1.o
```

Рис. 4.6: Предоставление прав доступа в символьном и двоичном виде

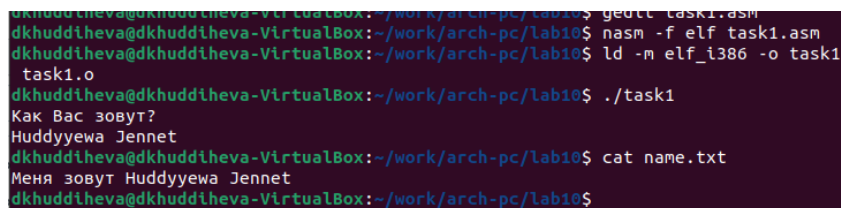
## 4.2 Задание для самостоятельной работы

Пишу код программы, выводящей приглашения “Как Вас зовут?”, считывающей с клавиатуры фамилию и имя и создающую файл, в который записывается сообщение “Меня зовут”ФИ”. (рис. [4.7])

```
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
12 mov eax,msg1
13 call sprintLF
14
15 mov ecx, name
16 mov edx, 255
17 call sread
18
19 mov ecx, 0777o
20 mov ebx, filename
21 mov eax, 8
22 int 80h
23
24 mov ecx, 2
25 mov ebx, filename
26 mov eax, 5
```

Рис. 4.7: Написание текста программы

Создаю исполняемый файл и проверяю его работу. Проверяю наличие файла и его содержимое с помощью команд `ls` и `cat`. (рис.[4.8])



```
dkhuddiheva@dkhuddiheva-VirtualBox: ~/work/arch-pc/lab10$ gcc task1.asm
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/arch-pc/lab10$ nasm -f elf task1.asm
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/arch-pc/lab10$ ld -m elf_i386 -o task1
task1.o
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/arch-pc/lab10$ ./task1
Как Вас зовут?
Huddyewwa Jennet
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/arch-pc/lab10$ cat name.txt
Меня зовут Huddyewwa Jennet
dkhuddiheva@dkhuddiheva-VirtualBox:~/work/arch-pc/lab10$
```

Рис. 4.8: Запуск исполняемого файла и проверка его работы

Программа работает корректно.

Код программы:

```
%include 'in_out.asm'
```

```
SECTION .data
```

```
msg1 db 'Как Вас зовут?', 0h
```

```
filename db 'name.txt', 0h
```

```
msg2 db 'Меня зовут ', 0h
```

```
SECTION .bss
```

```
name resb 255
```

```
SECTION .text
```

```
global _start
```

```
_start:

mov  eax,msg1

call sprintf

mov  ecx, name

mov  edx, 255

call sread

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 eax, name
```

```
call slen
```

```
mov edx, eax
```

```
mov ecx, name
```

```
mov ebx, esi
```

```
mov eax, 4
```

```
int 80h
```

```
mov ebx, esi
```

```
mov eax, 6
```

```
int 80h
```

```
call quit
```



## 5 Выводы

Благодаря данной лабораторной работе я приобрела навыки написания программ для работы с файлами.