

Отчёт по лабораторной работе №10

Дисциплина: Архитектура компьютера

Луангсуваннавонг Сайпхачан

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выполнение лабораторной работы	7
5	Выполнение заданий для самостоятельной работы	12
6	Выводы	17
7	Ответы на вопросы для самопроверки	18
8	Список литературы	20

Список иллюстраций

4.1	Создание файлов	7
4.2	Копирование файла	7
4.3	Редактирование файла	8
4.4	Редактирование файла	9
4.5	Запуск исполняемого файла	9
4.6	Чтение содержимого файла	9
4.7	Список информации о файлах	9
4.8	Изменение разрешения доступа к файлу	10
4.9	Запуск исполняемого файла	10
4.10	Изменение разрешения доступа к файлу	10
4.11	Запуск файла	11
4.12	Изменение разрешения доступа к файлу	11
4.13	Список информации о файлах	11
5.1	Создание файла	12
5.2	Редактирование файла	13
5.3	Запуск исполняемого файла	14

1 Цель работы

Целью данной лабораторной работы является приобретение навыков написания программ для работы с файлами.

2 Задание

1. Выполнение лабораторной работы
2. Выполнение заданий для самостоятельной работы

3 Теоретическое введение

ОС GNU/Linux является многопользовательской операционной системой. Для обеспечения защиты данных одного пользователя от действий других пользователей существуют специальные механизмы разграничения доступа к файлам. Кроме ограничения доступа, данный механизм позволяет разрешить другим пользователям доступ данным для совместной работы.

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. Владелцем файла является его создатель.

В операционной системе Linux существуют различные методы управления файлами, на пример, такие как создание и открытие файла, только для чтения или для чтения и записи, добавления в существующий файл, закрытия и удаления файла, предоставление прав доступа

Обработка файлов в операционной системе Linux осуществляется за счет использования определенных системных вызовов. Для корректной работы и доступа к файлу при его отсутствии открытия или создании, файлу присваивается уникальный номер (16-битное целое число) дескриптор файла.

4 Выполнение лабораторной работы

Я создаю новый каталог, в котором я создам файлы с программами для лабораторной работы № 10, используя команду `mkdir`. Затем, используя команду `touch`, я создаю несколько файлов, таких как `lab10-1.asm`, `readme-1.txt` и `readme-2.txt` (Рис .4.1)

```
sayprachanh@desktop:~$ mkdir work/study/2024-2025/Архитектура\ компьютера/arch-pc/lab10
sayprachanh@desktop:~$ cd work/study/2024-2025/Архитектура\ компьютера/arch-pc/lab10
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ touch lab10-1.asm readme-1.txt readme-2.txt
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ ls
lab10-1.asm  readme-1.txt  readme-2.txt
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$
```

Рис. 4.1: Создание файлов

Я копирую файл `in_out.asm` из последней лабораторной работы, так как он будет использоваться с другой программой.(Рис .4.2)

```
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ cp ~/work/study/2024-2025/Архитектура\ компьютера/arch-pc/lab09/in_out.asm .
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ ls
in_out.asm  lab10-1.asm  readme-1.txt  readme-2.txt
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$
```

Рис. 4.2: Копирование файла

Я открываю созданный файл `lab10-1.asm` и вставляю программу, которая будет записывать данные в файл сообщения, которым в данном случае является файл `.txt`.(Рис .4.3)

```

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

```

Рис. 4.3: Редактирование файла

В исходном коде программы указано, что входные данные будут записываться в readme.txt.(Рис .4.4) Однако я не хочу создавать новый текстовый файл, я решил вместо этого заставить программу записывать входные данные в файл readme-1.txt.


```

6  SECTION .data
7  filename db 'readme-1.txt', 0h          ; Имя файла
8  msg db 'Введите строку для записи в файл: ', 0h      ; Сообщение
9

```

Рис. 4.4: Редактирование файла

Я создаю исполняемый файл и запускаю его. Я ввожу в программу текстовую строку “Hello World!”(Рис .4.5)

```

sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ nasm -f elf -g -l lab10-1.lst lab10-1.asm
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ ld -m elf_i386 -o lab10-1 lab10-1.o
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ ./lab10-1
Введите строку для записи в файл: Hello World!
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$

```

Рис. 4.5: Запуск исполняемого файла

Используя команду cat, я отображаю содержимое текстового файла readme-1.txt.(Рис .4.6) Как мы видим, в нем отображаются данные, которые я ввел ранее. Это означает, что программа работает правильно.

```

sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ cat readme-1.txt
Hello World!
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$

```

Рис. 4.6: Чтение содержимого файла

Используя утилиту ls и клавишу -l, я перечисляю информацию о файлах в каталоге “lab10”. Эта информация включает имена файлов, время и дату их создания, а также права доступа к файлам, с которыми мы будем работать дальше.(Рис .4.7)

```

sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ ls -l
total 44
-rw-rw-r-- 1 sayprachanh sayprachanh 3773 Dec  9 17:43 in_out.asm
-rwxrwxr-x 1 sayprachanh sayprachanh 9800 Dec  9 17:58 lab10-1
-rw-rw-r-- 1 sayprachanh sayprachanh 1418 Dec  9 17:58 lab10-1.asm
-rw-rw-r-- 1 sayprachanh sayprachanh 14284 Dec  9 17:58 lab10-1.lst
-rw-rw-r-- 1 sayprachanh sayprachanh 2576 Dec  9 17:58 lab10-1.o
-rw-rw-r-- 1 sayprachanh sayprachanh 13 Dec  9 17:58 readme-1.txt
-rw-rw-r-- 1 sayprachanh sayprachanh 0 Dec  9 17:39 readme-2.txt
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$

```

Рис. 4.7: Список информации о файлах

Используя команду `chmod`, я изменяю права доступа к файлу исполняемого файла `lab10-1`, добавляя `-x` к команде `chmod`, я удаляю разрешение на выполнение файла. Используя команду `ls -l`, я проверяю работу команды. В результате в исполняемом файле `lab10-1` в части прав доступа к файлу отсутствует символ “`x`”, что означает, что программа не может быть запущена.(Рис .4.8)

```
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ chmod -x lab10-1
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ ls -l
total 44
-rw-rw-r-- 1 sayprachanh sayprachanh 3773 Dec  9 17:43 in_out.asm
-rw-rw-r-- 1 sayprachanh sayprachanh 9800 Dec  9 17:58 lab10-1
-rw-rw-r-- 1 sayprachanh sayprachanh 1418 Dec  9 17:58 lab10-1.asm
-rw-rw-r-- 1 sayprachanh sayprachanh 14284 Dec  9 17:58 lab10-1.lst
-rw-rw-r-- 1 sayprachanh sayprachanh 2576 Dec  9 17:58 lab10-1.o
-rw-rw-r-- 1 sayprachanh sayprachanh 13 Dec  9 17:58 readme-1.txt
-rw-rw-r-- 1 sayprachanh sayprachanh 0 Dec  9 17:39 readme-2.txt
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$
```

Рис. 4.8: Изменение разрешения доступа к файлу

Когда я пытаюсь запустить файл, он выдает ошибку ‘В разрешении отказано’.(Рис .4.9) Это означает, что его невозможно выполнить. Поскольку, когда я менял разрешение на доступ к файлу с помощью команды `chmod`, я не указывал, у кого я буду снимать разрешение. По умолчанию команда удалит разрешение на выполнение у всех, включая владельца, группу и других.

```
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ ./lab10-1
bash: ./lab10-1: Permission denied
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$
```

Рис. 4.9: Запуск исполняемого файла

Используя команду `chmod +x`, я добавляю разрешение на выполнение в файл `lab10-1.asm`, затем проверяю работу команды, используя команду `ls -l`.(Рис .4.10)

```
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ chmod +x lab10-1.asm
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ ls -l
total 68
-rw-rw-r-- 1 sayprachanh sayprachanh 3773 Dec  9 17:43 in_out.asm
-rw-rw-r-- 1 sayprachanh sayprachanh 9800 Dec  9 17:58 lab10-1
-rwxrwxr-x 1 sayprachanh sayprachanh 1418 Dec  9 17:58 lab10-1.asm
-rw-rw-r-- 1 sayprachanh sayprachanh 14284 Dec  9 17:58 lab10-1.lst
-rw-rw-r-- 1 sayprachanh sayprachanh 2576 Dec  9 17:58 lab10-1.o
```

Рис. 4.10: Изменение разрешения доступа к файлу

Я пытаюсь запустить файл lab10-1.asm, он выдает ошибку(Рис .4.11). Поскольку файл lab10-1.asm не является исполняемым файлом, это файл исходного кода (написанный на ассемблере). Следовательно, он не может быть выполнен напрямую, и когда вы попытаетесь его выполнить, он выдаст ошибку, и программа ничего не сделает.

```
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ ./lab10-1.asm
./lab10-1.asm: line 1: fg: no job control
./lab10-1.asm: line 3: SECTION: command not found
./lab10-1.asm: line 4: filename: command not found
./lab10-1.asm: line 4: Имя: command not found
./lab10-1.asm: line 5: msg: command not found
./lab10-1.asm: line 5: Сообщение: command not found
./lab10-1.asm: line 7: SECTION: command not found
./lab10-1.asm: line 8: contents: command not found
./lab10-1.asm: line 8: переменная: command not found
./lab10-1.asm: line 10: SECTION: command not found
./lab10-1.asm: line 11: global: command not found
./lab10-1.asm: line 12: _start:: command not found
./lab10-1.asm: line 14: syntax error near unexpected token `;'
./lab10-1.asm: line 14: `;--- Печать сообщения `msg`'
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$
```

Рис. 4.11: Запуск файла

Используя команду chmod, я установил соответствующие права доступа к файлам readme-1.txt и readme-2.txt.(Рис .4.12) Согласно моему варианту (вариант 13), для файла readme-1.txt будет в символьном виде (-w- -x -), а для файла readme-2.txt - в двоичной системе (110 011 001) Для readme-2.txt Я преобразую двоичную систему счисления в восьмеричную, в этом случае соответствующие права доступа к файлам будут равны 631 или в буквенном (rw- -wx -x)

```
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ chmod u-r,g-rw+x,o-r readme-1.txt
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ chmod 631 readme-2.txt
```

Рис. 4.12: Изменение разрешения доступа к файлу

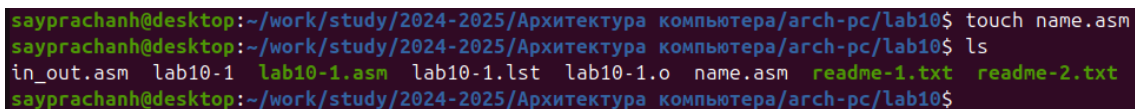
Затем, используя команду ls -l, я проверяю работу выполненной команды(Рис .4.13). Работа выполнена правильно.

```
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ ls -l readme-1.txt readme-2.txt
--w---x--- 1 sayprachanh sayprachanh 13 Dec  9 18:47 readme-1.txt
-rw--wx--x 1 sayprachanh sayprachanh  0 Dec  9 17:39 readme-2.txt
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$
```

Рис. 4.13: Список информации о файлах

5 Выполнение заданий для самостоятельной работы

Используя команду “touch”, я создаю файлы name.asm(Рис .5.1)



```
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ touch name.asm
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ ls
in_out.asm lab10-1 lab10-1.asm lab10-1.lst lab10-1.o name.asm readme-1.txt readme-2.txt
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$
```

Рис. 5.1: Создание файла

Я вхожу в программу, которая будет работать в соответствии с алгоритмом: выводит приглашение “Как вас зовут?”, запрашивает ввод данных, создает name.txt файл, записывает сообщение “Меня зовут” в name.txt файл, записывает вводимые данные в файл, затем закрывает файл.(Рис .5.2)

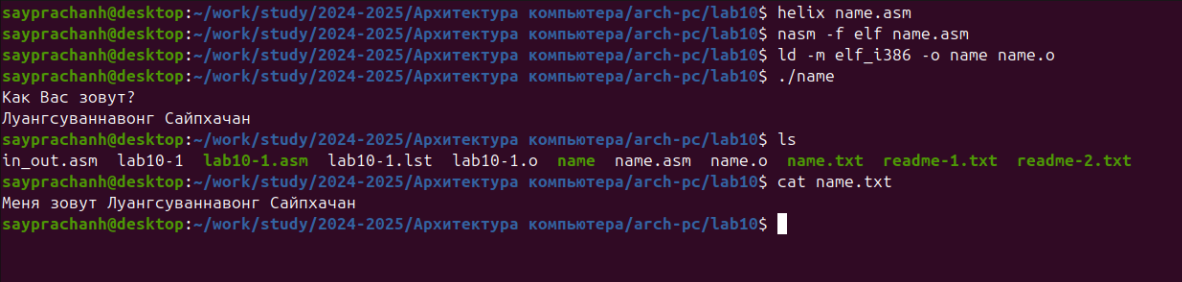
```

1  %include 'in_out.asm'
2
3  section .data
4  filename db 'name.txt', 0h
5  msg1 db 'Как Вас зовут?', 0h
6  msg2 db 'Меня зовут ', 0h
7
8
9  section .bss
10 name resb 255
11
12 section .text
13 global _start
14 _start:
15
16 mov eax, msg1
17 call sprintLF
18
19
20 mov ecx, name
21 mov edx, 255
22 call sread
23
24
25 mov eax, 8
26 mov ebx, filename
27 mov ecx, 0744o
28 int 80h
29
30 mov esi, eax
31
32
33 mov eax, msg2
34 call slen
35 mov edx, eax
36 mov ecx, msg2
37 mov ebx, esi
38 mov eax, 4

```

Рис. 5.2: Редактирование файла

Я создаю исполняемый файл и запускаю его(Рис .5.3). Я ввожу свою фамилию и свое имя в программу. Я проверяю работу программы, сначала используя команду ls, мы видим, что name.txt был создан в каталоге. Затем, используя команду cat, я отображаю содержимое файла name.txt, и действительно, в нем есть мое имя и текст сообщения.



```
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ helix name.asm
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ nasm -f elf name.asm
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ ld -m elf_i386 -o name name.o
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ ./name
Как Вас зовут?
Луангсуваннавонг Сайпхачан
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ ls
in_out.asm lab10-1 lab10-1.asm lab10-1.lst lab10-1.o name name.asm name.o name.txt readme-1.txt readme-2.txt
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$ cat name.txt
Меня зовут Луангсуваннавонг Сайпхачан
sayprachanh@desktop:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab10$
```

Рис. 5.3: Запуск исполняемого файла

Программа для выполнения задачи

```
%include 'in_out.asm'

section .data
filename db 'name.txt', 0h
msg1 db 'Как Вас зовут?', 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 eax, 8
```

```
mov ebx, filename
```

```
mov ecx, 0744o
```

```
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
```


6 Выводы

При выполнении данной лабораторной работы, Я приобрел навыки написания написания программ для работы с файлами.

7 Ответы на вопросы для самопроверки

1. Каким образом в Unix-подобных ОС определяются права доступа к файлу?

В Unix-подобных ОС права делятся на три категории: владельца, группы и остальных пользователей. Каждая категория может иметь права на чтение (r), запись (w) и выполнение (x). Эти права отображаются как строка, например, `rwXr-Xr--`.

2. Как ОС определяет, является ли файл исполняемым? Как регулировать права на чтение и запись?

Файл исполним, если у пользователя есть право на выполнение (x). Чтобы изменить права на чтение или запись, используется команда `chmod` (например, `chmod +r` для чтения, `chmod +w` для записи).

3. Как разграничить права доступа для различных категорий пользователей?

Права доступа можно настроить с помощью команд:

`chmod` — для изменения прав на файл,
`chown` — для изменения владельца файла,
`chgrp` — для изменения группы файла.

4. Какой номер имеют системные вызовы `sys_read`, `sys_write`, `sys_open`, `sys_close`, `sys_creat`.

Номера системных вызовов:

`sys_read`: 0

sys_write: 1

sys_open: 2

sys_close: 3

sys_creat: 8 (обычно используется через sys_open с флагом O_CREAT).

5. Какие регистры и как используют системные вызовы sys_read, sys_write, sys_open, sys_close

Для системных вызовов используются следующие регистры:

sys_read/sys_write: EBX (дескриптор файла), ECX (буфер), EDX (количество байт).

sys_open/sys_creat: EBX (имя файла), ECX (флаги), EDX (режим).

sys_close: EBX (дескриптор файла).

6. Что такое дескриптор файла?

Дескриптор файла — это число, которое процесс использует для работы с файлом. Он помогает операционной системе отслеживать открытые файлы и управлять операциями с ними, такими как чтение, запись и закрытие.

8 Список литературы

Архитектура ЭВМ