

# **Отчёт по лабораторной работе №10**

Дисциплина: архитектура компьютеров и операционные системы

Светцова Анна

## **Содержание**

### **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 `EDX`, 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 `EDX`, 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. 1)

```
cd 01_directory
adsvetsova@annpc ~$ mkdir -p /work/study/2023-2024/"Computer architecture"/arh-pc/labs/lab10/lab10
adsvetsova@annpc ~$ cd -/work/study/2023-2024/"Computer architecture"/arh-pc/labs/lab10/lab10
adsvetsova@annpc ~/w/s/2/C/arh-pc/l/lab10/lab10 master +4 130 714 $ touch lab10-1.asm readme-1.txt readme-2.txt
adsvetsova@annpc ~/w/s/2/C/arh-pc/l/lab10/lab10 master +4 130 715 $
```

рис.1 Создание файлов для лабораторной работы

Ввожу в файл lab10-1.asm текст программы, записывающей в файл сообщения, из листинга 10.1. (рис. 2)

```
Open  [F]  -/work/study/2023-2024/Computer architecture/arh-pc/labs/lab10/lab10  Save  [M]
lab10-1.asm  name.asm
2 SECTION .data
3 filename db 'readme.txt', 0h ; имя файла
4 msg db 'Введите строку для записи в файл: ', 0h ; Сообщение
5 SECTION .bss
6 contents resb 255 ; переменная для вводной 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 ecx, 255
16 call read
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
25 mov ecx, contents ; в 'eax' запишется количество
26 call strlen ; введенных байтов
27 ; Записываем в файл 'contents' ('sys_write')
28 mov ecx, eax
29 mov ecx, contents
30 mov ebx, esi
31 mov ecx, 4
32 int 80h
33 ; Закрываем файл ('sys_close')
34 mov ebx, esi
35 mov eax, 6
36 int 80h
37 call quit
38
```

рис.2 Ввод текста программы из листинга 10.1

Создаю исполняемый файл и проверяю его работу. (рис. 3)

```
adsvetsova@annpc ~/w/s/2/C/arh-pc/l/lab10/lab10 master +4 130 715 $ nasm -f elf
lab10-1.asm
adsvetsova@annpc ~/w/s/2/C/arh-pc/l/lab10/lab10 master +4 130 715 $ ld -m elf_i
386 -o lab10-1.asm lab10-1.o
adsvetsova@annpc ~/w/s/2/C/arh-pc/l/lab10/lab10 master +4 130 715 $ ./lab10-1.a
sm
Введите строку для записи в файл: hello world
adsvetsova@annpc ~/w/s/2/C/arh-pc/l/lab10/lab10 master +4 130 715 $ cat readme-
1.txt
adsvetsova@annpc ~/w/s/2/C/arh-pc/l/lab10/lab10 master +4 130 715 $ cat readme-
1.txt
hello world
```

рис.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)

```
adsvetsova@annpc ~/w/s/2/C/arh-pc/l/lab10/lab10 master +4 130 715 $ chmod u-x l
ab10-1.asm
adsvetsova@annpc ~/w/s/2/C/arh-pc/l/lab10/lab10 master +4 130 715 $ ./lab10-1.a
sm
zsh: permission denied: ./lab10-1.asm
adsvetsova@annpc ~/w/s/2/C/arh-pc/l/lab10/lab10 master +4 130 715 $
```

рис.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. 5)

```
adsvetsova@annpc ~/w/s/2/C/ark-pc/l/lab10/lab10 master +4 130 715 $ chmod u+x l
ab10-1.asm
adsvetsova@annpc ~/w/s/2/C/ark-pc/l/lab10/lab10 master +4 130 715 $ ./lab10-1.a
sm
Введите строку для записи в файл: hello world
adsvetsova@annpc ~/w/s/2/C/ark-pc/l/lab10/lab10 master +4 130 715 $ chmod 631
```

рис.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 (10) в таблице 10.4 предоставляю права доступа к файлу `readme1.txt` представленные в символьном виде, а для файла `readme-2.txt` – в двоичном виде:

`rw- -wx -x 010 001 000`

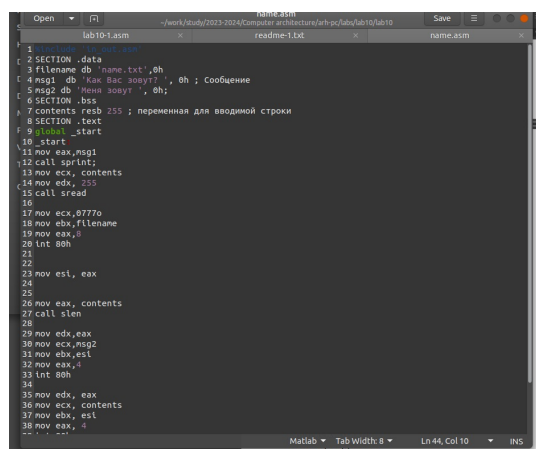
И проверяю правильность выполнения с помощью команды `ls -l`. (рис. 6)

```
adsvetsova@annpc ~/w/s/2/C/ark-pc/l/lab10/lab10 master +4 130 715 $ chmod 631
readme-1.txt
adsvetsova@annpc ~/w/s/2/C/ark-pc/l/lab10/lab10 master +4 130 715 $ chmod 210
readme-2.txt
adsvetsova@annpc ~/w/s/2/C/ark-pc/l/lab10/lab10 master +4 130 715 $ ls -l
total 24
-rw-rw-r-- 1 adsvetsova adsvetsova 3942 ноя 18 17:29 in_out.asm
-rwxrwxr-x 1 adsvetsova adsvetsova 9212 дек 16 17:12 lab10-1.asm
-rw-rw-r-- 1 adsvetsova adsvetsova 1472 дек 16 17:12 lab10-1.o
-rw-rwxr-x 1 adsvetsova adsvetsova 13 дек 16 17:14 readme-1.txt
-w---x-- 1 adsvetsova adsvetsova 0 дек 16 15:13 readme-2.txt
adsvetsova@annpc ~/w/s/2/C/ark-pc/l/lab10/lab10 master +4 130 715 $
```

рис.6 Предоставление прав доступа в символьном и двоичном виде

## 4.2 Задание для самостоятельной работы

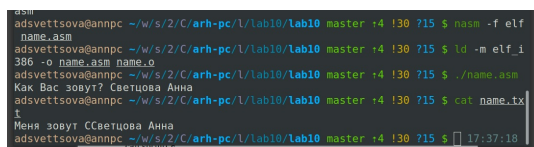
Пишу код программы, выводящей приглашения “Как Вас зовут?”, считывающей с клавиатуры фамилию и имя и создающую файл, в который записывается сообщение “Меня зовут”ФИ””. (рис. 7)



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

рис.7 Написание текста программы

Создаю исполняемый файл и проверяю его работу. Проверяю наличие файла и его содержимое с помощью команд ls и cat. (рис. 8)



```
adsvettsova@annpc ~/w/s/2/C/erh-pc/l/lab10/lab10 master +4 130 715 $ nasm -f elf
name.asm
adsvettsova@annpc ~/w/s/2/C/erh-pc/l/lab10/lab10 master +4 130 715 $ ld -m elf_i
386 -o name.asm name.o
adsvettsova@annpc ~/w/s/2/C/erh-pc/l/lab10/lab10 master +4 130 715 $ ./name.asm
Как Вас зовут? Светцова Анна
adsvettsova@annpc ~/w/s/2/C/erh-pc/l/lab10/lab10 master +4 130 715 $ cat name.tx
t
Меня зовут Светцова Анна
adsvettsova@annpc ~/w/s/2/C/erh-pc/l/lab10/lab10 master +4 130 715 $
```

рис.8 Запуск исполняемого файла и проверка его работы

Программа работает корректно.

Код программы:

```
%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
```

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

## 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](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).