

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ
НАРОДОВ**

**Факультет физико-математических и естественных наук
Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 10.**

дисциплина: Архитектура компьютеров

Студент: Подхалюзина Виолетта Михайловна

Группа: НКАбд-04-24

МОСКВА

2024 г.

Оглавление

1	Цель работы	3
2	Введение.....	3
3	Выполнение лабораторной работы	4
3.1	Начало работы	4
3.2	Самостоятельная работа	5
4	Контрольные вопросы для самопроверки	8
5	Список литературы	9

1 Цель работы

Приобрести навыки написания программ для работы с файлами

2 Введение

ОС GNU/Linux является многопользовательской операционной системой. И 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. Для предоставления прав доступа другому пользователю или другой группе командой `chown [ключи] <новый_пользователь>[:новая_группа] <файл>` или `chgrp [ключи] < новая_группа > <файл>` Набор прав доступа задается тройками битов и состоит из прав на чтение, запись и исполнение файла. В символьном представлении он имеет вид строк `gwx`, где вместо любого символа может стоять дефис. Всего возможно 8 комбинаций, приведенных в таблице 10.1. Буква означает наличие права (установлен в единицу второй бит триады `g` — чтение, первый бит `w` — запись, нулевой бит `x` — исполнение), а дефис означает отсутствие права (нулевое значение соответствующего бита). Также права доступа могут быть представлены как восьмеричное число. Так, права доступа `gw-` (чтение и запись, без исполнения) понимаются как три двоичные цифры `110` или как восьмеричная цифра `6`.

3 Выполнение лабораторной работы

3.1 Начало работы

Я создаю каталог для программ лабораторной работы №10, перехожу в него и создаю файл lab10-1.asm, readme-1.txt и readme-2.txt

```
podkhalyuzina_violetta_113224676@violetta-Mint:~$ mkdir ~/work/arch-pc/lab10
podkhalyuzina_violetta_113224676@violetta-Mint:~$ cd ~/work/arch-pc/lab10/
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$ touch lab10-1.asm readme-1.txt readme-2.txt
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$ ls
lab10-1.asm  readme-1.txt  readme-2.txt
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$
```

Ввожу в файл lab10-1.asm текст программы из листинга 10.1 (Программа записи в файл сообщения), а после создаю исполняемый файл и проверяю его работу. Хочу заметить, что использовала файл readme-1.txt, а не readme.txt, который был указан в тексте заданий, потому что такой файл мы не создавали.

```
GNU nano 7.2 lab10
;-----
; Запись в файл строки введенной на запрос
;-----
%include 'in_out.asm'
SECTION .data
filename db 'readme-1.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
```

```
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$ nasm -f elf -g -l lab10-1.lst lab10-1.asm
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$ ld -m elf_i386 -o lab10-1 lab10-1.o
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$ ./lab10-1
Введите строку для записи в файл: Hello world!
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$ ls -l
итого 44
-rw-rw-r-- 1 podkhalyuzina_violetta_113224676 podkhalyuzina_violetta_113224676 3942 ноя  4 19:00 in_out.asm
-rwxrwxr-x 1 podkhalyuzina_violetta_113224676 podkhalyuzina_violetta_113224676 9760 дек 12 22:46 lab10-1
-rw-rw-r-- 1 podkhalyuzina_violetta_113224676 podkhalyuzina_violetta_113224676 1289 дек 12 22:43 lab10-1.asm
-rw-rw-r-- 1 podkhalyuzina_violetta_113224676 podkhalyuzina_violetta_113224676 13715 дек 12 22:45 lab10-1.lst
-rw-rw-r-- 1 podkhalyuzina_violetta_113224676 podkhalyuzina_violetta_113224676 2544 дек 12 22:45 lab10-1.o
-rw-rw-r-- 1 podkhalyuzina_violetta_113224676 podkhalyuzina_violetta_113224676 13 дек 12 22:55 readme-1.txt
-rw-rw-r-- 1 podkhalyuzina_violetta_113224676 podkhalyuzina_violetta_113224676 0 дек 12 22:35 readme-2.txt
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$ cat readme-1.txt
Hello world!
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$
```

С помощью команды `chmod` изменяю права доступа к исполняемому файлу `lab10-1`, запретив его выполнение. После этого пытаюсь выполнить файл.

```
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$ chmod 666 lab10-1
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$ ./lab10-1
bash: ./lab10-1: Отказано в доступе
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$
```

Я поменяла права владельца, запретив исполнять файл, после чего система отказала в исполнении файла, то есть, я запретила самой себе исполнять программу, так как владелец - я.

С помощью команды `chmod` я изменила права доступа к файлу `lab10-1.asm` с исходным текстом программы, добавив права на исполнение, после этого я попыталась выполнить его и поняла, что исполняемый текстовый файл интерпретирует каждую строку как команду, так как ни одна из строк не является командой `bash`, программа ничего не делает.

```
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$ chmod 777 lab10-1
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$ ./lab10-1
Введите строку для записи в файл: Hello world!
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$
```

В соответствии с вариантом в таблице 10.4 я предоставила права доступа к файлу `readme1.txt` представленные в символьном виде, а для файла `readme-2.txt` – в двоичном виде, после проверила правильность выполнения с помощью команды `ls -l`.

```
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$ chmod 771 readme-1.txt
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$ chmod 675 readme-2.txt
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$ ls -l
итого 44
-rw-rw-r-- 1 podkhalyuzina_violetta_113224676 podkhalyuzina_violetta_113224676 3942 ноя  4 19:00 in_out.asm
-rwxrwxr-x 1 podkhalyuzina_violetta_113224676 podkhalyuzina_violetta_113224676 9760 дек 12 22:46 lab10-1
-rw-rw-r-- 1 podkhalyuzina_violetta_113224676 podkhalyuzina_violetta_113224676 1289 дек 12 22:43 lab10-1.asm
-rw-rw-r-- 1 podkhalyuzina_violetta_113224676 podkhalyuzina_violetta_113224676 13715 дек 12 22:45 lab10-1.lst
-rw-rw-r-- 1 podkhalyuzina_violetta_113224676 podkhalyuzina_violetta_113224676 2544 дек 12 22:45 lab10-1.o
-rwxrwx-x 1 podkhalyuzina_violetta_113224676 podkhalyuzina_violetta_113224676 13 дек 12 23:22 readme-1.txt
-rw-rw-r-- 1 podkhalyuzina_violetta_113224676 podkhalyuzina_violetta_113224676 0 дек 12 22:35 readme-2.txt
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$
```

3.2 Самостоятельная работа

Я написала программу, которая работает по следующему алгоритму:

- Вывод приглашения “Как Вас зовут?”

- ввести с клавиатуры свои фамилию и имя
- создать файл с именем name.txt
- записать в файл сообщение “Меня зовут”
- дописать в файл строку введенную с клавиатуры
- закрыть файл

```
GNU nano 7.2
%include 'in_out.asm'
SECTION .data
filename db 'name.txt', 0
prompt db 'Как Вас зовут?', 0
intro db 'Меня зовут ', 0
SECTION .bss
name resb 255
SECTION .text
global _start
_start:
mov eax, prompt
call sprint
mov ecx, name
mov edx, 255
call sread
mov eax, 8
mov ebx, filename
mov ecx, 0744o
int 80h
mov esi, eax
mov eax, intro
call slen
mov edx, eax
mov ecx, intro
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

podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$ nasm -f elf sWork.asm
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$ ld -m elf_i386 -o sWork sWork.o
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$ ./sWork
Как Вас зовут?Виолетта
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$ 
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$ ls
in_out.asm  lab10-1.asm  lab10-1.o  readme-1.txt  sWork      sWork.o
lab10-1     lab10-1.lst  name.txt   readme-2.txt  sWork.asm
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$ 
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$ cat name.txt
Меня зовут Виолетта
podkhalyuzina_violetta_113224676@violetta-Mint:~/work/arch-pc/lab10$ 
```

Вывод: в процессе выполнения лабораторной работы я приобрела навыки

написания программ для работы с файлами, научилась редактировать права
для файлов.

4 Контрольные вопросы для самопроверки

1. В Unix-подобных ОС права доступа к файлу определяются тройками битов, представляющими разрешения на чтение, запись и выполнение. Эти права задаются для трех категорий пользователей: владельца, группы владельца и всех остальных. Например, права `rwX r-X --X` обозначают полный доступ для владельца, чтение и выполнение для группы и только выполнение для остальных. Управлять правами можно через команды `chmod`, `chown` и `chgrp`.

2. ОС определяет, является ли файл исполняемым, по наличию разрешения на выполнение (бит `x`) в правах доступа. Проверить это можно через команду `ls -l`. Управлять правами на чтение и запись можно через `chmod`, устанавливая соответствующие биты (`r` для чтения и `w` для записи).

3. Права доступа разграничиваются на уровне категорий пользователей: владелец (`u`), группа (`g`) и все остальные (`o`). Для каждой категории можно установить свои права, используя команды, такие как `chmod`, `chown` и `chgrp`.

4. Номера системных вызовов:

- `sys_read`: 3
- `sys_write`: 4
- `sys_open`: 5
- `sys_close`: 6
- `sys_creat`: 8

5. Использование регистров в системных вызовах: - `sys_read`: `eax` (номер вызова), `ebx` (дескриптор файла), `ecx` (адрес в памяти), `edx` (количество байтов). - `sys_write`: `eax` (номер вызова), `ebx` (дескриптор файла), `ecx` (адрес строки), `edx` (количество байтов). - `sys_open`: `eax` (номер вызова), `ebx` (имя файла), `ecx` (режим доступа), `edx` (права доступа). - `sys_close`: `eax` (номер вызова), `ebx` (дескриптор файла). - `sys_creat`: `eax` (номер вызова), `ebx` (имя файла), `ecx` (права доступа).

6. Дескриптор файла — это уникальный идентификатор (обычно 16-битное целое число), который присваивается файлу при его открытии или создании. Он используется для последующих операций с файлом, таких как чтение, запись или закрытие.

5 Список литературы

1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learningbash-Shell-Programming-Nutshell/dp/0596009658>.
6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017.
11. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
13. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВПетербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
14. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: http://www.stolyarov.info/books/asm_unix.

15. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).

16. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).

